УДК [519.876.5:004.942]:004.05 DOI: 10.31673/2412-4338.2024.023648

Шантир Антон Сергійович
Державний університет інформаційно-комунікаційних технологій, м.Київ
ORCID 0000-0002-0466-3659

Чичкарьов Євген Анатолійович
Державний університет інформаційно-комунікаційних технологій, м.Київ
ORCID 0000-0002-4362-5129

Березівський Максим Юрійович
Державний університет інформаційно-комунікаційних технологій, м.Київ
ORCID 0000-0002-6602-2515

Зінченко Вячеслав Валентинович
Державний університет інформаційно-комунікаційних технологій, м.Київ
ORCID 0009-0000-7087-1678

ПРИНЦИП ЗАСТОСУВАННЯ ІЄРАРХІЧНОГО ПІДХОДУ В МЕЖАХ СТВОРЕННІЯ БАГАТО-ЦІЛЬОВИХ МОДЕЛЕЙ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

Анотація: *У даній статті розглядається принцип застосування ієрархічного підходу (ІП) в контексті розробки багатоцільових моделей якості програмних систем (ПС). Автори аналізують основні концепції та методи, пов'язані з ієрархічним моделюванням, а також визначають його переваги та обмеження. Досліджуються можливості використання ієрархічної структури для підвищення ефективності процесу оцінки якості програмних систем. Робиться акцент на важливості правильного побудови ієрархічних моделей якості з урахуванням специфіки конкретної системи. Висвітлюються можливі підходи до інтеграції ієрархічних моделей у процес розробки програмного забезпечення для забезпечення високої якості продукту. Мета статті полягає в дослідженні, аналізі та розробці методів інтеграції ієрархічного підходу в створенні багатоцільових моделей якості програмних систем з метою поліпшення процесу їхнього оцінювання та управління. Реалізація поставленої мети передбачає вирішення наступних цілей: Провести аналіз особливостей деталізованого порівневого ієрархічного розбиття механізму оцінювання якості ПС; Провести аналіз застосовування конкретних математичних теорій при формуванні принципу ієрархічної структури багатоцільових моделей якості ПС; Розглянути основні етапи принципу ІП до оцінювання якості ПС; Розробити математичний апарат для моделі на базі реалізації порівневого ієрархічного розбиття механізму оцінювання якості ПС. Висновки статті можуть бути корисними для фахівців у галузі розробки програмного забезпечення та викладачів, що викладають дисципліни з якості програмного забезпечення. В межах розробки математичного апарату для моделей на основі ієрархічного розбиття механізму оцінювання якості ПС: розроблено математичний інструментарій, який включає формульний математичний апарат для оцінювання якості ПС. Важливо підкреслити потребу у додаткових дослідженнях для вдосконалення цього математичного апарату та його відповідності конкретним вимогам та умовам використання на практиці.*

Ключові слова*: програмне забезпечення, метрики якості, потреби користувачів, інформаційні технології, підтримка системи, стандарти якості, математичний апарат, ієрархічні моделі, програмні засоби.*

Shantyr Anton
State University of Information and Communication Technologies, Kyiv
ORCID 0000-0002-0466-3659

Chychkarov Yevgen
State University of Information and Communication Technologies, Kyiv
ORCID 0000-0002-4362-5129

Berezivskyi Maksym

State University of Information and Communication Technologies, Kyiv
ORCID 0000-0002-6602-2515

Zinchenko Viacheslav
State University of Information and Communication Technologies, Kyiv
ORCID 0009-0000-7087-1678

PRINCIPLE OF APPLYING THE HIERARCHICAL APPROACH WITHIN THE FRAMEWORK OF CREATING MULTI-OBJECTIVE QUALITY MODELS OF SOFTWARE SYSTEMS

Abstract: *The article discusses the principle of applying a hierarchical approach (HA) in the context of developing multi-objective quality models for software systems (SS). The authors analyze the main concepts and methods associated with hierarchical modeling, as well as identify its advantages and limitations. The possibilities of using a hierarchical structure to enhance the efficiency of the software system quality evaluation process are explored. Emphasis is placed on the importance of constructing hierarchical quality models considering the specificities of individual systems. Various approaches to integrating hierarchical models into the software development process to ensure high product quality are highlighted. The aim of the article is to research, analyze, and develop methods for integrating the hierarchical approach in creating multi-objective quality models for software systems to improve their evaluation and management processes. The realization of this aim involves addressing the following objectives: conducting an analysis of the peculiarities of detailed hierarchical breakdown mechanisms for assessing SS quality; conducting an analysis of the application of specific mathematical theories in forming the hierarchical structure of multi-objective quality models for SS; examining the main stages of the HA principle for SS quality evaluation; and developing a mathematical apparatus for the model based on the implementation of hierarchical breakdown mechanisms for assessing SS quality. The conclusions of the article may be useful for professionals in the software development field and educators teaching software quality disciplines. In the development of the mathematical apparatus for models based on hierarchical breakdown mechanisms for assessing SS quality, a mathematical toolkit has been developed, including a formulaic mathematical apparatus for SS quality assessment. It is essential to emphasize the need for further research to improve this mathematical apparatus and its suitability for specific usage requirements and conditions.*

Keywords: *software, quality metrics, user needs, information technology, system support, quality standards, mathematical framework, hierarchical models, software tools.*

Вступ

У швидкозмінному світі інженерії програмного забезпечення якості програмних систем стає надзвичайно важливим. Зі зростанням складності та різноманітності програмного забезпечення (ПЗ) зросла потреба у надійних моделях якості, які ефективно можуть відображати та оцінювати різні характеристики якості. У цьому контексті ІП виступає, як фундаментальний принцип у створенні багатоцільових моделей якості ПС.

