

Шантир А.С., к.т.н., Зінченко В.В.,
Єльченко С.В., к.т.н., Кравчук П.О., PhD.

ЗАСАДИ УДОСКОНАЛЕННЯ МОДЕЛЕЙ ОЦІНКИ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ НА ПРИКЛАДІ МОДЕЛЕЙ СОСОМО ТА ISO 9126/25010

Shantyr A.S., Zinchenko V.V., Yelchenko S.V., Kravchuk P.O. The principles of improving models for assessing the quality of software systems using the cocomo models and iso 9126/25010.

The article is dedicated to the theoretical and practical analysis of improving models for assessing the quality of software systems, using COCOMO and ISO 9126/25010 models as examples. It is noted that achieving the set goal involves addressing tasks such as identifying shortcomings at the level of mathematical representation of these models, developing mathematical expressions for cost and adjustment models, as well as describing approaches to mathematical enhancement of COCOMO and ISO 9126/25010 models. The need for such mathematical expressions is justified by their potential advantages, such as increased accuracy of assessment, the ability to forecast project costs and quality, standardization of the assessment process, and flexibility in adaptation to various situations. Developing mathematical expressions for COCOMO and ISO 9126/25010 models can significantly enhance their effectiveness and applicability in practical usage for assessing the quality of software systems. The analysis conducted suggests that developing mathematical expressions for software quality assessment models reflects the need for further refinement and development of assessment methods to improve objectivity, accuracy, and applicability in practical conditions. The generalized conclusion at the modeling level is that creating multi-purpose models for software quality is a key step in ensuring their successful development, implementation, and operation. These principles not only help evaluate various aspects of software quality, such as functionality, performance, reliability, efficiency, usability, and security, but also provide a systematic approach to development, taking into account diverse requirements and stakeholder expectations. Creating such models involves the use of mathematical formulas, algorithms, and methods that objectively assess the quality level of the system, integrating various aspects and considering specific usage contexts. Successful implementation of these principles allows enhancing the efficiency of software development, reducing risks and costs, improving collaboration among project stakeholders, and ensuring high quality and user satisfaction with the product.

Keywords: software, quality metrics, user needs, information technology, mathematical framework, quality assessment models, software systems.

Шантир А.С., Зінченко В.В., Єльченко С.В., Кравчук П.О. Засади удосконалення моделей оцінки якості програмних систем на прикладі моделей cocomo та Iso 9126/25010. Стаття присвячена теоретичному та практичному аналізу покращення моделей оцінки якості програмних систем, використовуючи моделі COCOMO та ISO 9126/25010 як приклади. Зазначається, що реалізація поставленої мети включає вирішення таких завдань, як розгляд недоліків на рівні математичного представлення цих моделей, розробка математичного виразу для моделей витрат і коригувань, а також опис підходів до математичного удосконалення моделей COCOMO та ISO 9126/25010. Потреба у такому математичному виразі обґрунтовується його потенційними перевагами, такими як підвищення точності оцінки, можливість прогнозування витрат і якості проєктів, уніфікація процесу оцінки та гнучкість в адаптації до різних ситуацій. Розробка математичного виразу для моделей COCOMO та ISO 9126/25010 може значно підвищити їхню ефективність та застосовність у практичному використанні при оцінці якості програмних систем. В ході проведеного аналізу було встановлено, що розробка математичних виразів для моделей оцінки якості програмних систем відображає потребу у подальшому вдосконаленні та розвитку методів оцінки з метою підвищення об'єктивності, точності та застосовності в практичних умовах. Узагальнений висновок на рівні моделювання полягає в тому, що створення багатоцільових моделей якості програмних систем є ключовим етапом у забезпеченні їхньої успішної розробки, впровадження та експлуатації. Ці принципи не лише допомагають оцінити різні аспекти якості програмного забезпечення, такі як функціональність, продуктивність, надійність, ефективність, зручність використання та безпека, але й забезпечують системний підхід до розробки, що враховує різноманітні вимоги та очікування зацікавлених сторін.

Створення таких моделей передбачає використання математичних формул, алгоритмів та методів, які дозволяють об'єктивно оцінити рівень якості системи, інтегруючи різноманітні аспекти та враховуючи специфічний контекст використання. Успішна реалізація цих принципів дозволяє підвищити ефективність розробки програмного забезпечення, зменшити ризики та витрати, покращити співпрацю між різними сторонами проекту та забезпечити високу якість та задоволення від продукту в кінцевих користувачів.

Ключові слова: програмне забезпечення, метрики якості, потреби користувачів, інформаційні технології, математичний апарат, моделі оцінки якості, програмні системи.

