

ПІДХІД ДО ОЦІНЮВАННЯ ЕКОНОМІЧНИХ ХАРАКТЕРИСТИК ПРОЕКТНИХ РІШЕНЬ ПРИ РОЗРОБЦІ, МОДИФІКАЦІЇ ТА РЕІНЖИНІРИНГУ ПРОГРАМНИХ СИСТЕМ

Досліджені існуючі методи оцінювання характеристик проектів, на основі яких з урахуванням досвіду виконання проектів запропоновано методику оцінювання економічних характеристик проектів, що поєднує методи експертних оцінок, FPA та СОСОМО. Експериментально перевірено можливість використання оцінок за методом FPA як вхідних даних для моделі СОСОМО. За результатами експериментальних випробувань оцінено ефективність застосування методики на даних по виконаних проектах і показано, що розбиття даних на класи для калібрування моделі СОСОМО підвищує точність оцінок.

Вступ

Оцінювання характеристик, важливих в економічному аспекті, при виконанні проектів з розробки, модифікації або реінжинірингу ПС є необхідною складовою **процесу керування** розробкою та супроводженням життєздатних ПС [1]. Застосування методів оцінювання економічних характеристик дозволяє більш точно прогнозувати як матеріальні ресурси, необхідні для реалізації проекту, так і людські та часові ресурси. Це дає можливість приймати обґрунтовані рішення щодо формування пакету нових розробок ПС в організації, а також доцільності проектів модифікацій чи реінжинірингу супроводжуваних ПС. Внаслідок цього можна точніше планувати процеси виконання як окремого проекту, так і паралельних.

З іншого боку, існуюча система сертифікації рівня технологічної зрілості SEI CMM ставить впровадження в організації методики оцінки розміру проектів (одна з характеристик, розглядуваних у статті) як основну вимогу для отримання сертифікату рівня CMM Level 3.

Отже, використання методів оцінки економічних характеристик проектів значно підвищує рівень технологічної зрілості організації — розробника ПС, яка їх застосовує [2].

У даній статті висвітлено наступні задачі розглядуваної проблеми:

- здійснено огляд існуючих методів оцінки економічних характеристик проектів з розробки, модифікації та реінжинірингу ПС;
- розроблено підхід та запропонована методика оцінювання ефективності проектних рішень при розробці, модифікаціях чи реінжинірингу ПС з урахуванням досліджених методів;
- розглянуто питання ефективності застосування запропонованої методики.

1. Існуючі методи оцінювання проектів

В галузі управління проектами з розробки ПС гостро стоїть проблема оцінювання проектів з розробки, модифікації чи реінжинірингу ПС в аспекті необхідних для їх виконання ресурсів, бюджету та строків. Існування цієї проблеми пов'язане з тим, що процес створення ПС є процесом перетворення специфікацій та вимог до системи у програмні коди готової ПС. Складність управління такими процесами викликана складністю вимірювання характеристик цих процесів для отримання кількісних характеристик вхідної та результуючої інформації, складністю моделювання цих процесів. За відсутності засобів підтримки прийняття цих рішень управління проектами часто ведеться методом проб та помилок, а успіх залежить від досвіду та інтуїції керівника.

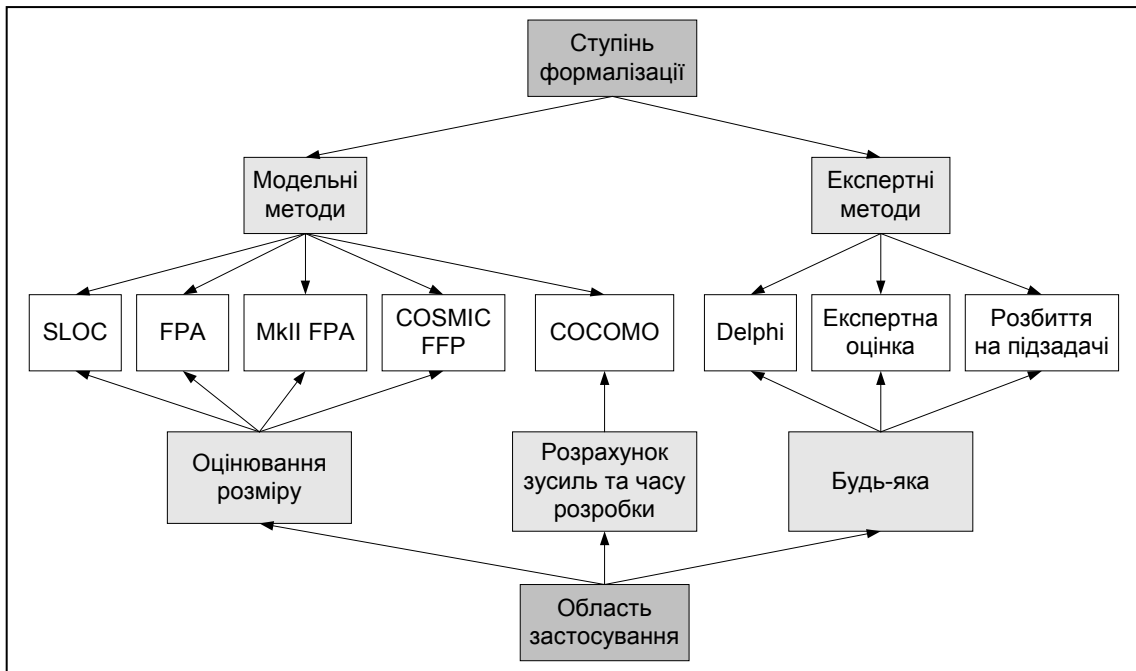


Рис. 1. Класифікація методів оцінювання проектів

Найголовнішими економічними характеристиками, що впливають на ефективність управління проектами, є такі:

- розмір проекту;
- трудомісткість;
- час, необхідний для виконання;
- вартість.

Існуючі підходи до оцінювання цих характеристик можна класифікувати за їх застосуванням та ступенем формалізації, як показано на рис. 1.

Методи, що базуються на експертних оцінках, більш універсальні і можуть застосовуватися у різних цілях, тоді як модельні прив'язані до конкретної цілі. Внаслідок цього експертні методи є найбільш широко застосовуваними. Модельні ж методи, як для вимірювання розмірів розроблених систем, так і моделювання процесів розробки, потребують менших витрат при їх застосуванні та дають результати, які є об'єктивними та легко перевіряються [2–4].

Коротко розглянемо методи, приведені на рис. 1. Детально вони описані в існуючій літературі [2–3].

