

## ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ МЕТРИК ЯКОСТІ НА ЕТАПІ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

*У статті виділено метрики якості ПЗ етапу проектування з точними та прогнозованими значеннями, досліджено діапазони значень метрик якості, а також визначено значущість, кількість інформації та ефективність кожної метрики якості ПЗ етапу проектування.*

*In article the design stage software quality metrics with the exact and predicted values were chosen, ranges of quality metrics have been researched, the significance, the quantity of information and the efficiency of each quality metric have been determined.*

Ключові слова: програмне забезпечення (ПЗ), етап проектування ПЗ, метрики якості ПЗ на етапі проектування, діапазони значень метрик якості ПЗ, ефективність метрик якості ПЗ.

### Вступ

Ключовим фактором забезпечення ефективного застосування програмних продуктів є ретельне оцінювання та досягнення високих значень показників якості. Постійне підвищення складності функцій, реалізованих програмами в інформаційних системах, безпосередньо призводить до збільшення їх обсягу та трудомісткості створення. Відповідно до змін складності програм зростає кількість виявлених та залишених в них дефектів та помилок, що відображається на їх якості. Програма обсягом в мільйони рядків тексту в принципі не може бути безпомилковою. Проблема виявлення та усунення помилок загострюється по мірі збільшення складності задач та програм, які їх вирішують, і загрожує катастрофами в галузях, що використовують програмне забезпечення (ПЗ) [1–3].

Якість програмного забезпечення – це характеристика ПЗ, яка відображає ступінь його відповідності вимогам. При цьому вимоги можуть трактуватись досить широко, що породжує цілий ряд незалежних визначень поняття якості. Якість можна трактувати також як ступінь відповідності присутніх характеристик вимогам. Також якість – це повнота властивостей і характеристик продукту, процесу або послуги, які забезпечують здатність задовольняти оголошеним або передбачуваним потребам.

Проблема полягає у забезпеченні потрібної якості функціонування ПЗ з врахуванням того, що деяка невідома кількість помилок та дефектів завжди залишається в складних комплексах програм, і повинна бути блокована або скорочена їх негативна дія до допустимого рівня. В зв'язку з цим стратегічна задача в життєвому циклі сучасного ПЗ – забезпечення якості програмних продуктів [4].

Забезпечення якості – це сукупність планованих та систематичних заходів, необхідних для впевненості в тому, що продукція або процеси задовольняють певним вимогам до якості. Система забезпечення якості застосовно до програмних засобів – це сукупність методів та засобів організації керуючих та виконавчих підрозділів підприємства, які беруть участь в проектуванні, розробці та супроводженні комплексів програм з метою надання їм властивостей, що забезпечують задоволення певних потреб замовників та споживачів при мінімальних або допустимих витратах ресурсів [4].

Якість ПЗ визначається якістю методів та інструментальних засобів, які застосовувались для забезпечення всього їх життєвого циклу. На практиці важливо оцінювати якість програм не лише в завершеному вигляді, але й в процесі їх проектування і розроблення.

На сьогодні, ПЗ є визначальною складовою багатьох систем, серед яких системи критичного застосування, вбудовані та спеціалізовані системи різноманітного призначення. Для зазначених систем наявність помилок та низька якість ПЗ загрожує катастрофами, які призводять до людських жертв, екологічних катаклізмів, економічних втрат. Проблема виявлення та усунення помилок загострюється по мірі збільшення складності ПЗ. Розвиток сучасних технологій розроблення ПЗ вимагає динамічного розвитку засобів оцінки якості ПЗ, причому вже на етапі проектування (з точки зору економічної та часової доцільності).