Вступ

Розробка математичних виразів для моделей оцінки якості програмних систем (ПС), які засновані на поєднанні принципів моделі COCOMO (Constructive Cost Model) і стандарту ISO 9126/25010 спрямована на об'єктивну оцінку різних аспектів якості програмного забезпечення та забезпечення їхньої успішної розробки та впровадження. Проте, існують деякі області, де можна подальше удосконалення і розвиток цих методів.

Одним з напрямків удосконалення є розширення моделей якості для врахування нових технологій та платформ. Швидкі зміни в технологічному середовищі потребують постійного оновлення методів оцінки, щоб вони залишалися актуальними та ефективними.

Крім того, можливе вдосконалення методів урахування користувацьких вимог та очікувань. Розвиток інтерфейсів та підвищення уваги до зручності використання вимагає постійного апгрейду моделей якості з урахуванням цих аспектів.

Також важливим є подальше розширення метрик якості та вдосконалення їхнього використання. Забезпечення комплексного покриття різних аспектів якості, таких як надійність, продуктивність, безпека тощо, дозволить отримати більш повне уявлення про якість програмного забезпечення.

Нарешті, важливо продовжувати розвивати методи врахування контексту використання програмного забезпечення. Розуміння та врахування потреб та специфіки різних галузей та сегментів користувачів дозволить створити більш адаптивні та ефективні моделі оцінки якості.

Узагальнюючи, подальший розвиток та удосконалення моделей оцінки якості ПС базується на поєднанні математичних підходів моделі COCOMO та стандарту ISO 9126/25010 з огляду на постійні зміни у технологічному середовищі та зростаючі потреби користувачів. Це допоможе підвищити об'єктивність, точність та застосовність оцінки якості ПС в практичних умовах, що в свою чергу сприятиме покращенню ефективності розробки програмного забезпечення та задоволенню користувачів від кінцевого продукту.

Постановка завдання. Потреба у розробці математичного вираження моделей COCOMO та ISO 9126/25010 в межах оцінки якості ПС виражається у кількох аспектах:

– Точність оцінки: Математичне вираження дозволяє зробити процес оцінки більш точним і об'єктивним. Замість суб'єктивної оцінки експертів або використання історичних даних, математичні моделі можуть враховувати більш широкий набір факторів і зв'язків між ними.

– Прогностичні можливості: Математичні моделі можуть бути використані для прогнозування витрат, термінів, або якості програмних проектів на основі конкретних вхідних параметрів. Це дозволяє здійснювати планування і управління проектами більш ефективно.

– Уніфікація: Математичні моделі надають можливість уніфікувати процес оцінки якості програмних проектів, що полегшує порівняння між різними проектами та використання стандартних методів оцінки.

– Гнучкість та адаптабельність: Математичні моделі можуть бути модифіковані та адаптовані для врахування конкретних особливостей проектів або контексту використання, що дозволяє використовувати їх в різних ситуаціях.

Отже, розробка математичного виразу для моделей СОСОМО та ISO 9126/25010 може значно підвищити їхню ефективність та застосовність у практичному використанні при розробці нових моделей оцінки якості ПС.

Аналіз останніх досліджень. В праці [1] висвітлено аспекти об'єктивності в оцінці якості ПС: цей погляд акцентує на тому, що підходи, як СОСОМО, так і ISO 9126/25010, мають свої обмеження у визначенні об'єктивної якості ПС. Наприклад, СОСОМО в основному базується на історичних даних та експертній думці, тоді, як ISO 9126/25010 пропонує більш систематичний підхід. Однак обидва підходи можуть бути підвержені упередженням та недостатній об'єктивності, оскільки оцінка залишається в основному суб'єктивною.

В праці [2] відмічається, що СОСОМО (Constructive Cost Model) базується на історичних даних і експертній думці, що може призвести до упередження або необ'єктивності в оцінці. Такий підхід може бути особливо непрозорим, якщо експерти мають різні погляди або якщо історичні дані не відображають реальну ситуацію. Натомість в праці [3] вказується, що стандарти ISO 9126/25010 намагаються надати більш систематичний підхід до оцінки якості ПС, використовуючи структурований набір критеріїв. Однак, навіть з таким підходом, можуть виникати проблеми з об'єктивністю, оскільки оцінка якості все ще може бути суб'єктивною і залежати від того, яким чином кожен критерій і його важливість інтерпретуються.