У цій статті ми досліджуємо застосування ІП у контексті створення комплексних моделей якості для програмних систем. Аналізуючи основні концепції та методології, пов'язані з ієрархічним моделюванням, ми маємо на меті прояснити його значення, переваги та обмеження в контексті оцінки якості програмного забезпечення. Ми досліджуємо, як ієрархічні структури можуть підвищити ефективність процесу оцінки якості та розглядаємо важливість побудови ієрархічних моделей якості з урахуванням специфіки окремих систем.

Крім того, ми обговорюємо потенційні підходи до інтеграції ієрархічних моделей у процес розробки програмного забезпечення з метою забезпечення високої якості продуктів. Отримані у цій статті уваги можуть бути корисними для фахівців у галузі розробки програмного забезпечення та для викладачів, які викладають курси з забезпечення якості програмного забезпечення. За допомогою ІП зацікавлені сторони можуть більш ефективно орієнтуватися у складних процесах оцінки якості програмного забезпечення, що в кінцевому підсумку сприяє розвитку надійних та ефективних програмних систем.

Постановка проблеми

Незважаючи на значний прогрес у сфері розробки ПЗ, забезпечення якості залишається актуальним завданням. Сучасні ПС стають все складнішими та різноманітнішими, що ускладнює процес оцінки їх якості. Традиційні підходи до визначення якості, такі як моделі водопаду, або спіралі, можуть бути недостатньо ефективними для адаптації до нових вимог та умов ринку. Крім того, важливість багатогранності якості, яка охоплює не лише технічні аспекти, але й користувацькі вимоги та бізнесову цінність, потребує розробки моделей, які здатні враховувати цю комплексність. Таким чином, постановка проблеми полягає в розробці ефективних та адаптивних моделей якості, які можуть враховувати складність сучасних програмних систем та задовольняти різноманітні потреби зацікавлених сторін. Зокрема дослідження які пов’язані із аналізом аспектів принципового застосування ІП в межах створення багато-цільових моделей якості програмних систем, відображає практичну потребу в розробці та застосуванні ефективних методів оцінки якості програмних систем. В загальному плані проблема полягає в тому, що потрібно на методично-практичному рівні описати, як застосувати ІП у створенні моделей якості, для того, щоб вони які були більш цілеспрямованими та комплексними. Також підтята тематична спрямованість водночас пов’язана із практичною потребою у розробці покращених методів оцінки якості ПЗ, щоб врахувати різноманітність вимог та контекстуальні умови в сучасному програмному середовищі. Таким чином, стаття спрямована на розв'язання конкретної практичної проблеми, яка відображена в її назві, а саме, як ефективно застосовувати ІП у створенні багатоцільових моделей якості програмних систем, що відповідають потребам сучасного програмного ринку.

Аналіз останніх досліджень та публікацій

В праці [1] вказується, що принцип ІП дозволяє розглядати складність якості ПС, як систему взаємопов'язаних підсистем, що сприяє більш системному підходу до їхньої оцінки. На практиці це означає, що при використанні ІП розробники можуть розглядати ПС, як сукупність різних компонентів, які взаємодіють між собою. Наприклад, якщо ми розглядаємо якість веб-додатку, ми можемо розглядати окремо фронтенд, бекенд, базу даних, а також їхню взаємодію. Це дозволяє отримати більш повний та системний огляд системи, що полегшує процес оцінки її якості та виявлення можливих проблем.

Натомість в праці [2] відмічається, що ІП дозволяє зберігати структурованість та організованість в процесі розробки багатоцільових моделей якості, що полегшує їх розуміння та використання. На практиці це означає, що при використанні ІП розробники можуть систематизувати та структурувати моделі якості, розглядаючи їх на різних рівнях абстракції. Наприклад, модель може включати загальні аспекти якості на вищому рівні і більш деталізовані аспекти на нижчому рівні. Це допомагає розробникам легше розуміти та використовувати моделі якості під час розробки ПС.

В праці [3] вказано, що використання ІП у створенні моделей якості дозволяє ефективно управляти складністю та об'ємом інформації, що враховується при оцінці якості ПС. На практиці це означає, що при використанні ІП розробники можуть керувати обсягом інформації та складністю моделей якості. Розбиття моделі на менші компоненти дозволяє краще керувати об'ємом даних та зробити їх більш зрозумілими та керованими. Це полегшує процес оцінки якості та дозволяє розробникам зосередитися на найважливіших аспектах.

Відповідно авторським колективом праці [4] відмічається, що ІП може забезпечити більшу гнучкість та адаптивність у врахуванні різноманітних вимог до якості ПС з різних точок зору зацікавлених сторін. На практиці це означає, що розробники можуть адаптувати моделі якості до конкретних вимог та потреб різних зацікавлених сторін, таких як клієнти, користувачі, менеджмент, аналітики тощо. Також ІП дозволяє створювати моделі, які можуть бути гнучко адаптовані до різних вимог та умов [5]. Зокрема згідно [6] даний принцип дозволяє систематизувати різні аспекти якості ПС на різних рівнях абстракції, що сприяє кращому розумінню та управлінню цими аспектами. На практиці це означає, що розробники

можуть організувати аспекти якості на різних рівнях деталізації, починаючи від загальних аспектів та переходячи до деталей. Це дозволяє краще розуміти та керувати різними аспектами якості, роблячи їх більш доступними для аналізу та управління.