Для одержання оцінки значень показників якості за стандартом [5] використовуються такі методи:

1) вимірювальний – базується на використанні інструментальних, вимірювальних та спеціальних програмних засобів для одержання інформації про властивості та характеристики ПЗ (обсяг ПЗ, кількість рядків коду, кількість операторів, кількість гілок в програмі, кількість точок входу/виходу та ін);

2) реєстраційний – заснований на одержанні інформації під час випробувань або функціонування ПЗ, коли реєструються або підраховуються певні події (час і кількість збоїв та відмов, час передачі керування від одного модуля до іншого, час початку і завершення роботи ПЗ);

3) органолептичний – заснований на використанні інформації, одержаної в результаті аналізу сприйняття органів чуття, і застосовується для визначення таких показників як зручність застосування, ефективність;

4) розрахунковий – базується на використанні теоретичних та емпіричних залежностей (на ранніх етапах розробки), статистичних даних, зібраних при проведенні випробувань, експлуатації та супроводженні ПЗ. Розрахунковими методами оцінюються показники надійності, точності, стійкості, час реакції, необхідні

ресурси та ін.;

5) експертний – здійснюється групою експертів. Їх оцінка базується на досвіді та інтуїції, а не на безпосередніх результатах розрахунків або експериментів. Цей метод реалізується шляхом перегляду програм, кодів, супровідних документів, описів вимог до ПЗ групою експертів. Для цього встановлюються контрольовані ознаки, корельовані з одним або декількома показниками якості і включені в карти опитування експертів. Метод застосовується при оцінці таких показників, як наочність, повнота та доступність програмної документації, легкість засвоєння, аналізованість, документованість, структурованість ПЗ та ін.

З опису методів вимірювання показників (метрик) якості зрозуміло, що на етапі проектування ПЗ неможливо виміряти жодної характеристики ще не розробленого ПЗ, неможливо реєструвати моменти процесу виконання ще не існуючого ПЗ і неможливо сприйняти органами чуття інформацію щодо нерозробленого ПЗ. Отже, на етапі проектування є можливість визначити якість ПЗ лише із застосуванням розрахункових та експертних методів.

Згідно [6], метрика визначається як міра ступеня володіння властивістю, яка має числове значення. Взагалі, метрика ПЗ – це міра, яка дозволяє одержати числове значення деякої властивості ПЗ або його специфікації.

У галузі дослідження програмних метрик залишається *ряд невирішених питань*:

1) технологія вимірювання якості ще не досягла зрілості – лише 1.5% софтверних організацій намагаються оцінити якість процесів і готового продукту кількісно, за допомогою метрик, і лише 0.5% софтверних організацій намагаються покращити роботу, керуючись кількісними критеріями якості з метою випуску бездефектних продуктів;

2) відсутні єдині стандарти на метрики – створено більше тисячі метрик, кожен постачальник "вимірювальної" системи пропонує власні способи оцінки якості і відповідно метрики;

3) існує проблема складності інтерпретації величин метрик – значення метрик, одержані за допомогою "вимірювальних систем", неінформативні або малоінформативні для користувача, замовника, а часто і для програміста;

4) метрики розраховуються лише для готового ПЗ – всі "вимірювальні" системи орієнтовані на розрахунок метрик для програмного коду, але часто є необхідність у розрахунку метрик вже на етапі проектування – метрик якості з точними значеннями для проекту ПЗ і метрик якості з прогнозованими значеннями для розроблюваного за проектом програмного забезпечення;

5) низький рівень автоматизації аналізу та опрацювання метрик якості програмного забезпечення – на сьогодні автоматизовано лише процеси збирання, реєстрації та обчислення метричної інформації;

6) відсутня можливість порівняння принципово нового проекту з попередніми, що призводить до неможливості інтерпретації одержаних метрик для нового проекту;

7) основними параметрами при виборі варіанту реалізації ПЗ є вартість та тривалість розроблення і репутація фірми-проектувальника, але рішення, прийняті на основі цих параметрів, не завжди гарантують належну якість ПЗ.

Саме через невирішеність цих питань поки що неможливо створити бездефектне високоякісне ПЗ. Складність обґрунтування вибору та інтерпретації метрик в процедурах прийняття виробничих рішень та ігнорування етапів життєвого циклу ПЗ не дозволяють повноцінно використовувати метрики для підвищення якості ПЗ.