Проте у відповідності до [4] знання про те, як СОСОМО та стандарти ISO 9126/25010 працюють, допомагає командам вибрати найбільш підходящий метод оцінки якості залежно від контексту проекту, доступної інформації та інших факторів.

В праці [5] аналізуються засади інтеграції метрик якості в процес розробки програмного забезпечення: Цей погляд стверджує, що моделі якості, такі як ISO 9126/25010, допомагають в інтеграції метрик якості в усі етапи процесу розробки програмного забезпечення. Це дозволяє забезпечити відповідність вимогам до якості протягом усього життєвого циклу розробки програми.

Згідно з аналізом, який висвітлено в праці [6] досить важливо зосередитися саме на удосконаленні методів прогнозування вартості та трудомісткості розробки ПЗ: СОСОМО, зокрема, використовується для прогнозування вартості та трудомісткості проектів. Проте він може бути недостатньо гнучким у врахуванні різних факторів, які впливають на розробку програмного забезпечення. У контексті розширення моделі СОСОМО з точки зору оцінки якості може бути корисною для більш точних прогнозів.

У відповідності до [7] стандартизація процесу оцінки якості ПЗ: ISO 9126/25010 надає стандартну базу для оцінки якості ПС. Це дозволяє забезпечити єдність підходів до оцінки якості і сприяє порівнянню між різними проектами.

На основі аналізу праць [1–14] можна відмітити, що недоліками існуючих моделей, таких як СОСОМО та ISO 9126/25010, є їхні обмеження у врахуванні різноманітних аспектів якості. Тому виникає потреба в розробці альтернативних моделей, які б краще враховували специфіку проектів і були більш адаптованими до сучасних методів розробки ПЗ.

Автори праці [15] відмічають про важливість використання агільних методів в оцінці якості ПЗ: Агільні методи розробки ПЗ ставлять більший акцент на зміну вимог і швидкі ітерації. Однак існуючі моделі оцінки якості, як СОСОМО та ISO 9126/25010, можуть бути менш ефективними в контексті агільних проектів через їхню статичність та важкість адаптації до постійних змін.

В праці [16] аналізуються особливості використання машинного навчання для покращення моделей оцінки якості ПЗ: Машинне навчання може бути використане для аналізу великих обсягів даних та ідентифікації складних залежностей між параметрами розробки та якістю продукту. Це може допомогти покращити якість та точність оцінки якості ПЗ.

Відповідно праці [17] нині існує нагальна потреба в інтеграції стандартів та моделей: існують виклики у взаємній сумісності та інтеграції різних стандартів та моделей оцінки якості, які ускладнюють процес прийняття рішень та порівняння різних продуктів.

Автори праці [18] відмічають важливість розробки гібридних моделей оцінки ПЗ: гібридні моделі комбінують підходи різних моделей, таких як COCOMO та ISO 9126/25010, для отримання більш точних та повних оцінок якості ПЗ. Згідно аналізу авторів вище наведеної праці розробка гібридних моделей оцінки якості ПЗ є дуже перспективним напрямком у дослідженнях і практичних застосуваннях. Такі гібридні моделі можуть включати в себе не лише COCOMO та ISO 9126/25010, а й інші існуючі моделі, методи аналізу даних, машинне навчання та інші техніки. Головна перевага гібридних моделей полягає в тому, що вони поєднують у собі переваги кожного використовуваного підходу, компенсуючи їхні недоліки. Наприклад, можна використовувати історичні дані COCOMO для побудови початкової оцінки вартості розробки, а потім додати аналіз якості за критеріями ISO 9126/25010 для отримання більш повного зображення загальної якості ПЗ. Крім того, гібридні моделі можуть бути адаптовані до конкретних потреб та характеристик проекту, що робить їх більш гнучкими та ефективними в реальних умовах розробки ПЗ. Узагальнюючи, розвиток гібридних моделей оцінки якості ПЗ може виявитися ключовим для покращення точності та об'єктивності оцінки якості ПС у майбутньому.

В праці [19] відмічається важливість врахування ролі культури якості в процесі розробки ПЗ: Культура якості, яка відображається в цінностях та підходах команди розробників, може впливати на ефективність використання моделей оцінки якості. Недоліки у культурі якості можуть призвести до неправильного використання або ігнорування моделей.

В праці [20] акцентується увага на потребі в оновленні та адаптації існуючих моделей до змін у технологічному середовищі: Існуючі моделі, такі як COCOMO та ISO 9126/25010, можуть бути застарілими або неадаптованими до нових технологій та підходів у розробці ПЗ. Їхнє оновлення та адаптація є важливими для забезпечення актуальності та ефективності оцінки якості ПЗ.