В межах аналізу застосування ІП в праці [7] відмічено, що завдяки його застосуванню створюється можливість для інтеграції різноманітних методів оцінки якості на різних рівнях моделі, що дозволяє отримувати більш об'єктивні та комплексні результати. На практиці це означає, що розробники можуть використовувати різні методи оцінки якості на різних рівнях моделі для отримання більш об'єктивних та повних результатів. Інтеграція цих методів дозволяє отримати комплексні результати, які враховують різні аспекти якості ПС.

В праці [8] відмічено, що використання ІП допомагає зробити оцінку якості ПС більш структурованою та системною, що полегшує виявлення пріоритетів та прийняття рішень. На практиці це означає, що розробники можуть організувати процес оцінки якості ПС у вигляді структурованої ієрархії, яка включає різні рівні та аспекти. Це допомагає легше визначити пріоритетні напрямки для вдосконалення якості та прийняти обґрунтовані рішення.

В праці [10] відмічено, що поділ якості на більш прості та зрозумілі компоненти за допомогою ІП дозволяє краще виявляти та усувати проблеми в програмному забезпеченні. На практиці це означає, що розробники можуть розглядати якість програмного забезпечення як сукупність окремих аспектів та компонентів. Це дозволяє легше виявляти та аналізувати проблемні місця, що дозволяє швидше та ефективніше усувати їх.

Згідно [11] ІП сприяє розподілу відповідальностей за оцінку якості між різними командами та експертами, що може полегшити та прискорити процес. На практиці це означає, що різні команди та експерти можуть бути відповідальні за оцінку якості на різних рівнях абстракції. Це дозволяє зменшити навантаження на окремих спеціалістів та розподілити відповідальності, що може покращити ефективність процесу оцінки.

У відповідності до [12] було доведено, що застосування ІП дозволяє більш точно адаптувати моделі якості до конкретних потреб та характеристик ПС. На практиці це означає, що розробники можуть налаштовувати моделі якості з урахуванням конкретних вимог та особливостей кожної програмної системи. Це дозволяє отримувати більш точні та адаптовані результати оцінки.

В праці [13] досить слушно відмічено, що нині існує досить велика кількість можливих напрямків для вдосконалення якості програмного забезпечення, що ускладнює вибір та прийняття рішення, проте ІП дозволяє систематизувати ці напрямки та вибрати найбільш пріоритетні.

В праці [14] відмічено, що використання ІП в створенні моделей якості допомагає уникнути перевантаження інформацією та забезпечує більш зрозумілі та компактні результати оцінки.

Проте в працях [1–14] лише частково проаналізовані проблеми, які зв'язані з принципом застосування ІП в межах створення багатоцільових моделей якості програмних систем, а також потребою додаткових досліджень. Зокрема в праці [1] відмічена проблема щодо складності практичного застосування ІП, яка пов’язана із моделюванням ієрархічних структур: побудова ієрархічних моделей якості може бути складною через потребу визначення відповідних рівнів абстракції та встановлення зв'язків між ними. На практиці дана проблема зводиться до того, що розробники потребують додаткових досліджень для розробки ефективних методів побудови та управління ієрархічними моделями. В праці [2] відмічається недостатня точність оцінки на різних рівнях ієрархії: Оцінка якості на нижчих рівнях може бути менш точною через спрощення та агрегацію даних. Відповідно на практиці необхідно дослідження для розробки методів, які забезпечать належну точність оцінки якості на різних рівнях ієрархії.

В праці [3] вказується проблема, яка пов’язана із управлінням зв'язками між рівнями ієрархії: необхідно ефективно керувати зв'язками та залежностями між різними рівнями ієрархії, щоб забезпечити консистентність та вірогідність моделей. Відповідно на практиці

виникає потреба в проведені дослідженні цього питання, щодо пошуку можливостей вдосконалити методики управління зв'язками та забезпечити надійність моделей.

В працях [4–7] відмічена проблема, щодо співвідношення між деталізацією та агрегацією даних: важливо знайти оптимальний баланс між деталізацією даних на нижчих рівнях ієрархії та їх агрегацією на вищих рівнях. В результаті виникає нагальна потреба в проведенні додаткових дослідженнь для розробки методів, які допоможуть забезпечити належний баланс між деталізацією та агрегацією даних.

В працях [8–9] відмічена проблема практичного застосування ІП, яка пов'язана із узгодженістю ієрархічних моделей з бізнес-потребами: важливо, щоб ієрархічні моделі відображали дійсні бізнес-потреби та вимоги зацікавлених сторін. Усі вище наведені проблеми демонструють необхідність подальших досліджень у галузі ієрархічних моделей якості ПС для їхнього ефективного використання на практиці.

Таким чином за піднятою тематикою виникає нагальна потреба в проведені додаткових досліджень.