### 1. Метрики якості програмного забезпечення

У роботах [7–9] були визначені метрики якості етапу проектування ПЗ з точними значеннями:

1) *метрики зв'язності* (зв'язність (cohesion) – внутрішня характеристика програмного модуля) – залежать від типу модуля або проекту. Правила визначення рівня зв'язності модуля та сили зв'язності (СЗ):

- якщо модуль – одинична проблемно-орієнтована функція, то рівень зв'язності – функційний (частини модуля разом реалізують одну функцію) і СЗ=10;

- якщо дії всередині модуля зв'язані даними, і порядок дій всередині модуля важливий, то рівень зв'язності – інформаційний (вихідні дані однієї частини використовуються як вхідні дані в іншій частині модуля) і СЗ=9;

- якщо дії всередині модуля зв'язані даними, і порядок дій всередині модуля не має жодного значення, то рівень зв'язності – комунікативний (частини модуля зв'язані даними – працюють з однією структурою даних) і СЗ=7;

- якщо дії всередині модуля зв'язані потоком керування, і порядок дій всередині модуля важливий, то рівень зв'язності – процедурний (частини модуля зв'язані порядком виконуваних ним дій, які реалізують деякий сценарій поведінки) і СЗ=5;

- якщо дії всередині модуля зв'язані потоком керування, і порядок дій всередині модуля не має жодного значення, то рівень зв'язності – часовий (частини модуля не зв'язані, але необхідні в один і той же момент роботи системи) і СЗ=3; недоліком такого типу зв'язності є сильний взаємний зв'язок з іншими модулями, через що проект стає дуже чутливим до внесення змін;

- якщо дії всередині модуля жодним чином не зв'язані, але належать до однієї категорії, то рівень зв'язності – логічний (частини модуля об'єднані за принципом функційної подібності) і СЗ=1; недоліками логічної зв'язності є складне спряження та велика ймовірність внесення помилок при зміні спряження заради

однієї з функцій;

- якщо дії всередині модуля жодним чином не зв'язані, а також не належать до однієї категорії, то рівень зв'язності – за співпадінням (в модулі відсутні явно виражені внутрішні зв'язки) і  $C3=0$ .

Можливі більш складні випадки, коли з модулем асоціюються декілька рівнів зв'язності. В такому випадку, враховуючи статистичну та експертну інформацію про тип та область застосування розроблюваного ПЗ, модулю присвоюють найсильніший рівень зв'язності або найслабший рівень зв'язності.

Чим вище зв'язність модуля, тим кращий результат проектування, тобто чим "чорніша" скринька проекту (захисна оболонка модуля), тим менше "важелів керування" на ній знаходиться і тим вона простіша. Такі типи зв'язності, як зв'язність за співпадінням, логічна зв'язність та часова зв'язність – результат невірною планування архітектури, а процедурний тип зв'язності – результат недбалого планування архітектури додатку;

2) *метрики зчеплення* (зчеплення (coupling) – зовнішня характеристика модуля, яку бажано і слід зменшувати) – міра взаємозалежності модулів за даними. Кількісно зчеплення вимірюється ступенем зчеплення ( $C3ч$ ). Виділимо наступні типи зчеплення:

- зчеплення за даними ( $C3ч=1$ ) – один модуль викликає інший модуль, всі вхідні і вихідні параметри модуля, що викликається, – прості елементи даних;

- зчеплення за зразком ( $C3ч=3$ ) – в якості вхідних і вихідних параметрів використовуються структури даних;

- зчеплення за керуванням ( $C3ч=4$ ) – один модуль явно керує функціонуванням іншого модуля за допомогою прапорців або перемикачів і надсилає йому керуючі дані;

- зчеплення за зовнішніми посиланнями ( $C3ч=5$ ) – модулі посилюються на один і той же глобальний елемент даних;

- зчеплення за спільною областю ( $C3ч=7$ ) – модулі поділяють одну й ту ж глобальну структуру даних;

- зчеплення за вмістом ( $C3ч=9$ ) – один модуль прямо посилюється на вміст іншого модуля не через його точку входу.