Також варто відмітити, що в працях [1–20] відмічається важливість врахування особливостей конкретної галузі при оцінці якості ПЗ: Деякі галузі можуть мати унікальні вимоги до якості ПЗ, які не завжди можуть бути адекватно враховані існуючими моделями. Тому важливо розробляти або адаптувати моделі з урахуванням специфіки конкретної галузі. В праці [21] зазначаються сучасні тенденції в області створення ефективних механізмів зворотного зв'язку: Для постійного вдосконалення моделей оцінки якості ПЗ необхідно мати ефективні механізми зворотного зв'язку, які дозволяють отримувати оцінки, порівнювати їх з фактичною якістю продукту та вносити необхідні корективи.

Автори праць [4,7] підкреслюють практичну важливість узгодження оцінки якості з потребами користувачів: Оцінка якості ПЗ повинна бути здійснена з урахуванням потреб та очікувань користувачів. Моделі, такі як ISO 9126/25010, допомагають стандартизувати цей процес та забезпечують зв'язок між технічними характеристиками ПЗ та користувацькими очікуваннями.

Таким чином на основі вище наведено виникає потреба в необхідності розробки рішень щодо математичного вираження моделей COCOMO та ISO 9126/25010.

Метою статті є теоретично-практичний розгляд засад удосконалення моделей оцінки якості ПС на прикладі моделей COCOMO та ISO 9126/25010.

Реалізація поставленої мети передбачає вирішення наступних цілей:

1. Проведення розгляду недоліків на рівні математичного представлення моделей оцінки якості ПС на прикладі моделей COCOMO та ISO 9126/25010;
2. Розробки математичного вираження моделі витрат і коригувань (COCOMO) та моделі ISO 9126/25010;
3. Опису підходів щодо математичного удосконалення моделей COCOMO та ISO 9126/25010.

Виклад основного матеріалу дослідження.

Розглянемо недоліки на рівні математичного представлення моделей оцінки якості ПС на прикладі моделей COSOMO та ISO 9126/25010.

Згідно з [2] в моделі COSOMO є недолік який пов'язаний з спрощеною лінійною функцією вартості: базується на лінійній функції вартості, яка може бути недостатньою для адекватного врахування складності та різноманітності програмних проектів. В праці [3] відмічається, що параметри моделі COSOMO є статичними і не враховують можливих змін у характеристиках проекту під час його розробки, що може призвести до неточностей у прогнозуванні вартості та тривалості проекту.

В працях [4–6] вказується на існування недоліку який пов'язаний із суб'єктивністю визначення метрик якості: Деякі метрики якості, визначені в стандарті ISO 9126/25010, можуть бути суб'єктивними і піддаються інтерпретації, що ускладнює їхнє точне вимірювання та порівняння між різними проектами.

В праці [7] акцентується на проблемі обмеженості області застосування: деякі аспекти якості програмного забезпечення можуть бути складно формалізовані та виміряні за допомогою існуючих метрик, що обмежує застосування стандарту ISO 9126/25010 у деяких ситуаціях.

Щоб покращити ці моделі, можна розглянути наступні аспекти:

- Додаткові параметри для моделі COSOMO: Розширення моделі COSOMO за допомогою додаткових параметрів, які враховують складність проекту та його специфікації;

- Об'єктивніші метрики для ISO 9126/25010: Розробка об'єктивних метрик якості, які були б менш суб'єктивними та легше вимірюваними, для забезпечення більш точної та об'єктивної оцінки якості програмного забезпечення;

- Врахування динамічних змін в параметрах: Розробка моделей, які враховують динамічні зміни в характеристиках проекту та здатні адаптуватися до них для більш точного прогнозування вартості та тривалості розробки програмного забезпечення.

Для математичного вираження моделі витрат і коригувань (COSOMO), її можна представити у вигляді емпіричних рівнянь, які описують залежність між різними параметрами проекту та зусиллями/вартістю розробки програмного забезпечення. Основні компоненти моделі COSOMO включаються у такі параметри:

- Розмір програми *Size*: визначається кількістю рядків коду, або іншими метриками обсягу програмного продукту;

- Складність продукту *Complexity*: виражається через рівень складності програми, що враховується наявністю алгоритмічних складнощів, вимог до надійності, рівня технологічної складності тощо;

- Тип проекту *Type*: визначається відповідно до характеристик самого проекту: організаційні аспекти, середовище розробки, досвід розробників тощо;

- Фактори коригування *AdjustmentFactors*: враховуються додаткові фактори, що впливають на процес розробки, такі, як вміння команди розробників, необхідність забезпечення якості, технологічна складність тощо.