1.1. Експертні методи. Звичайний метод експертних оцінок. Цей метод [5] базується на використанні неформалізованих знань експертів у

певній проблемній області. При цьому результати залежать від підготовки, досвіду або навіть настрою чи стану здоров'я експерта. Також варіюються потреби експертів у документації по оцінюваному проекту, хоча зазвичай чим більше даних про проект може обробити експерт, тим точнішою буде оцінка. Важливою є наявність документації по раніше виконаних проектах з тієї ж прикладної області зі схожими вимогами.

Метод Delphi. Метод Delphi [6] є розвитком методу експертних оцінок у напрямку підвищення об'єктивності та зниження похибки оцінки за рахунок компенсації розбіжностей в оцінках різних експертів.

Для зниження похибки використовуються результати оцінки декількох експертів, а для підвищення об'єктивності обмін результатами між ними здійснюється анонімно. Практичне застосування цього методу виглядає так:

- Обираються від 2 до 5 експертів для участі у робочій групі.
- Кожен експерт отримує пакет проектної документації, необхідні довідкові та історичні матеріали і форму для внесення до неї результатів.
- Експерти збираються на зустріч, на якій обговорюються всі необ-

хідні організаційні та технічні моменти щодо визначення розміру проекту.

- Кожен експерт робить власну оцінку, використовуючи будь-який узгоджений метод, та анонімно заносить результати до форми.

- Експерти збираються на зустріч, де обговорюються найнижча та найвища оцінки, при цьому авторство оцінок не розкривається.

- Якщо експерти не можуть дійти згоди щодо найнижчої та найвищої оцінок, процес продовжується з кроку 3.

Перевагою такого методу є більша точність оцінки, недоліком — збільшення потреби у людських та часових ресурсах у порівнянні зі звичайним методом експертних оцінок пропорційно кількості задіяних експертів.

Експертна оцінка з розбиттям проекту на підзадачі. Цей метод [5] є розвитком методу експертних оцінок у напрямку підвищення об'єктивності та зниження похибки оцінки за рахунок розбиття проекту на менші задачі, які або легше піддаються оцінюванню, або вже мають добре відому та чітко визначену оцінку. Звичайно вимагається, щоб ієрархія задач мала не менше двох та не більше трьох рівнів у глибину. Кількість задач на кожному рівні не обмежується. Розбиття проекту та оцінювання окремих задач можуть виконуватися різними експертами.

Перевагою такого методу є більша точність та можливість перевірити результат, не повторюючи процедуру оцінювання, недоліком — потреба у більших часових ресурсах у порівнянні зі звичайним методом експертних оцінок.

1.2. Методи оцінки розміру. SLOC (Source Lines Of Code). Цей метод [7] базується на вимірюванні найбільш очевидної характеристики програмного забезпечення — кількості рядків коду у вихідних текстах.

Головними недоліками методу є різне трактування поняття "рядок коду" для різних мов програмування, різний розмір однієї і тієї ж системи залежно від мов та технологій, використаних для її створення, залежність кі-

лькості рядків коду від стилю його написання.

Позитивними рисами цього методу є його наочність, легкість автоматизації, відсутність необхідності у навчанні спеціалістів, які будуть використовувати цей метод, легкість перевірки результатів. Завдяки цим якостям метод SLOC набув широкого застосування, оскільки головні його недоліки можна компенсувати введенням стандартів на стиль написання коду та чітким визначенням поняття рядка коду.

FPA (Function Point Analysis). Головною метою розробників методу FPA [8] було встановлення загальної одиниці виміру функціонального розміру програмного забезпечення (ПЗ), шкали виміру та методу вимірювання функціонального розміру. На відміну від методу SLOC функціональний розмір (ФР) не залежить від технологій, використаних для створення системи, стилю написання коду або мови програмування. За стандартом ISO/IEC, функціональний розмір програмного забезпечення є кількісною мірою його функціональності та визначається як кількісна оцінка функціональних вимог користувача до ПЗ. Функціональні вимоги визначають процеси та процедури, які будуть виконуватися ПЗ, або, іншими словами, що буде програма робити. ФР не залежить від того, як буде ПЗ працювати (наприклад, від вимог до якості або продуктивності), або від особливостей реалізації ПЗ. Метод FPA придатний до вимірювання ПЗ у всіх сферах застосувань і був успішно застосований у широкому колі проектів, від систем керування ракетною зброєю до систем фінансової звітності. FPA базується на таких поняттях: **користувач** (людина або ПЗ, які обмінюються інформацією з вимірюваним ПЗ), **межа ПЗ** (відокремлює ПЗ, що вимірюється, від користувачів), **елементарний процес** (будь-який обмін інформацією між ПЗ та користувачем, тут розрізняються на **зовнішній ввід** — **ЗВВ**, **зовнішній вивід** — **ЗВ**, **зовнішній запит** — **ЗЗ**), **файл** (розрізняються **внутрішні логічні файли** — **ВЛФ** та **зовнішні інтерфейс-**

ні файли — ЗІФ), елемент (елементарний процес або файл).

Після ідентифікації елементів визначається їх рівень складності та відповідна кількісна оцінка в одиницях функціональності. Функціональний розмір всього ПЗ отримується як сума оцінок його елементів.

Для оцінки складності елементів розглядаються такі їх характеристики, як кількість задіяних елементарних даних (ЕД), типів записів (ТЗ) та типів файлів (ТФ).

Функціональний розмір елементів системи залежно від кількості задіяних ТФ, ТЗ та ЕД визначається спеціальними таблицями.

Нарешті, розмір всієї системи в умовних одиницях функціональності отримується як

$$Size = \sum Size_{в\lambda\phi} + \sum Size_{з\lambda\phi} + \sum Size_{з\beta\beta} + \sum Size_{з\beta} + \sum Size_{зз}.$$

МкII FPA (Mark II Function Point Analysis). Метод МкII [9] є клоном оригінального методу FPA, орієнтованим на оцінювання систем, що працюють з базами даних. За МкII система складається з трьох типів елементів:

Вхідні дані, що надходять у систему, перетинаючи її межу.

Результуючі дані, що виходять із системи, перетинаючи її межу.

Обробка даних — процес, який використовує вхідні дані та породжує результуючі.

На відміну від FPA метод МкII включає всі випадки повторного використання одних і тих же типів даних у загальну оцінку.

Показник функціонального розміру системи розраховується за формулою

$$Size = N_{ex} \times W_{ex} + N_{рез} \times W_{рез} + N_{обр} \times W_{обр},$$

де N — відповідно кількість вхідних або результуючих даних або процесів обробки даних, а константи W мають значення $W_{ex} = 0.58$, $W_{рез} = 0.26$, $W_{обр} = 1.66$ [5].