3) *метрика звертання до глобальних змінних* – пара (модуль, глобальна змінна) – пара ( $p, r$ ), де  $p$  – модуль, який має доступ до глобальної змінної  $r$ . В залежності від наявності реального звертання до змінної  $r$  формуються 2 типи пар ( $p, r$ ) – фактичні і можливі. Характеристика  $A_{up}$  показує, скільки разів модуль дійсно одержить доступ до глобальної змінної, а характеристика  $R_{up}$  – скільки разів він міг би одержати такий доступ. Тоді наближену ймовірність посилання довільного модуля на довільну глобальну змінну можна обчислити як відношення:

$$R_{up} = \frac{A_{up}}{P_{up}}. \quad (1)$$

Чим вище така ймовірність, тим вище ймовірність "несанкціонованої" зміни певної глобальної змінної, що призводить до ускладнення модифікації програми;

4) *час модифікації моделей* – метрика процесу розробки ПЗ;

5) *кількість знайдених помилок при інспектуванні моделей та прототипів підсистем, модулів, функцій, вимог та густота помилок* (кількість помилок на одну підсистему, модуль, функцію, вимогу) – вказує на проблемну підсистему, модуль, функцію, вимогу. Це метрика процесу розробки ПЗ.

У роботах [7–9] також були визначені метрики якості етапу проектування ПЗ з прогнозованими значеннями:

1) *загальний час розроблення ПЗ* – метрика процесу розробки ПЗ, вимірюється у робочих днях;

2) *час виконання робіт процесу проектування* – метрика процесу розробки ПЗ, вимірюється у робочих днях;

3) *очікувана вартість розроблення ПЗ:*

$$ВАРТІСТЬ_i = LOC_{оч_i} \times ВАРТІСТЬ_{рядка_i}; \quad (2)$$

вартість рядка є константою і не змінюється від реалізації до реалізації; вимірюється у гривнях або доларах США;

4) *прогнозована вартість перевірки якості* – метрика процесу розробки ПЗ; вимірюється у гривнях або доларах США;

5) *прогнозована продуктивність розроблення ПЗ* – вимірюється у хвилинах на один рядок коду;

6) *прогнозовані витрати на реалізацію програмного коду* – вимірюється у гривнях або доларах США;

7) *прогнозований функційний розмір FP* – вимірює суть можливостей майбутньої програми. Для обчислення функційного розміру: ідентифікуються очікувані від програмного додатку функції за критеріями International Function Point Users Group (IFPUG). Метод визначення функційного розміру опишемо покроково наступним чином:

- виділити функції додатку;

- для кожної потенційної виділеної функції слід поррахувати кількість зовнішніх входів  $EI$ , які по-різному впливають на виконувану функцію; кількість зовнішніх виходів  $EO$ , для істотно різних

алгоритмів і нетривіальної функційності; кількість зовнішніх запитів  $EIN$ ; кількість внутрішніх логічних файлів або унікальних логічних груп користувачьких даних  $ILF$ ; кількість зовнішніх логічних файлів або унікальних логічних груп користувачьких даних  $ELF$ ;

- кожен з визначених на попередньому кроці факторів множиться на коефіцієнт, який визначається складністю даного фактору в програмному проекті ( $EI - 3$  або  $4$  або  $6$  (від простого до складного),  $EO - 4$  або  $5$  або  $7$ ,  $EIN - 3$  або  $4$  або  $6$ ,  $ILF - 7$  або  $10$  або  $15$ ,  $ELF - 5$  або  $7$  або  $10$ ). Ці добутки додаються за кожною функцією;

- визначити ваги для 14 загальних характеристик проекту (від 0 до 5); ваги вказуються в формі діапазонів, що відображає невпевненість відносно функцій; призначення ваг вимагає певного досвіду у використанні методу функційного розміру;

- обчислити уточнений функційний розмір за формулою:

$$\text{Уточн.функц.розмір} = \text{Наближений\_функц\_розмір} \times [0.65 + 0.01 \times (\text{Сума\_загальних\_характеристик})] \quad (3)$$

Якщо до проекту не висувається жодних спеціальних вимог (всі загальні характеристики дорівнюють 0), то неуточнений функційний розмір слід зменшити на 35%, інакше слід збільшити на 1% на кожну одиницю значень загальних характеристик.