Математично модель COSOMO можна виразити у вигляді рівняння (1):

$$Effort = A \times Size^B \times \prod_{i=1}^n EM_i, \quad (1)$$

Де *Effort* – зусилля (час, люди-місяці) на розробку проекту; *Size* – розмір програми (наприклад, кількість рядків коду); *A* та *B* – коефіцієнти, що залежать від типу проекту і складності; *EM_i* – фактори коригування.

Недоліки даної моделі, зокрема, обмежена точність прогнозів через спрощені розрахункові підходи та велику залежність від хронології даних, можна врахувати за

допомогою введення додаткових коефіцієнтів адаптації, або покращенням методології збору та аналізу даних.

Продовжуючи розгляд математичних аспектів моделі витрат і коригувань СОСОМО, додамо, що у цій моделі також можуть бути визначені коефіцієнти \bar{A} і \bar{B} для кожного типу проекту. Ці коефіцієнти визначаються на підставі хронологічної обробки даних та експертних оцінок і відображають вплив різних факторів на загальне зусилля розробки.

Далі, для кожного з факторів коригування EM_i , таких як досвід розробника, вміння команди, технічна складність і т. д., можна встановити коефіцієнти впливу на загальні зусилля. Наприклад, якщо вміння команди вважаються дуже важливими, коефіцієнт для цього фактору може бути встановлений на високому рівні.

Отже, у формулі СОСОМО враховуються кілька важливих параметрів, які спільно впливають на зусилля розробки. Математично вони моделюються у вигляді множника, який враховує розмір програми, та додаткових коефіцієнтів, що відображають вплив різних факторів на загальні зусилля.

Розглянемо узагальнені фактори коригування моделі СОСОМО для оцінки якості ПС в межах принципів створення багатоцільових моделей якості (Multiple Quality Models): Зазвичай для оцінки якості програмного забезпечення використовуються фактори якості, такі як ефективність, надійність, зручність в експлуатації, підтримка та інші.

З факторів коригування моделі СОСОМО, які можуть бути використані при оцінці якості ПС в межах окреслених принципів створення багатоцільових моделей якості можуть бути:

- розмір продукту: враховує обсяг програмного забезпечення, яке потрібно розробити або підтримувати. Чим більший обсяг, тим складніше забезпечити його якість [11];
- складність продукту: оцінюється на основі складності функцій, алгоритмів, інтерфейсів тощо. Складніші продукти потребують більшої уваги до якості [15];
- тривалість робочого часу: визначається кількістю людських годин, які потрібні для розробки або підтримки продукту. Довша тривалість може привести до більшої потреби у вдосконаленні якості [9];
- технічне середовище: враховує рівень складності технічного середовища, у якому виконується розробка. Наприклад, використання новітніх технологій може вимагати більшої уваги до якості [10];
- досвід команди: враховує рівень досвіду та компетентності команди розробників. Досвідчені команди можуть краще забезпечити високу якість продукту [18];
- вимоги до надійності та безпеки: цей фактор визначає, наскільки критичною є надійність і безпека продукту. Більш високі вимоги можуть вимагати додаткових зусиль щодо забезпечення якості [8].

Зазначені узагальнені фактори коригування можуть бути використані для модифікації базових оцінок моделі СОСОМО з урахуванням особливостей проекту та його вимог до якості.

Для подальшого удосконалення моделі СОСОМО можна розглянути додавання додаткових факторів коригування та уточнення коефіцієнтів, які враховують вплив різних аспектів на зусилля розробки програмного забезпечення.

В праці [9] для вдосконалення моделі СОСОМО пропонується застосувати врахування організаційної складності: додавання факторів, що враховують рівень складності організаційних процесів та особливості в межах співпраці команди розробників, таких як комунікація, управління проектом, ступінь структурованості організації.

В праці [11] акцентують увагу на потребі уточнення факторів технічної складності моделі СОСОМО: розгляд використання нових технологій, архітектурних патернів, інструментів розробки, що може впливати на загальну складність проекту.

В праці [16] відзначають про доцільність розгляду адаптивності та гнучкості в межах вдосконалення моделі СОСОМО: уточнення моделі для врахування гнучкості та можливостей

адаптації проекту до змін у вимогах, технологічних аспектах або експлуатаційному середовищі.

В праці [20] для вдосконалення моделі СОСОМО пропонують зосередитися на комплексній деталізації аналізу ризиків пов'язаних із практичною розробкою та експлуатацією ПС: врахування більш деталізованого аналізу ризиків, що може впливати на зусилля розробки та вартість проекту.