Формулювання цілі статті

Мета статті полягає в дослідженні, аналізі та розробці методів інтеграції ієрархічного підходу в створенні багатоцільових моделей якості програмних систем з метою поліпшення процесу їхнього оцінювання та управління. Реалізація поставленої мети передбачає вирішення наступних цілей:

1. Провести аналіз особливостей деталізованого порівневого ієрархічного розбиття механізму оцінювання якості ПС;
2. Провести аналіз застосовування конкретних математичних теорій при формуванні приципу ієрархічної структури багатоцільових моделей якості ПС;
3. Розглянути основні етапи принципу ІП до оцінювання якості ПС;
4. Розробити математичний апарат для моделі на базі реалізації порівневого ієрархічного розбиття механізму оцінювання якості ПС.

Виклад основного матеріалу

В межах порушеної тематичної спрямованості застосуємо ІП до моделювання механізмів оцінювання якості ПС, який математично можна описати наступним чином:

Нехай $Q_1, Q_2, \ldots, Q_n$ будуть метриками якості на рівні деталізації, де n – кількість рівнів в ієрархії. На кожному рівні можуть бути визначені відповідні метрики якості, що відображають конкретні аспекти якості системи на цьому рівні деталізації.

Тоді загальна метрика якості $Q_{ієрарх.}$ може бути виражена, як сума метрик на всіх рівнях ієрархії (1):

$$Q_{ієрарх.} = Q_1 + Q_2 + \ldots + Q_n, \quad (1)$$

де Q_i – може бути додатково розкрита як сукупність більш деталізованих метрик на більш низьких рівнях ієрархії.

Цей підхід дозволяє систематично розглядати та моделювати різні аспекти якості програмних систем, починаючи від загальних концепцій та розподіляючи їх на більш конкретні аспекти на різних рівнях абстракції.

ІП до моделювання процесу оцінювання якості програмних систем базується на розбитті процесу на більш прості та керовані частини [5]. Основна ідея полягає в тому, щоб розглядати процес оцінки якості ПС на різних рівнях деталізації, починаючи від загальних оцінок та переходячи до детального аналізу окремих аспектів [8].

В табл.1. наведено результати аналізу особливостей деталізованого порівневого ієрархічного розбиття механізму оцінювання якості ПС.

Таблиця 1.

Результати аналізу особливостей деталізованого порівневого ієрархічного розбиття механізму оцінювання якості ПС

Рівень	Опис	Підрівні	Вага	Особливості
Рівень 1	Оцінка загальної продуктивності ПС	Відгуковість	0.2	Важливо для користувачів, впливає на сприйняття швидкості реакції системи.
		Пропускна здатність	0.3	Важливо для високозавантажених систем, впливає на швидкодію обробки запитів.
		Швидкодія	0.5	Впливає на загальний враження від роботи програми, може бути критичною для деяких додатків.
Рівень 2	Оцінка надійності і безпеки ПС	Відновлюваність	0.4	Важливо для забезпечення безперервної роботи системи в разі виникнення помилок або відмов.
		Відмовостійкість	0.4	Впливає на стійкість системи до внутрішніх і зовнішніх впливів, таких як помилки програмного забезпечення або кібератаки.
		Захищеність	0.2	Важливо для забезпечення конфіденційності, цілісності та доступності даних.
Рівень 3	Оцінка ефективності ресурсів ПС	Використання пам'яті	0.5	Впливає на ефективне використання обсягу пам'яті, важливо для оптимізації ресурсів.
		Використання процесорного часу	0.5	Впливає на ефективне використання обчислювальних можливостей, важливо для оптимізації швидкодії.
Рівень 4	Оцінка зручності користування ПС	Інтуїтивність	0.6	Важливо для забезпечення зручного та інтуїтивного користування системою, впливає на швидкість навчання та задоволення користувачів.
		Документація	0.4	Важливо для надання інформації та підтримки користувачам, зменшує час на вирішення проблем та недорозумінь.
Рівень 5	Оцінка інтеграційної сумісності ПС з іншими програмами	Сумісність з мовами програмування	0.5	Важливо для інтеграції з іншими системами та розширення функціональності.
		Інтеграція з іншими програмами	0.5	Важливо для забезпечення спільної роботи з іншими програмами та системами.

Відповідно табл.1. може стати корисним інструментом для команд розробників та тестувальників для покращення якості програмної системи на всіх її рівнях. В спектрі аналізу наведеної інформативності в табл. 1 досить деталізовано розглянута узагальнена структура оцінки якості ПС на різних рівнях. Зокрема в даній таблиці межах реалізації ієрархічної деталізації рівнів якості: кожен рівень має свої власні підрівні, що дозволяє ретельно розглянути різні аспекти якості програмної системи. Завдяки відповідному присвоєнню ваги кожному підрівню на практиці виникає можливість визначити їхню значимість у загальному процесі оцінюванні якості ПС, де така структура оцінювання в межах подачі інформації дозволяє організувати процес аналізу якості програмної системи, розбивши його на окремі етапи та аспекти.

В результаті застосування деталізованого розгляду різних аспектів якості ПС на практиці можна легше виявити та вирішити проблеми в конкретних областях.

В табл. наведено результати аналізу застосування конкретних математичних теорій при формуванні принципу ієрархічної структури багатоцільових моделей якості ПС.