Функційний розмір використовується як відносна метрика для порівняння з попередніми проектами, за його допомогою можна обчислити кількість рядків коду, що дозволяє визначити загальну трудомісткість та терміни проекту. Є вільно поширювані інструменти для обчислення функційного розміру;

8) прогнозована оцінка трудовитрат за моделлю Боєма трудовитрати на розробку програмних додатків зростають швидше, ніж розмір додатків. Для представлення даного співвідношення використовується експоненційна функція зі значенням показника, близьким до 1.12:

$$\text{Трудовитрати} = a \times KLOC^b, \quad (4)$$

де  $KLOC$  – кількість тисяч рядків коду,  $a, b$  – коефіцієнти СОСОМО. Для органічного (самостійного) програмного проекту:  $a = 2,4; b = 1,05$ . Для вбудованих програмних проектів (інтеграція апаратного та програмного забезпечення):  $a = 3,6; b = 1,20$ . Для проміжних програмних проектів (не органічні, але й не жорстко вбудовані):  $a = 3,0; b = 1,12$ .

9) прогнозована оцінка тривалості проекту за моделлю Боєма – тривалість проекту за моделлю Боєма зростає експоненційно разом з докладеними до проекту зусиллями. Однак в даному випадку значення експоненти менше 1 і складає близько 0.35:

$$\text{Тривалість} = c \times \text{Трудовитрати}^d = c \times (a \times KLOC^b)^d = c \times a^d \times KLOC^{b \times d}, \quad (5)$$

де  $c, d$  – коефіцієнти СОСОМО. Для органічного (самостійного) програмного проекту:  $c = 2,5; d = 0,38$ . Для вбудованих програмних проектів (інтеграція апаратного та програмного забезпечення):  $c = 2,5; d = 0,32$ . Для проміжних програмних проектів (не органічні, але й не жорстко вбудовані):  $c = 2,5; d = 0,35$ .

## 2. Діапазони значень метрик якості програмного забезпечення

Для розрахунку максимальних значень метрик якості ПЗ слід ввести певні обмеження на проекти та ПЗ, метрики якості яких аналізуватимуться.

Розрахуємо максимальні значення метрик якості ПЗ при використанні наступних обмежень:

1) кожен модуль реально одержує доступ до глобальної змінної стільки разів, скільки може одержати такий доступ;

2) ПЗ містить не більше 50000 рядків вихідного коду;

3) тип проекту не задано;

4) постановка задачі займає 10% загального часу розроблення ПЗ і проектної вартості; проектування – 35% загального часу і проектної вартості; програмування – 35% загального часу і проектної вартості; тестування, налагодження і перевірка якості – 20% загального часу і проектної вартості;

5) 25% часу проектування займає побудова та модифікація моделей;

6) максимальна кількість помилок моделей та прототипів одного модуля не повинна перевищувати 100;

7) ПЗ має не більше 50 модулів;

8) 50% часу та вартості, відведених на тестування, налагодження і перевірку якості, займає перевірка якості ПЗ;

9) кількість зовнішніх входів кожного модуля – 49, кількість зовнішніх виходів кожного модуля – 49, кількість зовнішніх запитів до кожного модуля – 49;

10) кожен модуль має максимум 50 внутрішніх логічних файлів та використовує 50 зовнішніх логічних файлів.

Використовуючи формули (1)–(5) та введені обмеження для розрахунку метрик, отримаємо наступні діапазони значень метрик якості етапу проектування програмного забезпечення (таблиця 1).

## Інтервали значень метрик якості ПЗ етапу проектування

№	Метрики з точними значеннями	Метрики з прогнозованими значеннями
1	Метрика зв'язності: 0..10	Загальний час розроблення ПЗ: 0..520 (робочих днів)
2	Метрика зчеплення: 0..9	Час етапу проектування: 0..182 (робочих днів)
3	Метрика звертання до глобальних змінних: 0..1	Продуктивність розроблення ПЗ: 0..5 (хвилин на один рядок коду)
4	Час модифікації моделей: 0..46	Вартість перевірки якості ПЗ: 0..20000 (грн)
5	Кількість виявлених помилок при інспектуванні моделей та прототипів: 0..5000	Вартість розроблення ПЗ: 0..200000 (грн)
6		Вартість реалізації коду програми: 0..70000 (грн)
7		Функційний розмір (FP): 0..2945
8		Оцінка трудовитрат за моделлю Боема: 0..394 (людиномісяців)
9		Оцінка тривалості проекту за моделлю Боема: 0..520 (робочих днів)