В свою чергу варто відмітити, що вдосконалення моделі СОСОМО за допомогою вищезазначених підходів може допомогти забезпечити більш точні та релевантні прогнози щодо зусиль та витрат на розробку програмного забезпечення.

Уточнимо математичний апарат моделі СОСОМО, додавши нові фактори та удосконаливши існуючі. Відповідно [19] уточнення моделі СОСОМО може включати додавання нових факторів коригування її застосування. В нашому випадку будуть застосовані фактори коригування EM_i , які враховують додаткові аспекти, такі як культурна складність C , технічна складність T , екологічні аспекти E та гнучкість F . Тоді на математичному рівні нова удосконалена формула моделі СОСОМО може мати вигляд (2):

$$Effort = A \times Size^B \times EM_1 \times EM_2 \times \dots \times EM_n \times EM_C \times EM_T \times EM_E \times EM_F, \quad (2)$$

де кожен з факторів коригування EM_i визначається відповідними коефіцієнтами, які враховують вплив відповідного аспекту на зусилля розробки; $Effort$ – зусилля (час, люди-місяці) на розробку проекту, це величина, яка визначає кількість зусиль, необхідних для розробки програмного продукту. Вище указана формула визначає загальне зусилля на основі розміру продукту та факторів коригування; $Size$ – це метрика, яка вимірює розмір програмного продукту. Вона може бути виражена, наприклад, у рядках коду, функціональних точках, або інших метриках розміру; A та B – коефіцієнти, що залежать від типу проекту і складності: коефіцієнт A відображає базові зусилля, а коефіцієнт B відображає ступінь, в якій зусилля змінюються з розміром програмного продукту; EM_i – фактори коригування: EM_1 – фактор коригування, який відображає врахування впливу досвіду розробника; EM_2 – фактор коригування, який відображає врахування впливу технічної складності; EM_C – фактор коригування, який відображає врахування впливу культурної складності тощо.

Кожен із зазначених вище компонентів удосконаленої моделі СОСОМО враховується у загальній формулі для оцінки зусиль, необхідних для розробки програмного продукту. Математичне моделювання таких факторів дозволяє враховувати різні аспекти, які впливають на розробку, і підвищує точність прогнозування зусиль.

Математично удосконалена модель СОСОМО дозволить більш точно прогнозувати зусилля та витрати на розробку програмного забезпечення, враховуючи більший спектр аспектів, що впливають на проект.

Згідно [10] математичні позиції моделі оцінки якості СОСОМО доцільно розглядати на рівні моделювання технічних аспектів. В даному разі фактори коригування розглядаються, як множники, що впливають на загальні зусилля. Їх вплив може бути виражений у вигляді математичних формул, які враховують важливість кожного фактора та його взаємозв'язок з іншими аспектами проекту. Можна також врахувати функцію помилок та дисперсії для кожного аспекту моделі, щоб оцінити точність прогнозування та визначити рівень невизначеності [11]. Математично це може бути виражено через функцію відхилення та середньоквадратичне відхилення. В удосконаленій моделі удосконалена модель СОСОМО ризики можуть бути враховані за допомогою ймовірнісних розподілів та статистичних методів, що дозволяють визначити ймовірність виникнення подій та їх вплив на зусилля розробки. Усі ці математичні позиції дозволяють створювати більш точні та надійні моделі для прогнозування зусиль та витрат на розробку програмного забезпечення.

Щоб уточнити математичний апарат моделі якості ISO 9126/25010, розглянемо кожен аспект окремо:

Атрибути якості: в моделі ISO 9126/25010 атрибути якості можуть бути виражені як Q_i , де i – індекс атрибуту якості. Наприклад, Q_1 може відповідати функціональності, Q_2 – надійності, і так далі.

Підатрибути якості: Кожен атрибут якості може мати підатрибути, що визначають конкретні аспекти цього атрибуту. Нехай Q_{ij} буде підатрибутом Q_i , де j – індекс підатрибуту.

Метрики якості: для кожного підатрибуту можна визначити відповідні метрики якості. Нехай M_{ijk} буде метрикою для Q_{ij} , де k – індекс метрики якості.

Відповідно[12] якість програмного продукту може бути оцінена, як сукупність значень метрик для всіх підатрибутів і атрибутів якості. Математично це може бути виражено, як сума значень метрик з урахуванням їх важливості (3):

$$Quality = \sum_{i=1}^n \sum_{j=1}^{m_i} \sum_{k=1}^{p_{ij}} w_{ijk} \times M_{ijk}, \quad (4)$$

де n – кількість атрибутів якості, m_i – кількість підатрибутів для Q_i , p_{ij} – кількість метрик якості для Q_{ij} , w_{ijk} – вага M_{ijk} .