Таблиця 2

Результати аналізу застосування конкретних математичних теорій при формуванні приципу ієрархічної структури багатоцільових моделей якості ПС

Модель якості ПС	Застосована математична теорія	Ієрархія застосування	Переваги	Недоліки	Оптимізація
ISO/IEC 25010	Системний аналіз	Опис категорій якості	Стандартизація характеристик якості; широкий охоплення	Складність використання для невеликих проектів; нефлекскісність	Підвищення гнучкості; спрощення використання
ISO/IEC 9126	Теорія функціональних та нефункціональних вимог	Дефініція атрибутів якості	Добре вивчена; простота застосування	Обмежений охоплення; застарілість	Розширення охоплення; апгрейд до сучасних вимог
QAW	Групова робота	Аналіз потреб стейкхолдерів	Стейкхолдерсь-кий підхід; гнучкість	Залежність від стейкхолдері; суб'єктивність	Оптимізація взаємодії зі стейкхолдерами
ATAM	Архітектурний аналіз	Оцінка впливу архітектури	Архітектурна спрямованість; управління ризиками	Складність використання; залежність від експертів	Спрощення процесу оцінки; підвищення доступності експертів
GQM	Метричні та цільові підходи	Формулювання цілей	Об'єктивність; адаптивність	Складність формалізації; велика кількість метрик	Оптимізація процесу формулювання цілей та метрик

З табл.2 детальний аналіз переваг та недоліків застосування конкретних математичних теорій при формуванні принципу ієрархічної структури та особливостей кожної з багатоцільових моделей якості ПС показав, що модель ISO/IEC 25010 (SQuaRE) завдяки використанню міжнародних стандартів ISO/IEC, забезпечує однорідність і зрозумілість у визначенні характеристик та метрик якості ПС. Також модель ISO/IEC 25010 (SQuaRE) [5] охоплює широкий спектр характеристик якості, від функціональних до нефункціональних. Проте завдяки широкому охопленню, зазначена модель може бути складною для реалізації та використання, особливо в менших організаціях. Серед недоліків даної моделі якості є те що вона інкапсулює в собі багато характеристик, але не завжди може бути достатньо гнучкою для адаптації до специфічних потреб певного проекту [7].

Аналіз моделі ISO/IEC 9126 показав що ця модель має довгу історію та широке використання, тому вона добре вивчена та зрозуміла. ISO/IEC 9126 може бути легко використана і впроваджена у різних проектах без значних зусиль [4]. Проте у порівнянні з більш сучасними моделями, ISO/IEC 9126 може бути обмеженою у визначенні деяких аспектів якості, таких як безпека та доступність. Оскільки ця модель була запропонована давно, вона може не враховувати сучасні тенденції та вимоги до якості ПС.

Розглядаючи модель Quality Attributes Workshop (QAW) варто зазначити, що модель дозволяє активно залучати зацікавлених сторін до визначення та оцінки характеристик якості, що допомагає забезпечити відповідність потребам клієнтів [11]. Окрім того QAW дозволяє адаптуватися до різних контекстів та особливостей проектів. Проте практична реалізація механізмів даної моделі залежність від активної участі стейкхолдерів, що може ускладнити процес або призвести до конфліктів інтересів. Також оцінки якості які здійснені на базі застосування приципів моделі QAW можуть бути суб'єктивними залежно від думок та переконань учасників робочої групи.

Розглядаючи модель Architecture Tradeoff Analysis Method (ATAM) доцільно відмітити її архітектурну спрямованість: модель дозволяє оцінювати вплив архітектури на характеристики якості, що дозволяє забезпечити їх відповідність до потреб бізнесу [10]. Також ATAM допомагає ідентифікувати та управляти ризиками, пов'язаними з архітектурними рішеннями. Проте використання ATAM може бути складним і часомовкладеним процесом, особливо для великих та складних проектів. Самаж реалізація ATAM на принциповому рівні потребує наявності висококваліфікованих експертів, що може бути викликом для організацій.