COSMIC FFP (Common Software Measurement International Consortium

Full Function Points). COSMIC FFP [10] базується на тих самих визначеннях межі ПЗ та елементарного процесу, що і FPA. Але, на відміну від нього, відповідно до [10], система складається з процесів переміщення та обробки даних.

Для оцінювання розміру у системі виділяються чотири типи базових процесів: *зовнішній ввіг*, *зовнішній вивіг*, *внутрішнє зчитування*, *внутрішній запис*.

Зовнішні операції відбуваються між виділеним шаром системи та користувачем, при цьому перетинаючи межу системи, тоді як внутрішні відбуваються між шарами системи всередині межі.

Розмір i -го шару системи у одиницях Cfsu (Cosmic Functional Size Unit) отримується як

$$Size_i = Card(вв\text{в}\text{д}) + Card(в\text{в}\text{в}\text{д}) + Card(\text{зчитування}) + Card(\text{запис}),$$

де $Card$ (множина) — це потужність множини підпроцесів певного типу.

Оцінювання процесів маніпулювання даними у методі версії v.2.2 не було визначено.

1.3. Методи оцінювання трудомісткості, тривалості та вартості проєктів. **COCOMO** (Constructive Cost Model). В основі моделі COCOMO лежить припущення [11], що розмір системи ($Size$), виражений у одиницях SLOC, та величина зусиль, необхідних для розробки проєкту, виражена у людино-місяцях (person-month — PM), пов'язані співвідношенням вигляду

$$\log PM = A \times \log Size + B.$$

Це припущення було перевірено на базі, яка включала 161 проєкт з розробки ПЗ [12]. На цій же базі було визначено фактори, які мають значний вплив на хід виконання проєкту. Відповідні кількісні їх значення входять у змінні A та B .

Розрахунок зусиль на розробку. Оцінка зусиль на виконання проєкту (у людино-місяцях) за COCOMO проводиться за наведеними формулами. Індекс NS у значень PM та $TDEV$ позначає те, що ці оцінки зусиль та тривалості

відносяться до номінального розкладу, тобто обрхованого за моделлю COSOMO. Бюджет, виражений у людино-місяцях PM_{NS} оцінюється за формулою

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^{17} EM_i, \text{ де}$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j, \quad (1)$$

де константи A та B дорівнюють відповідно 2.94 та 0.91 [13]; EM_i — мультиплікативні фактори; SF_j — експоненційні фактори COSOMO. Розмір має бути вираженим в одиницях KSLOC (1 KSLOC = 1000 SLOC).

Розрахунок тривалості та ціни. Тривалість розробки (time to develop — TDEV) $TDEV_{NS}$ оцінюється за формулою

$$TDEV_{NS} = C \times (PM_{NS})^F,$$

де

$$F = D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j = D + 0.2 \times (E - B),$$

де константи C та D дорівнюють відповідно 3.67 та 0.28 [13].

Вартість проекту отримується множенням кількісної величини зусиль, виражених у людино-місяцях, на вартість людино-місяця.

2. Підхід до створення методики оцінювання економічних характеристик проектів з розробки, модифікацій та реінжинірингу ПС

Розглянемо коротко рішення, які приймаються у цьому аспекті в більшості організацій, ролі, задіяні у процесі оцінювання, та методи прийняття рішень, що при цьому використовуються.

Рішення — про відповідність проекту профілю організації-розробника (наявність обладнання та програмного забезпечення, досвіду розробки, можливість забезпечення потрібного рівня якості), про визначення розміру проекту, необхідних для його виконання зусиль, графіка та терміну виконання, складу проектної команди, вар-

тості проекту, економічну доцільність, прийняття проекту у розробку.

Ролі — вище керівництво, представник замовника, менеджер проекту, технічний керівник проекту, член проектної команди.

Методи прийняття рішень — експертна оцінка (використовується для отримання оцінок розміру проекту в цілому та окремих його частин, також цей метод застосовувався для безпосереднього оцінювання зусиль та розкладу виконання проекту), експертна оцінка із розбиттям проекту на підзадачі.

Неголіки існуючого підходу:

- немає визначеної схеми взаємодії ролей та прийняття рішень під час оцінювання проектів;
- використовуються виключно методи експертних оцінок;
- не використовуються методи визначення розміру проектів та не встановлено внутрішні стандарти на оцінки розміру.

Вимоги до створюваної методики оцінювання. Така методика повинна:

- давати оцінки для розміру, бюджету, вартості та тривалості проекту;
- бути застосовною на всіх етапах виконання проекту та на етапі розгляду його доцільності;
- якнайменше залежати від експертних оцінок;
- використовувати базу даних параметрів раніше виконаних проектів;
- визначати процедуру внесення змін у саму методику.

2.1. Підхід до створення методики оцінювання. Аналіз застосовності досліджених методів та їх використання. Як уже було відмічено, найбільш важливими параметрами проекту для управління є його розмір, трудомісткість, час, необхідний для виконання, та вартість. Серед них три останні залежать від першого — розміру. Тому можна виділити два етапи:

- оцінювання розміру;
- оцінювання трудомісткості, вартості та тривалості з використанням вже отриманого значення розміру.

Для 2-го етапу добре підходить модель СОСОМО, оскільки вона розроблялася саме з метою оцінки бюджету, вартості та тривалості проектів і є зараз найбільш широко застосовуваною моделлю. Модель СОСОМО використовує як вхідний параметр розмір ПС у рядках вихідного коду (SLOC). Отримати точний розмір у таких одиницях можна лише після завершення проекту. На будь-якому попередньому етапі розмір має визначатися за допомогою експертних оцінок або деякої методики оцінювання розміру проекту, яка може дати оцінку в одиницях SLOC. У той же час найкращим вибором для оцінювання розміру проекту є метод FPA: він має об'єктивний характер, простий у застосуванні, найбільш широко використовуваний в індустрії розробки ПЗ, має статус міжнародного стандарту та загально доступний. Розмір ПС, отриманий за допомогою FPA, можна використати у моделі СОСОМО, оскільки для багатьох мов програмування існують залежності між розміром ПС у одиницях UFP та SLOC [2].

Існуючі методи експертних оцінок також включені в схему, оскільки

застосування більш формалізованих методів у певних випадках може бути неможливим або невигідним.

Схема поєднання згаданих методів приведена на рис. 2.