Ваги w_{ijk} визначаються на основі важливості кожної метрики для кінцевої оцінки якості.

Вони можуть бути визначені експертно, або шляхом аналізу впливу кожної метрики на загальну якість програмного продукту.

Такий математичний апарат дозволяє систематично оцінювати якість програмного продукту, враховуючи його різноманітні аспекти та атрибути.

Щоб удосконалити та оптимізувати математичний апарат моделі якості ISO 9126/25010, необхідно вирішити завдання, щодо уточнення метрик якості: ретельний аналіз і вибір оптимальних метрик для кожного підатрибуту та атрибуту якості. Метрики повинні бути максимально об'єктивними та релевантними для конкретного типу програмного продукту.

Після чого доцільно звернути увагу на адаптація ваг метрик: ваги w_{ijk} можна визначити динамічно в залежності від контексту проекту та вимог замовника, що в свою чергу дозволить краще враховувати особливості кожного конкретного проекту та покращити точність оцінки якості.

Для вирішення завдань, щодо врахування інтеракцій між метриками у моделі можна додати компонент, який враховує взаємодію між різними метриками. Це допоможе уникнути викривлення оцінки якості через недоліки адитивної моделі.

Також згідно з [14] доцільно застосувати використання алгоритмів машинного навчання, що дозволить забезпечити автоматичне визначення оптимальних ваг метрик та побудови більш точних моделей якості на основі історичних даних проектів.

У відповідності до [15] необхідно також забезпечити управління невизначеністю: важливо враховувати невизначеність та ризики при оцінці якості. В свою чергу відмітимо, що математичні моделі можуть включати методи обробки невизначеності, такі як використання імовірнісних розподілів та статистичних методів.

В межах реалізації механізму оцінювання якості ПС також необхідно забезпечити розробку інструментів підтримки рішень: створення програмних інструментів, які допомагатимуть автоматизувати процес оцінки якості та робити прогнози на основі математичних моделей.

Усі вище наведені заходи спрямовані на покращення точності, ефективності та застосовності моделі якості ISO 9126/25010, що дозволить краще управляти якістю програмного продукту та досягати кращих результатів у розробці ПЗ.

Врахування вище наведеного може бути здійснено шляхом визначення кожного атрибуту якості та його підатрибутів, а також відповідних метрик якості та їх ваг. Наприклад, якщо ми розглядаємо атрибут «Функціональність», ми можемо мати підатрибути, такі як «Точність», «Цілісність», «Зручність» тощо. Для кожного з цих підатрибутів ми визначаємо відповідні метрики якості і їх ваги.

Наприклад, якщо для підатрибути «Точність» ми використовуємо метрику «Кількість виявлених дефектів», то формула може виглядати наступним чином(5):

$$Quality_{Accuracy} = w_1 \times M_1, \quad (5)$$

де w_1 – вага метрики «Кількість виявлених дефектів», M_1 – значення цієї метрики.

Аналогічно, для інших підатрибутів та їх метрик формули будуть відрізнятися відповідно до конкретних вимог та особливостей програмного продукту.

Отже, узагальнена оцінка якості програмного продукту буде враховувати ваги та значення всіх метрик для всіх підатрибутів та атрибутів якості, що дозволить об'єктивно оцінити його якість з урахуванням різноманітних аспектів.

Висновки

У контексті проведеного розгляду недоліків на рівні математичного представлення моделей оцінки якості ПС на прикладі моделей СОСОМО та ISO 9126/25010 можна зробити наступні висновки:

– Потреба у вдосконаленні об'єктивності і точності: Обидва підходи мають свої обмеження, зокрема, у визначенні об'єктивної якості ПС. Математичне вираження моделей дозволить зробити процес оцінки більш точним і об'єктивним;

– Необхідність у стандартизації: Розробка математичних виразів для моделей СОСОМО та ISO 9126/25010 сприятиме стандартизації процесу оцінки якості ПС і полегшить порівняння між різними проектами;

– Підвищення прогностичних можливостей: Математичне вираження моделей дозволить здійснювати прогнозування витрат, термінів та якості програмних проектів на основі конкретних вхідних параметрів, що полегшить планування і управління проектами;

– Необхідність гнучкості та адаптабельності: Підходи щодо математичного удосконалення моделей СОСОМО та ISO 9126/25010 повинні бути гнучкими та адаптованими до різноманітних умов та особливостей конкретних програмних проектів.