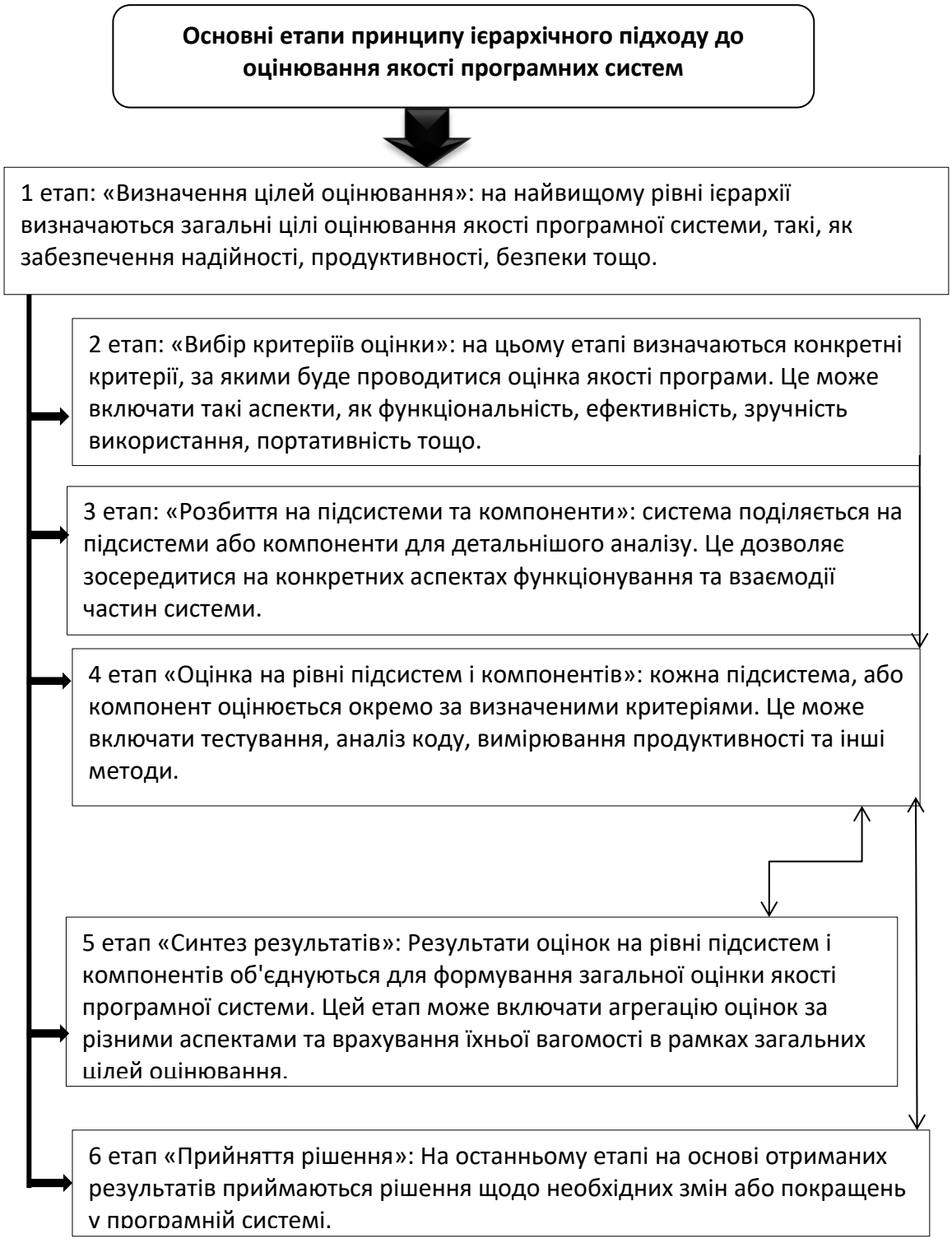

Рис.1. Основні етапи принципу ІП до оцінювання якості ПС

Розглядаючи модель Goal-Question-Metric (GQM) на принциповому рівні варто відмітити її об'єктивність: модель дозволяє формалізувати цілі та метрики для оцінки якості, що сприяє об'єктивності процесу [8]. Також GQM може бути адаптована до різних потреб та контекстів проектів [9].

Проте моделі оцінки якості ПС на базі GQM на принциповому рівні маю проблеми із складністю формалізації: Створення зв'язків між цілями, питаннями та метриками може бути складним завданням і вимагати додаткових зусиль. Також простежується проблема збільшення кількості метрик, яка може призвести до перевантаження та збільшення складності оцінки якості в межах ІП до оцінювання якості ПС.

На рис.1. наведено основні етапи принципу ІП до оцінювання якості ПС.

ІП дозволяє систематизувати процес оцінювання якості програмних систем та забезпечити більш глибокий та повний аналіз їхніх характеристик.

Для удосконаленні механізму аналізу особливостей деталізованого порівневого ієрархічного розбиття механізму оцінювання якості програмних систем доцільно додаткового застосувати цілий ряд спеціальних формул, зокрема формулу на основі якої буде реалізовуватися процедура загального оцінювання якості ПС $Q_{ієрарх.загал.}$ в межах застосування порівневого ієрархічного розбиття механізму оцінювання (2):

$$Q_{ієрарх.загал.} = \sum_{i=1}^{n} W_i \cdot S_i, \quad (2)$$

де $Q_{ієрарх.загал.}$ – загальна оцінка якості ПС в межах застосування порівневого ієрархічного розбиття механізму оцінювання; n – кількість рівнів якості; W_i – вага i-го рівня якості S_i – оцінка якості на i-му рівні розраховується у відповідності до виразу (3);

$$S_i = \sum_{j=1}^{m} W_{ij} \cdot P_{ii}, \quad (3)$$

де m – кількість підрівнів на кожному рівні; W_{ij} – вага j-го підрівня якості на i-му рівні; P_{ij} – оцінка якості j-го підрівня якості на i-му рівні.

Оцінка підрівня з особливостями $\left(P\right)$ розраховується у відповідності до виразу (4):

$$P = \frac{Q_{підрівня}}{Q_{\max підрівня}} \times 100\%, \quad (4)$$

де $Q_{підрівня}$ – оцінка якості ПС в межах підрівня; $Q_{\max підрівня}$ – максимально можливе значення оцінки якості ПС в межах підрівня.

Зважаючи на вище наведене формули (3) та (4) можуть бути використані для обчислення загальної оцінки якості ПС, оцінки кожного рівня якості, а також оцінки підрівня з урахуванням його особливостей.

Оцінка якості ПС з урахуванням особливостей (OS) реалізується згідно (5):

$$OS = \sum_{k=1}^{p} W_{ijk} \cdot P_{ijk}, \quad (5)$$

де OS – оцінка якості підрівня ПС з урахуванням особливостей; p – кількість особливостей на кожному підрівні; W_{ijk} – вага k-ї особливості на j-му підрівні на i-рівні; P_{ijk} – оцінка k-ї особливості на j-му підрівні на i-рівні.

Формули (12), (13) мають наступні зв'язки з принципами створення багато-цільових моделей якості:

– приділення ваги кожному аспекту якості: Вага кожного рівня визначає його значимість у загальному оцінюванні;

– врахування специфіки кожного рівня та підрівня якості: дозволяють врахувати унікальні аспекти кожного рівня при обчисленні загальної оцінки.