Організаційні аспекти підходу до оцінювання проектів. Запропонована організаційна схема наведена на рис. 3, де процес оцінювання ділиться на чотири етапи:

а) *підготовка* — вибір методу оцінювання, планування процесу оцінювання;

б) *виконання* — підготовка вхідних даних, оцінювання розміру, зусиль та вартості, термінів виконання проекту, підготовка звіту про оцінювання;

в) *перевірка* — контроль отриманих результатів, збір даних про процес оцінювання;

г) *аналіз та покращення процесу оцінювання* — зміна процесу оцінювання.

Особливості застосування методу FPA. Для успішного використання в пропонованому підході методу FPA необхідно попередньо оцінити такі його характеристики: *стабільність оцінок*, які дає цей метод, *їх узгодженість зі*

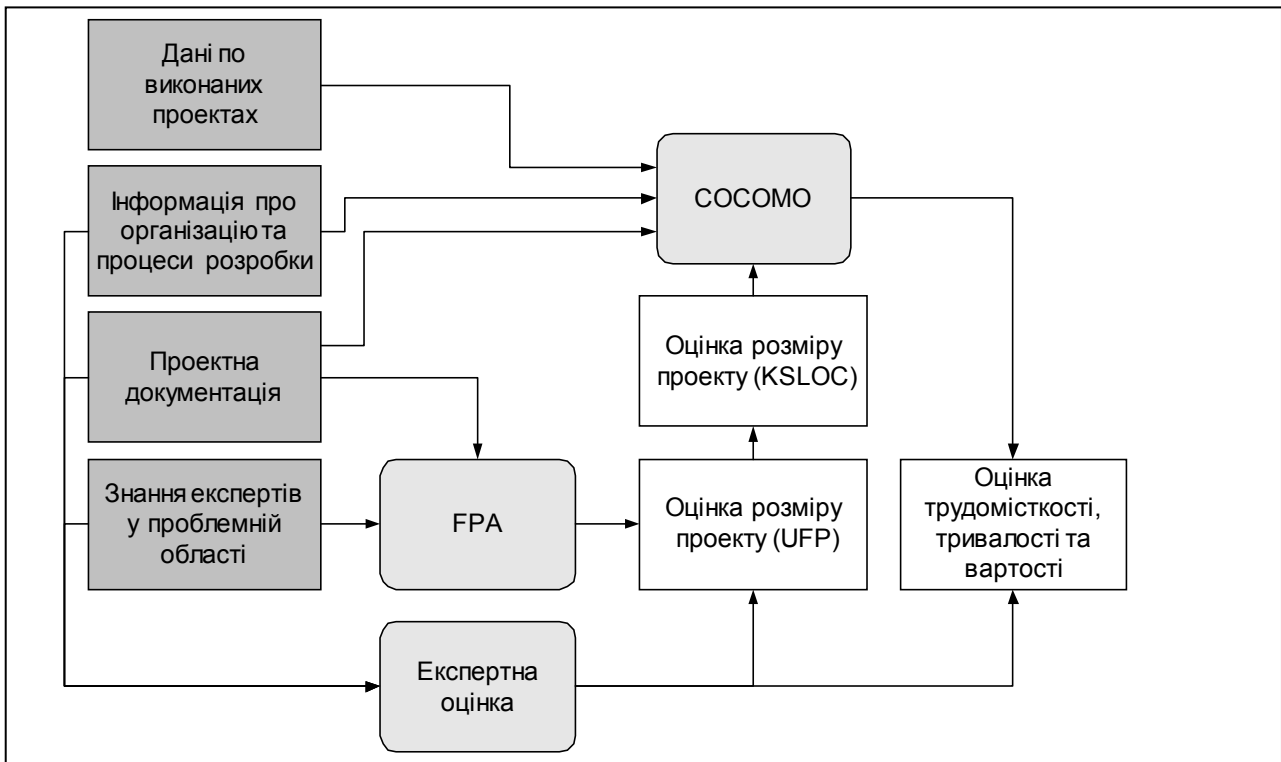


Рис. 2. Покриття цілей оцінювання за допомогою методів FPA та СОСОМО.