Отже, розробка математичних виразів для моделей оцінки якості ПС відображає потребу у подальшому вдосконаленні та розвитку методів оцінки з метою підвищення об'єктивності, точності та застосовності в практичних умовах.

Узагальнений висновок на рівні моделювання полягає в тому, що створення багатоцільових моделей якості ПС є ключовим етапом у забезпеченні їхньої успішної розробки, впровадження та експлуатації. Ці принципи не лише допомагають оцінити різні аспекти якості програмного забезпечення, такі як функціональність, продуктивність, надійність, ефективність, зручність використання та безпека, але й забезпечують системний підхід до розробки, що враховує різноманітні вимоги та очікування зацікавлених сторін. Створення таких моделей передбачає використання математичних формул, алгоритмів та методів, які дозволяють об'єктивно оцінити рівень якості системи, інтегруючи різноманітні аспекти та враховуючи специфічний контекст використання. Успішна реалізація цих принципів дозволяє підвищити ефективність розробки програмного забезпечення, зменшити ризики та витрати, покращити співпрацю між різними сторонами проекту та забезпечити високу якість та задоволення від продукту в кінцевих користувачів.

Список використаної літератури:

1. Yamming, C., & Shiyi, X. (2007). Exploration of complexity in software reliability. *Tsinghua Science & Technology*, 1(2), 266–269.
2. 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.
3. Aftab, S., Abbas, S., Ghazal, T. M., Ahmad, M., Hamadi, H. A., Yeun, C. Y., & Khan, M. A. (2023). A Cloud-Based Software Defect Prediction System Using Data and Decision-Level Machine Learning Fusion. *Mathematics*, 11(3), 632. <https://doi.org/10.3390/math11030632>
4. Foidl, H., & Felderer, M. (2018). Integrating software quality models into risk-based testing. *Software Quality Journal*, 26, 809–847.
5. 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.
6. Aggarwal, G., & Gupta, D. V. (2013). Neural network approach to measure reliability of software modules: A review. *International Journal of Advances in Engineering Sciences*, 3(2), 1–7.
7. Pietrantuono, R. (2020). On the testing resource allocation problem: Research trends and perspectives. *Journal of Systems and Software*, 161, 42.
8. Miguel, J. P., Mauricio, D., & Rodriguez, G. (2014). A Review of Software Quality Models for the Evaluation of Software Product. *International Journal of Software Engineering & Applications*, 5(6), 31–53.
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. 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.
11. 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.
12. 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. <https://doi.org/10.1016/j.infsof.2009.05.003>
13. Winther, R., Gran, B. A., & Dahll, G. (Eds.). (2005). Computer Safety, Reliability and Security: *Proceedings of the 24th International Conference SAFECOMP 2005, Friedrikstadt, Norway, September 28-30, 2005.* – 409 p.
14. 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).
15. Musa, J. D. (1993). Operational Profiles in Software Reliability Engineering. *IEEE Software*, 10(2), 14–32.
16. Lyu, M. R. (1997). Optimization Of Reliability Allocation And Testing Schedule For Software Systems. In *Proceedings Eighth International Symposium on Software Reliability Engineering (ISSRE '97)* (pp. 336–438).
17. Letichevsky, A., Kapitonova, J., Letichevsky Jr., A., Volkov, V., Baranov, S., & Kotlyarov, V. (2005). Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications. In *ISSRE 2004, WITUL, Rennes*, 4 (pp. 112–142).
18. Letichevsky A. (2005) Basic Protocols, Message Sequence Charts, and the Verification of Requerements Specifications / A. Letichevsky, J. Kapitonova, A.Letichevsky Jr., V. Volkov, S. Baranov, V. Kotlyarov // *ISSRE 2004, WITUL, Rennes*, 4 pp 112 – 142.

19. 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. doi:10.1109/TSE.2009.27

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

21. 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.

Автори статті

Шантир Антон - кандидат технічних наук, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

Зінченко Вячеслав - аспірант, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

Єльченко Сергій - кандидат технічних наук, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

Кравчук Петро - PhD, доцент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

Authors of the article

Shantyr Anton - Candidate of Science (technic), State University of Information and Communication Technologies, Kyiv, Ukraine.

Zinchenko Viacheslav - postgraduate, State University of Information and Communication Technologies, Kyiv, Ukraine.

Yelchenko Serhiy - Candidate of Science (technic), State University of Information and Communication Technologies, Kyiv, Ukraine.

Kravchuk Petro - PhD, Associate Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.