З технічно-практичної точки зору для того щоб провести більш деталізований аналіз та оцінку кожного елементу ієрархічної структури оцінювання якості ПС доцільно проводити додаткові оцінки, які будуть наведені нижче.

Оцінка загальної продуктивності ПС (PQ) реалізується згідно (6):

$$PQ = Відуговість \times 0.2 + Пропускназдатність \times 0.3 + Швидкодія \times 0.5, \quad (6)$$

де відгуковість – показник, який відображає швидкість відгуку системи на дії користувача; пропускна здатність показник, який вказує на кількість запитів або операцій, які система може обробити за одиницю часу; швидкодія показник визначає, який загальну швидкість виконання операцій, або завдань системою.

Таким чином формула (6) для визначення (PQ) дозволяє кількісно оцінити рівень продуктивності системи на основі важливих показників, таких, як відгуковість, пропускна здатність та швидкодія де на практиці користувачі можуть використовувати вище зазначену формулу для порівняння продуктивності різних програмних систем перед прийняттям рішення про вибір програмного забезпечення для своїх потреб; розробники програмного забезпечення використовуючи формулу (6) можуть здійснювати оцінки та порівняння різних версій свого продукту, щоб вдосконалювати його продуктивність. З допомогою застосування формули (6) менеджери проектів можуть використовувати цю формулу для моніторингу та управління продуктивністю програмних систем під час розробки та експлуатації. Отже запропонована формула (6) дозволяє кількісно оцінити та порівняти рівень продуктивності програмної системи на основі ключових показників, що є критичними для багатьох типів програмного забезпечення.

Оцінку надійності і безпеки ПС (RS) реалізується згідно (7):

$$RS = Відновлюваність \times 0.4 + Відмовостійкість \times 0.4 + Захищеність \times 0.2, \quad (7)$$

де відновлюваність – показник, який вказує на можливість системи відновлюватися після виникнення помилок або відмов; відмово стійкість – показник, який визначає стійкість системи до внутрішніх і зовнішніх впливів, таких, як помилки програмного забезпечення або кібератаки; захищеність – показник, який відображає рівень захищеності системи від несанкціонованого доступу та зловживань.

З технічної позиції запропонована формула (7) дозволяє оцінити рівень надійності і безпеки програмної системи на основі ключових показників, які є важливими для забезпечення стабільності та безпеки в різних умовах експлуатації. Застосування на практиці вище зазначеної формули (17) дозволяє об'єктивно оцінити безпеку, враховуючи різні аспекти безпеки з відповідними вагами: розробники та архітектори систем можуть використовувати цю формулу для оцінки та порівняння рівня надійності і безпеки різних версій або конфігурацій системи; аудитори та експерти з безпеки можуть використовувати цю формулу для оцінки безпеки системи та виявлення слабких місць; менеджери проектів можуть використовувати цю формулу для визначення пріоритетів та планування заходів щодо підвищення рівня надійності і безпеки програмної системи.

Оцінку ефективності ресурсів (ES), яка реалізується згідно (8):

$$ES = Використання\ пам'яті \times 0.5 + Використання\ процесорного\ часу \times 0.5, \quad (8)$$

де використання пам'яті – показник, який вказує на ефективність використання оперативної пам'яті системою; використання процесорного часу – показник що визначає, наскільки ефективно система використовує обчислювальні ресурси.

З технічної позиції запропонована формула (8) дозволяє на практиці адміністраторам систем використовувати цю формулу для оцінки рівня ефективності використання ресурсів та виявлення потенційних проблем з їх недостатнім або неправильним використанням;

розробникам програмного забезпечення використовувати цю формулу для тестування та оптимізації продуктів під кутом оптимального використання ресурсів; менеджерам проектів використовувати цю формулу для моніторингу ресурсів, необхідних для різних аспектів роботи програмної системи та її підсистем.

Отже практична цінність запропонованої формули (8) зводиться до того, що вона дозволяє кількісно оцінити рівень ефективності використання ресурсів програмної системи на основі ключових показників, що є важливими для забезпечення оптимальної працездатності та продуктивності системи.

Оцінка зручності користування (US) реалізується згідно (9):

$$US = Інтуїтивність \times 0.6 + Документація \times 0.4, \quad (9)$$

де інтуїтивність – показник, який відображає, наскільки легко користувачі можуть розуміти та взаємодіяти з системою без попереднього навчання; документація – показник. який вказує на наявність та якість документації, яка допомагає користувачам зрозуміти функції та особливості системи.

Застосування формули (9) на практиці дозволяє користувачам використовувати цю формулу для порівняння зручності користування різними програмними продуктами перед прийняттям рішення про вибір. Натомість розробники ПС можуть використовувати формулу (9) для оцінки та вдосконалення інтерфейсу користувача та документації для поліпшення зручності використання системи, а менеджери проектів можуть використовувати цю формулу для визначення пріоритетів та планування заходів щодо підвищення рівня зручності користування програмною системою.

Отже практична цінність запропонованої формули (9) зводиться до того, що вона дозволяє кількісно оцінити рівень зручності користування програмною системою на основі ключових показників, що є важливими для забезпечення задоволення та продуктивності користувачів.