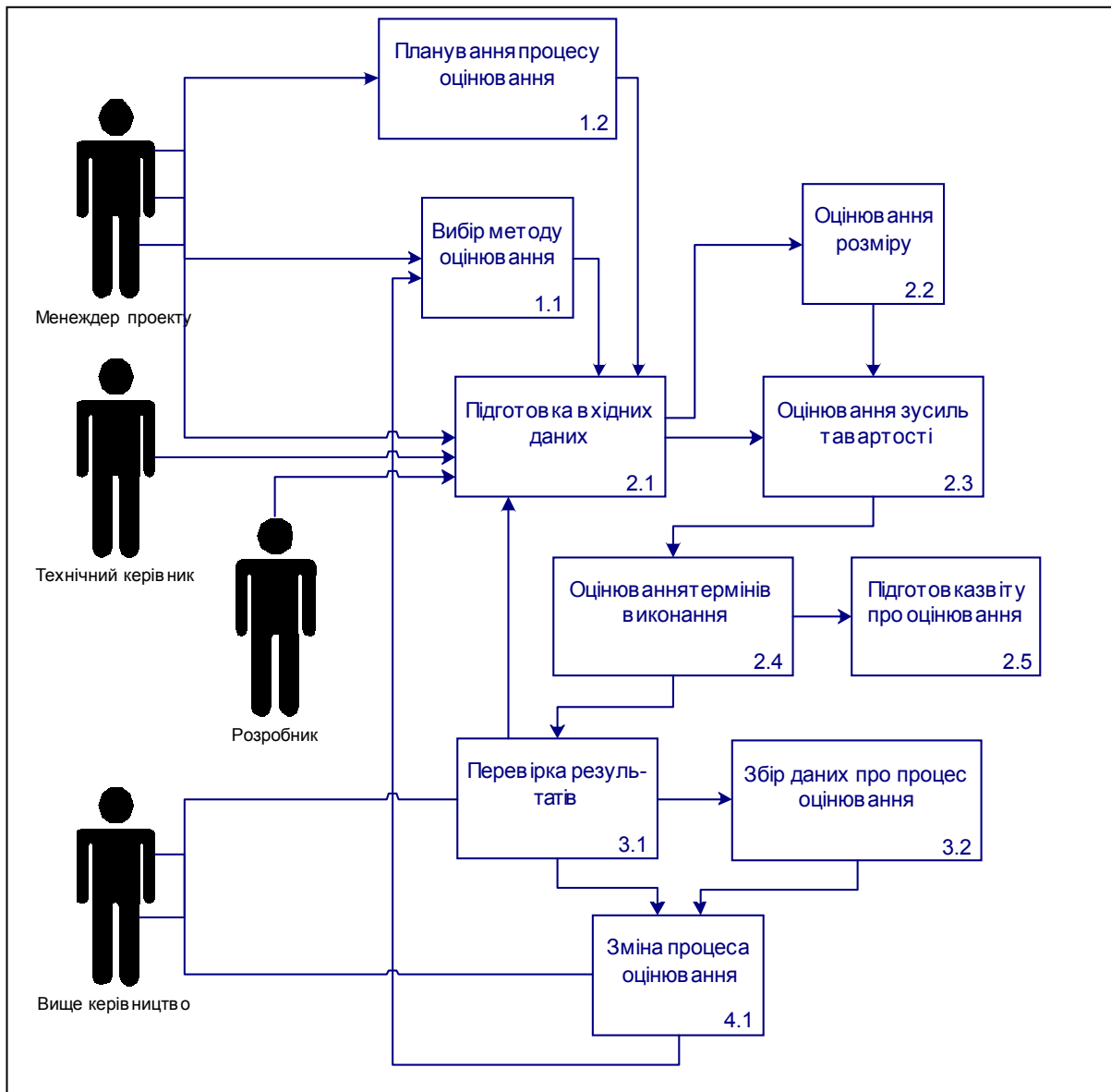


Рис. 3. Повна схема процесу оцінювання проекту

статистикою по індустрії ПС, узгодженість оцінок між проектами.

Підхід використовує припущення про існування співвідношення розмірів проектів в одиницях SLOC та UFP. Причому величина цього співвідношення для різних проектів однакова. Ці припущення були підтверджено експериментально.

Особливості застосування моделі СОСОМО. Модель СОСОМО дає змогу отримати більш точні результати, якщо відкалібрувати її за даними по історичних проектах, тобто визначити такі значення параметрів моделі, які б відображали особливості виконання проектів в умовах конкретної організації.

У рівнянні СОСОМО для оцінки бюджету проекту присутні дві калібровочні константи A та B .

Допустимо, є дані по n проектах. Нехай $j = \overline{1, n}$ — порядковий номер проекту. Після логарифмування та очевидних перетворень рівняння (1) матиме такий вигляд:

$$\log A + B \times \log \text{Size}_j = \log PM_j - \log \left(\prod_{i=1}^{17} EM_{ij} \right) - \left(0.01 \times \sum_{i=1}^5 W_{ij} \right) \times \log \text{Size}_j .$$

Позначимо

$$D_j = \log PM_j - \log \left(\prod_{i=1}^{17} EM_{ij} \right) - \left(0.01 \times \sum_{i=1}^5 W_{ij} \right) \times \log Size_j,$$

тоді $\log A + B \times \log Size_j = D_j$ або у векторному вигляді

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \log Size_1 & \log Size_2 & \dots & \log Size_n \end{bmatrix} \begin{bmatrix} \log A & B \end{bmatrix} = \begin{bmatrix} D_1 & D_2 & \dots & D_n \end{bmatrix},$$

Якщо ввести позначення

$$Y = \begin{bmatrix} D_1 & D_2 & \dots & D_n \end{bmatrix},$$

$$X = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \log Size_1 & \log Size_2 & \dots & \log Size_n \end{bmatrix},$$

$$K = \begin{bmatrix} \log A & B \end{bmatrix},$$

то матимемо $XK = Y$, звідки можна отримати $K = (X^T X)^{-1} X^T Y$ або

$$\log A = \frac{1}{Det} \left(\left(\sum_{j=1}^n D_j \log Size_j \right) \times \left(\sum_{j=1}^n \log Size_j \right) - \left(\sum_{j=1}^n D_j \right) \times \left(\sum_{j=1}^n \log Size_j^2 \right) \right),$$

$$B = \frac{1}{Det} \left(\left(\sum_{j=1}^n D_j \right) \left(\sum_{j=1}^n \log Size_j \right) - n \times \left(\sum_{j=1}^n D_j \log Size_j \right) \right),$$

$$Det = \det(X^T X) = \left(\sum_{j=1}^n \log Size_j \right)^2 - n \times \left(\sum_{j=1}^n \log Size_j^2 \right).$$

Таким чином, можна отримати значення калібровочних констант для певних умов виконання проектів.

Комплексну схему виконання пропонованого підходу відображає рис. 4.

3. Експериментальна перевірка застосовності пропонованого підходу

Було проведено оцінювання економічних характеристик декількох вибраних проектів, що виконувалися в організації.

Оцінка ефективності застосування методу FPA. В експерименті брали участь декілька експертів. Підбір проектів та експертів проводився за такими критеріями:

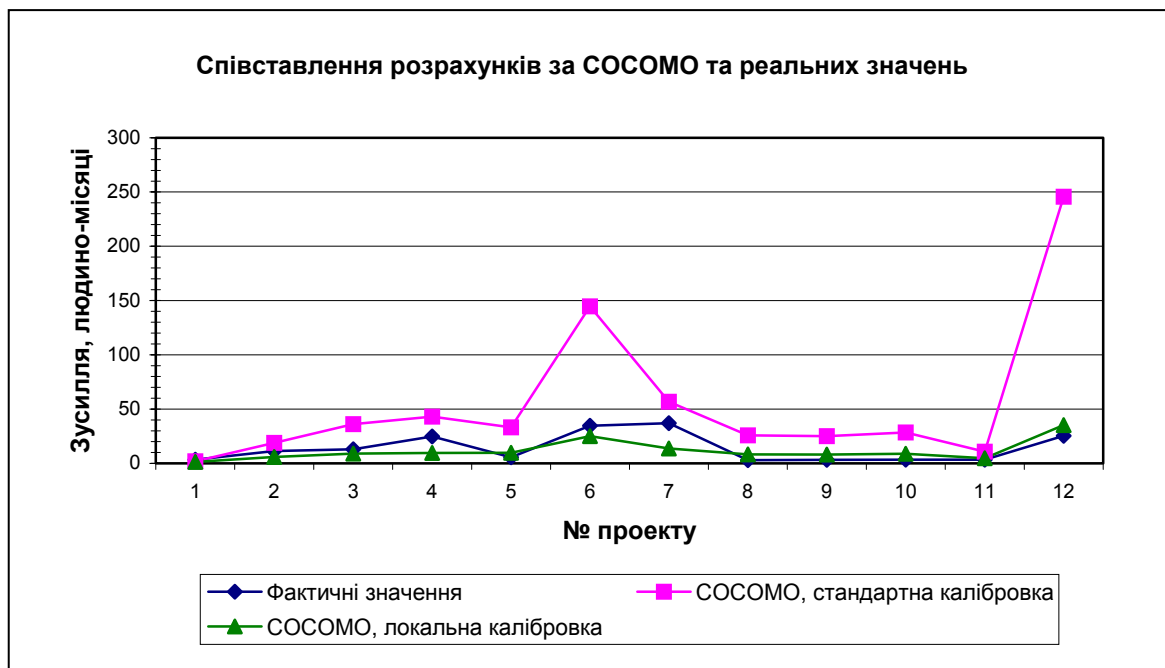


Рис. 4. Співставлення розрахунків за моделлю СОСОМО з фактичними даними

- проект реалізовано на мові C++ або Java;
- експерт мав досвід у відповідній прикладній області або мав змогу звернутися до розробника проекту за поясненнями;

- процес оцінювання мав покрити весь діапазон розмірів існуючих проектів.

В результаті було отримано такі дані (табл. 1):

Таблиця 1. Результати оцінювання проектів за методом FPA

№ проекту	1	2	3	4	5
Розмір, SLOC	8377	7321	18862	15425	27123
Розмір, UFP (1)	126	119	461	337	456
Розмір, UFP (2)	122	107	–	–	378
Розмір, UFP (3)	90	–	–	–	–
Середнє значення	112.67	113.00	–	–	417.00
СКВ	16.11	6.00	–	–	39.00
Відносне СКВ	14%	5%	–	–	9%
Мова програмування	C++	C++	Java	C++	Java

В табл. 1 наведено дані по п'яти проектах, які незалежно оцінювали від одного до трьох експертів. Розбіжності у результатах достатньо малі.

Якщо згрупувати оцінки за мовою програмування, то отримаємо наступні результати (табл. 2):

Таблиця 2. Статистика відношення SLOC/UFP для мов програмування

№ оцінки	C++	Java
1	66.48	40.92
2	68.66	59.48
3	93.08	71.75
4	61.52	–
5	68.42	–
6	45.77	–
Середнє значення	67.32	57.38
СКВ	13.94	12.68
Відносне СКВ	23%	21%

Враховуючи точність методу FPA, можна стверджувати, що між розміром системи в одиницях UFP та SLOC існує достатньо чітка відповідність.

Оцінка ефективності застосування методу СОСОМО. Проекти відбиралися за такими критеріями:

метою проекту було створення, модернізація чи реінжиніринг програмної системи;

проект розроблявся з використанням не більше двох різних мов програмування;

проект використовував такі мови програмування, для яких чітко визначені правила підрахунку кількості рядків;

проект є зовнішнім замовленням. Ці умови найбільш відповідають тим, для яких розроблялася модель СОСОМО.

Таблиця 3. Вихідні дані для калібрування COSOMO

№ проекту	Зусилля	PM	KSLOC	EAF	SF
1	514	3.38	1.20	0.54	25.35
2	1718	11.30	8.38	0.54	25.35
3	1976	13.00	14.64	0.54	25.35
4	3758	24.72	18.86	0.48	25.35
5	848	5.58	9.71	0.80	25.35
6	5265	34.64	34.46	0.80	25.35
7	5635	37.07	15.43	0.80	25.35
8	438	2.88	7.82	0.80	25.35
9	485	3.19	7.63	0.80	25.35
10	512	3.37	8.50	0.80	25.35
11	521	3.43	3.68	0.80	25.35
12	3843	25.28	54.35	0.80	25.35

Методика визначення параметрів проектів для моделі COSOMO. Дані по параметрах проектів наведено у табл. 3. У стовпчику "№ проекту" зазначений його номер у списку. Назви не наводяться через конфіденційність. Величини зусиль на виконання проекту було взято з проектної документації. Ці значення відображають фактично витрачений робочий час у людино-годинах. Значення змінної PM (людино-місяці) було розраховано, виходячи із зусиль у людино-годинах та типової кількості робочих годин за місяць — 152, згідно [11].

Значення SLOC отримано шляхом підрахунку кількості рядків всього вихідного коду, який розроблено та модифіковано у ході виконання проекту. Для цього було використано пакет CodeCount™, створений у Центрі програмної інженерії Університету Південної Каліфорнії.

Значення змінних EAF та SF визначалися за результатами опиту менеджерів відповідних проектів. Однакові значення змінних EAF відповідають схожим умовам розробки проектів з огляду на досвід проектної команди та якості вимог до проекту. Зменшення цих значень для певних проектів пояснюється наявністю досвіду, який було отримано під час розробки попередніх проектів, завдяки чому були розробле-

ні більш чіткі та повні вимоги до проектів, покращилося розуміння розробниками прикладної області.

Змінна SF характеризує організацію та її персонал в цілому, тому її значення для різних проектів однакові, оскільки у період, протягом якого розроблялися обрані проекти, суттєвих змін в організації, проектних командах або процесах розробки не відбувалося.

Калібрування моделі COSOMO та аналіз результатів. Розрахунки проводилися за вищенаведеними формулами. Отримані результати зведені у табл. 4 (стовпчик „Локальна калібровка”).

Таблиця 4. Результати калібрування моделі COSOMO

Константа	Стандартна калібровка	Локальна калібровка
A	2.54	2.23
B	0.91	0.49

Для порівняння у стовпчику "Стандартна калібровка" наведені значення констант для стандартної калібровки моделі COSOMO.

За результатами калібрування розраховані зусилля на розробку проектів. Отримані значення, результати обчислень за стандартною калібровою, фактичні дані та відхилення у від-

Таблиця 5. Результати обчислень за каліброваною моделлю

№ п/п	PM фактичне	PM, стандартна калібровка	Відхилення, %, стандартна калібровка	PM, локальна калібровка	Відхилення, %, локальна калібровка
1	3.38	1.96	42%	1.37	60%
2	11.30	18.83	67%	6.40	43%
3	13.00	16.09	24%	5.75	56%
4	24.72	43.03	74%	10.84	56%
5	5.58	33.11	493%	10.66	91%
6	34.64	144.57	317%	29.14	16%
7	37.07	56.75	53%	15.40	58%
8	2.88	25.73	793%	8.97	211%
9	3.19	25.03	685%	8.81	176%
10	3.37	28.38	742%	9.60	185%
11	3.43	10.72	213%	4.94	44%
12	25.28	109.66	334%	24.13	5%

сотках від фактичного значення подані у табл. 5.