## 3. Визначення ефективності метрик якості програмного забезпечення

Аналіз діапазонів значень метрик якості ПЗ показує, що існує великий розкид діапазонів значень вхідних даних – значення вхідних векторів розрізняються в десятки, сотні, тисячі, десятки та сотні тисяч разів. Опрацювання метрик якості з такими діапазонами значень призведе до того, що впливовими при визначенні якості проектів та ПЗ є метрики з великими діапазонами значень (метрики очікуваної вартості розроблення ПЗ, прогнозованих витрат на реалізацію коду, прогнозованої вартості перевірки якості ПЗ, загальної кількості знайдених помилок при інспектуванні моделей та прототипів), а метрики з малими діапазонами значень (метрики зв'язності, зчеплення, звертання до глобальних змінних, часу модифікації моделей, продуктивності розроблення ПЗ) взагалі не впливають на якість ПЗ, тобто частина метричної інформації не буде враховуватись при визначенні якості проекту та ПЗ.

Оцінимо інформацію, яку містять метрики якості ПЗ на етапі проектування. Критерії оцінювання інформації [10]: 1) релевантність – наявність зв'язку з проблемою (відповідність інтересам та проблемі) та здатність інформації внести ясність у процес розуміння проблеми; 2) достовірність – наскільки представлений опис відповідає дійсності; 3) значущість – розуміння інформації, повнота висвітлення предмету інтересу, своєчасність інформації та її достатність для прийняття рішень, а також кількість інформації [11]:

$$I = H(\text{апріорна}) - H(\text{апостеріорна}), \quad (6)$$

де  $H(\text{апріорна})$  – ентропія системи (показник незнання системи) до експерименту,  $H(\text{апостеріорна})$  – ентропія системи після експерименту, тоді, якщо  $I > 0$ , то в результаті експерименту одержано нову інформацію, показник незнання системи зменшився.

Щодо релевантності метрик якості ПЗ, то рівень релевантності дорівнює 1, оскільки саме метрики якості дають уявлення про якість проекту і ПЗ. Достовірність метрик якості ПЗ дорівнює 1, оскільки використовуються лише класичні метрики якості, описані в галузевій літературі.

Оскільки релевантність і достовірність метрик якості ПЗ є сталими величинами, то основним критерієм оцінювання інформації, який впливатиме на оцінку ефективності метрик якості ПЗ, є саме значущість інформації.

Сформуємо підсумковий критеріальний показник ефективності для кожної метрики якості за адитивним критерієм.

Адитивний критерій  $A$  формується шляхом ділення на число показників ефекту  $n$  суми добутоків часткових показників ефекту  $l_i$  на коефіцієнти значущості  $i$ -го показника  $g_i$  [12]:

$$A = (1/n) \cdot \sum_{i=1}^n (l_i \cdot g_i). \quad (7)$$

Значущість, а відтак і ефективність, кожної метрики якості ПЗ етапу проектування залежить від 4-х показників ефекту ( $n = 4$ ): розуміння інформації, повнота висвітлення предмету інтересу, своєчасність інформації та її достатність для прийняття рішень, кількість інформації. Кожен показник ефекту розглядатимемо для спрощення розрахунків як значення з інтервалу  $[0, 1]$ , де 0 – мінімальне значення показника, 1 – максимальне значення. Значущості показників ефекту для кожної метрики рівні, причому

$\sum_{i=1}^n g_i = 1$ , отже,  $g_i = \frac{1}{4} = 0,25$ . Потрібно тепер розрахувати 4 часткових показники ефекту  $l_i$  для кожної метрики якості ПЗ етапу проектування і відповідно ефективність кожної метрики за адитивним критерієм.