Оцінку сумісності з іншими системами (CS), доцільно реалізувати згідно формули (10):

$$CS = Сумісність\ з\ мовами\ програмування \times 0.5 + Інтеграція\ з\ іншими\ ПС \times 0.5, \quad (10)$$

де сумісність з мовами програмування – показник, який відображає наскільки легко програмна система може взаємодіяти з різними мовами програмування та технологіями; інтеграція з іншими програмами – показник, який вказує на здатність програми взаємодіяти з іншими програмними продуктами та сервісами.

Застосування формули (10) на практиці дозволяє розробникам ПС використовувати цю формулу для оцінки та планування сумісності своєї програмної системи з різними мовами програмування та програмними інтерфейсами; архітектори систем можуть використовувати цю формулу для аналізу впливу інтеграції з іншими програмними системами на загальну архітектуру та ефективність роботи системи; менеджери проектів можуть використовувати цю формулу для оцінки ризиків та планування робіт з інтеграції з іншими програмними системами.

В загальному аспекті на принциповому рівні формула (10) дозволяє кількісно оцінити рівень сумісності ПС з іншими системами на основі ключових показників, що є важливими для забезпечення успішної інтеграції та взаємодії з іншими компонентами програмного середовища.

Висновки

1. Проведений аналіз особливостей ієрархічного розбиття механізму оцінювання якості ПС дозволив встановити, що ієрархічне розбиття дозволяє систематизувати аспекти якості на різних рівнях абстракції, але не завжди ефективно враховує всі деталі та взаємозв'язки між ними. Для оптимального використання цього підходу важливо ретельно розробити методику розбиття на рівні та визначити зв'язки між ними для комплексної оцінки якості.

2. Проведений аналіз застосування математичних теорій в формуванні ієрархічної структури моделей якості ПС дозволив встановити, що для ефективного формування

ієрархічної структури моделей якості можуть бути застосовані різноманітні математичні теорії, такі як теорія графів, теорія ймовірностей, теорія систем тощо. Проте необхідно уважно вибирати та адаптувати ці теорії до конкретних потреб та характеристик програмних систем.

3. В ході розгляду основних етапів принципу ієрархічного підходу до оцінювання якості ПС було встановлено, що етапи використання ієрархічного підходу включають побудову ієрархії якості, визначення метрик на кожному рівні, збір та аналіз даних, оцінку та виправлення помилок. Важливо розробити чіткі методики для кожного етапу з метою забезпечення консистентності та точності оцінки.

4. В межах розробки математичного апарату для моделей на основі ієрархічного розбиття механізму оцінювання якості ПС: розроблено математичний інструментарій, який включає формульний математичний апарат для оцінювання якості ПС. Важливо підкреслити потребу у додаткових дослідженнях для вдосконалення цього математичного апарату та його відповідності конкретним вимогам та умовам використання на практиці.

Список використаних джерел

1. Azar, D., Harmanani, H., & Korkmaz, R. (2009). A hybrid heuristic approach to optimize rule-based software quality estimation models. *Information and Software Technology,* 1365–1376.

2. Bharathi, R., & Selvarani, R. (2020). Hidden Markov model approach for software reliability estimation with logic error. International Journal of Automation and Computing, 17, 305–320.

3. Foidl, H., & Felderer, M. (2018). Integrating software quality models into risk-based testing. *Software Quality Journal*, 26, 809–847.

4. Gordieiev, O., Kharchenko, V., Fominykh, N., & Sklyar, V. (2014). Evolution of Software Quality Models in Context of the Standard ISO 25010. *In The Ninth International Conference DepCoS-RELCOMEX: Proceedings* (pp. 223–232). Wroclaw, Poland.

5. Helander M.E. Planning Models for Software Reliability and Cost/ M.E. Helander,M. Zhao,N. *Ohlsson //IEEE Trans. Softw. Eng.* – 1998. –V. 24. – N. 6.– P. 420 – 434.

6. Helander, M. E. (1998). Planning Models for Software Reliability and Cost. *IEEE Transactions on Software Engineering, 24(6),* 420–434.

7. Kapur, P. K., Pham, H., Anand, S., & Yadav, K. (2011). A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. *IEEE Transactions on Reliability, 60(1),* 331–340.

8. Kemerer, C. F., & Paulk, M. C. (2009). The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP *Data. IEEE Transactions on Software Engineering, 35(4),* 534–550.

9. Lee, M. (Year). Software quality factors and software quality metrics to enhance software quality assurance. *Current Journal of Applied Science and Technology, 4(21),* 3069–3075.

10. Letichevsky, A., Kapitonova, J., Letichevsky Jr., A., Volkov, V., Baranov, S., & Kotlyarov, V. (2005). Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications. *In ISSRE 2004, WITUL, Rennes*, 4 (pp. 112–142).

11. Musa, J. D. (1993). Operational Profiles in Software Reliability Engineering. *IEEE Software, 10(2),* 14–32.

12. Ohlsson, N., Helander, M., & Wohlin, C. (1996). Quality Improvement by Identification of Fault-Prone Modules using Software Design Metrics. *In Proceedings Sixth International Conference on Software Quality* (pp. 1–13).

13. Sahu, K., & Srivastava, R. K. (2021). Predicting software bugs of newly and large datasets through a unified neuro-fuzzy approach: Reliability perspective. *Advances in Mathematics: Scientific Journal, 10(1),* 543–555.

14. Yamming, C., & Shiyi, X. (2007). Exploration of complexity in software reliability. *Tsinghua Science & Technology, 1(2),* 266–269.