Ці ж результати у графічному вигляді подані на рис. 4. Можна бачити, що калібрування моделі за базою історичних проектів дозволяє отримати оцінки, у більшості випадків більш близькі до фактичних, ніж з використанням стандартної калібровки. Але навіть для моделі з локальною калібруванням існує розкид відхилень оцінок від фактичних значень (стовпчик "Відхи-

лення, %, локальна калібровка"). Існують наступні можливі причини таких відхилень: мала вибірка, шуми або «викиди» вихідних даних, різні мови програмування проектів.

Для перевірки третьої причини, впливу специфіки мови програмування, модель СОСОМО було відкалібровано за двома різними вибірками проектів для мов C++ та Java. Результати калібрування наведено у табл. 6.

Таблиця 6. Результати калібрування моделі СОСОМО для різних мов програмування

Константа	Стандартна калібровка	Локальна калібровка	Калібровка для мови C++	Калібровка для мови Java
A	2.54	2.23	5.44	0.59
B	0.91	0.49	0.41	0.88

Таблиця 7. Результати калібрування моделі СОСОМО для мови C++ у порівнянні з калібруванням для всіх проектів

№ п/п	PM фактичне	PM, калібровка по всіх проектах	Відхилення, %, для всіх проектів	PM, калібровка по C++ - проектах	Відхилення, %, для C++ - проектів
1	3.38	1.38	59%	3.31	2%
2	11.30	5.90	48%	12.00	6%
3	13.00	5.33	59%	10.98	16%
4	37.07	13.78	63%	26.65	28%

Таблиця 8. Результати калібрування моделі СОСОМО для мови Java у порівнянні з калібрувкою для всіх проектів

№ п/п	PM фактичне	PM, калібровка по всіх проектах	Відхилення, %, для всіх проектів	PM, калібровка по Java - проектах	Відхилення, %, для Java-проектів
1	24.72	9.61	61%	7.79	68%
2	5.58	9.75	75%	6.13	10%
3	34.64	25.12	27%	25.66	26%
4	2.88	8.30	188%	4.80	67%
5	3.19	8.15	155%	4.68	47%
6	3.37	8.84	162%	5.28	57%
7	3.43	4.73	38%	2.05	40%
8	25.28	35.30	40%	42.93	70%

Видно, що параметри моделі для різних мов суттєво відрізняються. Результати застосування відкаліброваних моделей наведено у табл. 7, 8 та у графічному вигляді на рис. 5, 6. Для мови C++ максимальне відхилення становить 28%, що є прийнятним для використання, проте результати не можуть вважатися достовірними, оскільки отримані за малою кількістю вихідних даних. Результати для мови Java є не такими вражаючими, і максимальне відхилення становить 70%, проте такі оцінки вже можуть використовуватися

як орієнтири для подальшого процесу оцінювання.

Висновки

У статті досліджені існуючі методи оцінювання окремих характеристик проектів, на основі яких, з урахуванням досвіду виконання проектів, запропоновано комплексну методику оцінювання економічних характеристик проектів, що поєднує методи експертних оцінок, FPA та СОСОМО.

Запропоновано та експериментально перевірено можливість викорис-

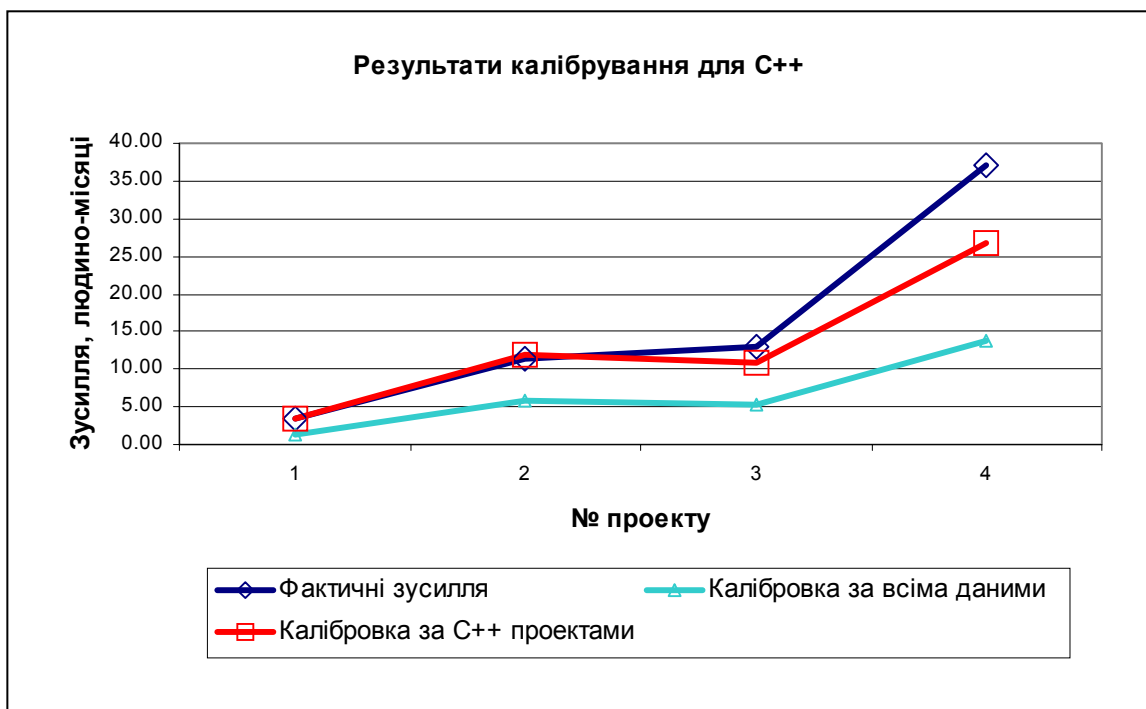


Рис. 5. Співставлення розрахунків для калібрувань за всією вибіркою та за проектами на C++