Показник "розуміння інформації"  $l_1$  дорівнює 1 для кожної з метрик, оскільки метрики якості

розраховані вірно, з повним розумінням. Показник "повнота висвітлення предмету інтересу"  $l_2$  залежить від повноти висвітлення метрикою оброблюваної інформації, функцій обробки інформації та переліку модулів, режимів роботи ПЗ. Показник "своєчасність інформації та її достатність для прийняття рішень"  $l_3$  залежить від достатності інформації про оброблювану інформацію, функцій обробки інформації та перелік модулів, а також про режими роботи ПЗ. Для розрахунку показника "кількість інформації"  $l_4$  вважатимемо, що після опрацювання всіх метрик якості є повна інформація про якість проекту та ПЗ, тому кількість інформації  $I$ , яку містять всі 14 метрик якості разом, згідно формули (6) дорівнює 1. Вважатимемо також, що показник незнання якості ПЗ зменшується в рівному ступені після опрацювання кожної метрики якості, тому кількість інформації кожної метрики якості  $I_m = \frac{1}{14} = 0,072$ .

Для визначення показників "повнота висвітлення предмету інтересу" та "своєчасність інформації та її достатність для прийняття рішень" виділимо інформаційні потоки метрик якості: *метрика зв'язності* враховує оброблювану інформацію та режими роботи проекту або ПЗ; *метрика зчеплення* приймає до уваги оброблювану інформацію, функції оброблення інформації і кількість модулів та режими роботи проекту або ПЗ; *метрика звертання до глобальних змінних* враховує оброблювану інформацію, а також функції оброблення інформації; *метрики часу модифікації моделей, кількості виявлених помилок при інспектуванні моделей та прототипів, загального часу розроблення ПЗ, часу етапу проектування, продуктивності розроблення ПЗ, вартості розроблення ПЗ, вартості перевірки якості ПЗ, вартості реалізації коду програми* приймають до уваги функції оброблення інформації та перелік модулів; *метрика функційного розміру* враховує оброблювану інформацію, функції оброблення інформації та перелік модулів, а також режими роботи ПЗ; *метрики оцінки трудовитрат та тривалості проекту за моделлю Боєма* приймають до уваги функції оброблення інформації та перелік модулів, а також режими роботи ПЗ.

Представимо перелік метрик із значеннями 4-х показників ефекту та визначеною за формулою (7) ефективністю у таблиці 2.

Таблиця 2

**Ефективність метрик якості програмного забезпечення на етапі проектування**

№	Метрика якості ПЗ	Показник $l_1$	Показник $l_2$	Показник $l_3$	Показник $l_4$	Ефективність $A$
1	Метрика зв'язності	1	0,666	0,666	0,072	0,150
2	Метрика зчеплення	1	1	1	0,072	0,192
3	Метрика звертання до глобальних змінних	1	0,4995	0,4995	0,072	0,129
4	Час модифікації моделей	1	0,333	0,333	0,072	0,109
5	Загальний час розроблення ПЗ	1	0,333	0,333	0,072	0,109
6	Кількість виявлених помилок при інспектуванні моделей та прототипів	1	0,333	0,333	0,072	0,109
7	Час етапу проектування	1	0,333	0,333	0,072	0,109
8	Продуктивність розроблення ПЗ	1	0,333	0,333	0,072	0,109
9	Вартість перевірки якості ПЗ	1	0,333	0,333	0,072	0,109
10	Вартість розроблення ПЗ	1	0,333	0,333	0,072	0,109
11	Вартість реалізації коду програми	1	0,333	0,333	0,072	0,109
12	Функційний розмір	1	1	1	0,072	0,192
13	Оцінка трудовитрат за моделлю Боєма	1	0,666	0,666	0,072	0,150
14	Оцінка тривалості проекту за моделлю Боєма	1	0,666	0,666	0,072	0,150

Отже, порівняно високу ефективність (значущість) мають метрики: зчеплення, функційного

розміру, зв'язності, оцінки трудовитрат та тривалості проекту за моделлю Боема, звертання до глобальних змінних; порівняно низьку ефективність мають метрики: часу модифікації моделей, кількості виявлених помилок при інспектуванні моделей та прототипів, загального часу розроблення ПЗ, часу етапу проектування, продуктивності розроблення ПЗ, вартості розроблення ПЗ, вартості перевірки якості ПЗ, вартості реалізації коду програми. Отже, очевидно, що неможливо не враховувати метрики з малими діапазонами значень тому, що ефективність багатьох з них достатньо висока, тобто їх неврахування призведе до істотного зниження ефективності оцінювання якості проекту та ПЗ.