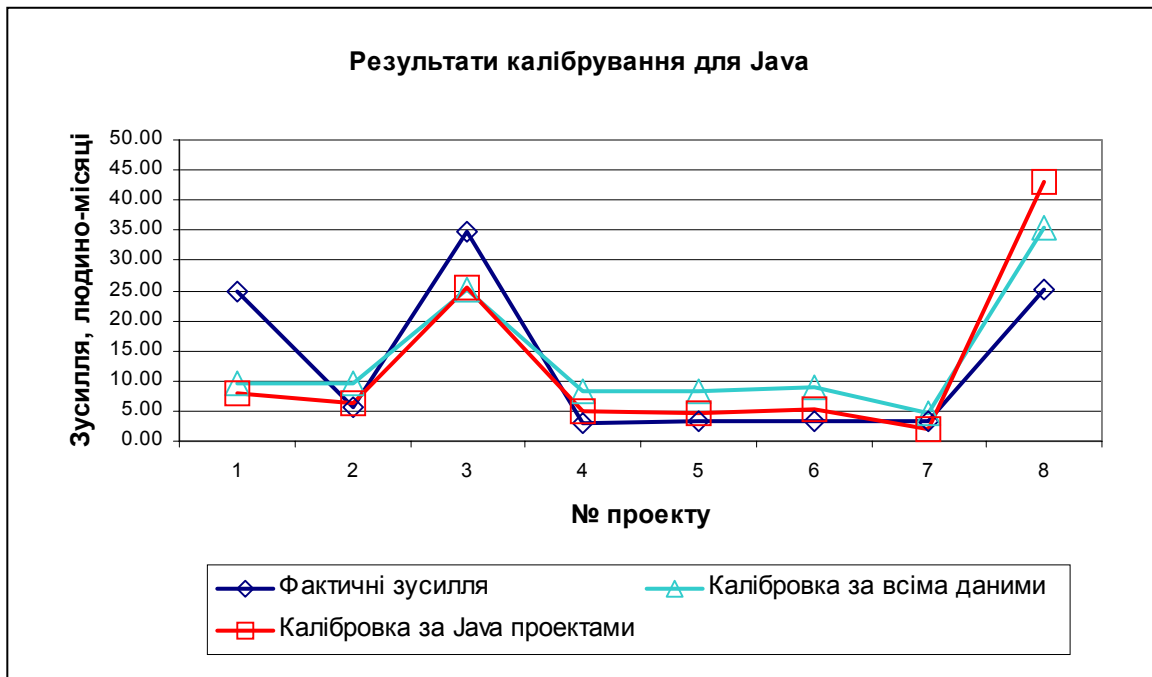


Рис. 6. Співставлення розрахунків для калібровок за всією вибіркою та за проектами на Java

тання оцінок за методом FPA як вхідних даних для моделі СОСОМО. За результатами експериментальних випробувань методики оцінено ефективність застосування методів FPA та СОСОМО на даних по виконаних проектах і показано, що розбиття даних на класи для калібрування моделі СОСОМО підвищує точність оцінок.

Використання формалізованої методики оцінювання економічних характеристик проектів дає певні переваги: по-перше, її застосування зменшує витрати на оцінювання проектів, по-друге, при систематичному її використанні можливе покращення точності оцінок за рахунок уточнення моделей та їх параметрів на даних по виконуваних проектах а, фактично, уточнення методики.

Крім того, впровадження формалізованих методів оцінювання має наступні позитивні особливості, які не є очевидними:

- при проведенні аналізу системи з огляду на оцінки розміру виявляються помилки, незрозумілі місця та неповні описи у вимогах до системи, що дозволяє зменшити кількість дефектів у системі на самому ранньому етапі розробки;

- спеціаліст, який має досвід в оцінюванні проектів за цими методами, буде здатний розробляти вимоги до наступних проектів із урахуванням необхідності їх вимірювання, а як результат — вимоги будуть більш структурованими, легше піддаватимуться вимірюванню та міститимуть менше дефектів;

- наявність певної структури результуючих документів дає можливість створити шаблони документації, які автоматизують розрахунки та конвертацію одиниць вимірювання, використовуваних різними методами;

- документи, створені за цими шаблонами, можуть слугувати також засобом накопичення даних по історичних проектах. Наявність певної структури в таких документах дає змогу автоматизувати збір цих даних.

1. *Игнатенко П.П.* Проблемы обеспечения жизнеспособности программных систем та підходи до їх вирішення // Пробл. програм. — 2002. — №3–4. — С. 58–73.
2. *Основы инженерии качества программных систем.* / Андон Ф.И., Коваль Г.И., Коротун Т.М., Суслов В.Ю. — К.: Академперіодика, 2002. — 504 с.
3. *Методика определения затрат на создание автоматизированных систем (версия 1.1)* / Волощук А.С., Коваль Г.И., Коротун Т.М., Суслов В.Ю., Портяной В.С., Слабоспиц-

- кая О.С. — К.: Институт программных систем, 1998. — 53с.
4. Boehm B.W., Abts C., Chulani S. Software development cost estimation approaches. A survey // Annals of Software Engineering. — 2000. — 10. — P. 177–205.
 5. Devnani-Chulani S. Bayesian Analysis of Software Cost and Quality Models — dissertation // Computer Science Department, USC Center for Software Engineering, 1999. — 231 p.
 6. Turoff M., Linstone H.A. The Delphi Method Techniques and Applications. — Los Angeles, CA.: University of Southern California, 1992. — 618 p.
 7. Park R. Software Size Measurement: A Framework for Counting Source Statements // Tech. Report. — Pittsburg: Software Eng. Inst. CMU/SEI-92-TR-020., 1992. — 242 p.
 8. Longstreet D. Function Points Analysis Training Course // www.softwaremetrics.com, 2002. — 116 p.
 9. KPMG, 2001: KPMG Consulting, Inc.: МК II Function Point Analysis Counting Practices Manual. Version 1.3.1 // www.kpmg.co.uk/-kpmg/uk. — 92 p.
 10. COSMIC, 2003: The Common Software Measurement International Consortium. The COSMIC FFP Measurement Manual. Version 2.2 // www.cosmicon.com. — 81 p.
 11. CSE, 1999: Center for Software Engineering. COCOMO II Model Definition Manual. // Computer Science Department, USC Center for Software Engineering, 1999. — 37 p.
 12. Clark B., Devnani-Chulani S., Boehm B.W. Calibrating the COCOMO II Post-Architecture Model. // ICSE, 1998. — p. 477–480.
 13. CSE, 1999: Center for Software Engineering. COCOMO II Reference Manual. // Computer Science Department, USC Center for Software Engineering, 1999. — 86 p.

Отримано 17.11.03

Про авторів

Стрелов Ігор Анатолійович,
аспірант

Ігнатенко Петро Петрович,
кандидат технічних наук, завідувач відділом

Місце роботи авторів
Інститут програмних систем НАН України,
просп. Академіка Глушкова, 40,
Київ-187, 03680, Україна
Тел. (044) 452 5791, 266 1540
E-mail: ignat@isofts.kiev.ua,
ihors@svitonline.com