Оцінимо кількість інформації, яка втрачається при неврахуванні 5-и метрик з малими діапазонами значень (метрики зв'язності, зчеплення, звертання до глобальних змінних, часу модифікації моделей, продуктивності розроблення ПЗ). Якщо кількість інформації кожної метрики якості  $I_M = 0,072$ , то кількість інформації п'яти метрик:  $I_{grp} = 5 \cdot 0,072 = 0,36$ . Отже, при неврахуванні впливу на якість проекту та ПЗ метрик якості ПЗ з малими діапазонами значень втрачається 36% значущої інформації.

#### Висновки

У статті визначено метрики якості ПЗ етапу проектування, досліджено діапазони значень метрик якості, проведено оцінювання показників ефекту та визначення ефективності кожної метрики якості ПЗ етапу проектування з точки зору оцінювання значущості її інформації за адитивним критерієм ефективності.

Виявлено, що через великий розкид діапазонів значень метрик якості відбувається втрата інформації, яку надають метрики якості ПЗ етапу проектування з малими діапазонами значень (метрики зв'язності, зчеплення, звертання до глобальних змінних, часу модифікації моделей, продуктивності розроблення ПЗ).

Оцінка кількості значущої інформації, яка втрачається при неврахуванні метрик з малими діапазонами значень, та оцінки ефективності метрик якості ПЗ етапу проектування показали, що неможливо не враховувати метрики з малими діапазонами значень через їх високу ефективність та через втрату значної кількості (36%) значущої інформації.

Отже, на основі проведених досліджень зроблено висновок, що необхідне попереднє опрацювання метричної інформації з метою рівнозначного врахування метрик з великими і з малими діапазонами значень для запобігання втрати значущої інформації та ефективності оцінювання якості проекту та ПЗ.

#### Література

1. Скляр В.В. Оценка качества и экспертиза программного обеспечения / Скляр В.В. – Харьков : НАУ "ХАИ", 2008. – 204 с.
2. Майерс Г. Надежность программного обеспечения / Майерс Г. ; пер. с англ. – М. : Мир, 1980. – 360 с.
3. Myers G.J. The Art of Software Testing. – New York: John Wiley and Sons, 1979. – 312 pp.
4. Липаев В.В. Программная инженерия. Методологические основы / Липаев В.В. – М. : ТЕИС, 2006. – 608 с
5. ГОСТ 28195-89. Оценка качества программных средств. Общие положения
6. IEEE Standard Glossary of Software Engineering Terminology / IEEE Std 610.12-1990
7. Поморова О.В. Интеллектуальный метод оцінювання результатів проектування та прогнозування характеристик якості програмного забезпечення / О.В. Поморова, Т.О. Говорущенко // Радіоелектронні і комп'ютерні системи – Харків : НАУ "ХАИ", 2010 – № 6. – С. 211–218.
8. Поморова О.В. Оцінювання результатів проектування та прогнозування характеристик якості програмного забезпечення / О.В. Поморова, Т.О. Говорущенко, О.С. Онишук // Вісник Хмельницького національного університету. – Хмельницький : ХНУ, 2011. – № 2. – С. 168–178.
9. Поморова О.В. Аналіз методів та засобів оцінки якості програмних систем / О.В. Поморова, Т.О. Говорущенко // Радіоелектронні і комп'ютерні системи. – Харків : НАУ "ХАИ", 2009 – № 6. – С. 148–158.
10. Нежданов И. Критерии оценки информации (важность, точность, значимость) [Електронний ресурс]. – Режим доступу : <http://www.police-russia.ru/showthread.php?t=44683>
11. Значимость информации: определение [Електронний ресурс]. – Режим доступу : <http://otvet.mail.ru/question/2703994>
12. Белик Т.В. Принципы формирования критериев эффективности [Електронний ресурс]. – Режим доступу : <http://lego.biuss.ru/paragraf/742>

Рецензент: д.т.н. Поморова О.В.  
Надійшла 20.2.2012 р.