

Електронний журнал «Ефективна економіка» включено до переліку наукових фахових видань України з питань економіки (Категорія «Б», Наказ Міністерства освіти і науки України № 975 від 11.07.2019). Спеціальності – 051, 071, 072, 073, 075, 076, 292. Ефективна економіка. 2025. № 9.

DOI: <http://doi.org/10.32702/2307-2105.2025.9.63>

УДК 004.421.2

Л. І. Гончар,

*к. е. н., доцент кафедри комп'ютерних наук,
Західноукраїнський національний університет
ORCID ID: <https://orcid.org/0000-0002-2985-1488>*

М. Я. Шпінталь,

*к. т. н., доцент кафедри комп'ютерних наук,
Західноукраїнський національний університет
ORCID ID: <https://orcid.org/0000-0002-9864-7050>*

О. О. Опалько,

*аспірант, Західноукраїнський національний університет
ORCID ID: <https://orcid.org/0009-0008-5311-9503>*

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ВАРТОСТІ ПРОГРАМНИХ СИСТЕМ НА ПРИКЛАДІ АДАПТОВАНОЇ МОДЕЛІ СОСОМО ДЛЯ ВБУДОВАНИХ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ

L. Honchar,

*PhD in Economics, Associate Professor of the Department of Computer Sciences,
West Ukrainian National University*

M. Shpintal,

*PhD in Technical Sciences, Associate Professor of the Department of Computer
Sciences, West Ukrainian National University*

O. Opalko,

Postgraduate student, West Ukrainian National University

MATHEMATICAL MODELLING OF SOFTWARE SYSTEM COST USING AN ADAPTED COCOMO MODEL FOR EMBEDDED INFORMATION-MEASURING SYSTEMS

У статті досліджено комплексні підходи до оцінки вартості програмних систем на основі систематичної адаптації моделі COSOMO (Constructive Cost Model) для вбудованих інформаційно-вимірювальних систем, що вирішує критичні виклики сучасної економіки програмної інженерії. Проаналізовано фундаментальні характеристики вартості програмного забезпечення, включаючи притаманну динамічність, структурну комплексність та багатогранні складові витрат, що охоплюють аналіз вимог, архітектурне проектування, імплементацію, комплексне тестування, процедури впровадження та довгострокові активності з підтримки. Представлено детальну класифікацію моделей оцінки витрат з особливим акцентом на базові та проміжні варіанти моделі COSOMO, поряд із порівняльним аналізом альтернативних методологій, включаючи аналіз функціональних точок, техніки експертної оцінки та сучасні підходи гнучкої розробки.

Особливу увагу приділено взаємозв'язку вартості та продуктивності в процесах розробки програмного забезпечення, підкреслюючи критичні метрики продуктивності, такі як функціональні точки, швидкість розробки та щільність дефектів, що безпосередньо впливають на економіку проєкту. Дослідження демонструє практичне впровадження через комплексний кейс-стаді, що включає інтеграцію автоматизованого тестування в стартап-середовищі, зосередженому на розробці складного мобільного додатку для фітнес-трекінгу з можливостями безшовної інтеграції носимих пристроїв. Застосунок вимагав обробки даних у реальному часі від різних сенсорів, включаючи пульсометри, акселерометри та GPS-пристрої, представляючи унікальні виклики, характерні для розробки вбудованих систем.

Методологія дослідження використовувала техніки проміжного моделювання COSOMO, спеціально адаптовані для вбудованих проєктів, включаючи п'ятнадцять різних факторів витрат, що враховують характеристики продукту, апаратні обмеження, кваліфікацію персоналу та змінні організаційного середовища. Кількісні результати демонструють значні поліпшення в економіці проєкту, включаючи суттєве 17% зниження загальних

трудозатрат, що перетворюється на приблизно 82 людино-місяці економії витрат, представляючи потенційну фінансову економію понад 240,000 доларів у типових ринкових умовах.

Крім того, впровадження досягло помітного зменшення загальної тривалості проєкту на 1,7 місяця, забезпечуючи конкурентні переваги через прискорене позиціонування виходу на ринок. Впровадження фреймворку автоматизованого тестування призвело до вимірюваного зниження рівня дефектів, циклів регресійного тестування та довгострокових витрат на підтримку, одночасно покращуючи надійність програмного забезпечення та метрики задоволеності користувачів. Дослідження також досліджує трансформаційний потенціал інтеграції сучасних практик DevOps, зокрема пайплайнів безперервної інтеграції та безперервного розгортання (CI/CD), для досягнення додаткового зниження витрат від 5-10% понад початкові переваги автоматизації.

Це дослідження підтверджує ефективність підходів математичного моделювання на основі COCOMO для складних проєктів, що включають вимоги апаратної інтеграції, обмеження обробки в реальному часі та середовища з обмеженими ресурсами, в кінцевому рахунку підтримуючи інформовані процеси управлінського прийняття рішень та підвищуючи організаційну конкурентоспроможність на швидкозмінних технологічних ринках.

The article investigates comprehensive approaches to software system cost estimation through systematic adaptation of the COCOMO (Constructive Cost Model) for embedded information-measuring systems, addressing critical challenges in modern software engineering economics. The research analyzes fundamental cost characteristics of software development, including inherent dynamism, structural complexity, and multifaceted cost components encompassing requirements analysis, architectural design, implementation, comprehensive testing, deployment procedures, and long-term maintenance activities. A detailed classification of cost estimation models is presented with particular emphasis on basic and intermediate COCOMO

model variants, alongside comparative analysis of alternative methodologies including function point analysis, expert judgment techniques, and contemporary Agile development approaches.

Special attention is devoted to the relationship between cost and productivity in software development processes, highlighting critical productivity metrics such as function points, development velocity, and defect density that directly impact project economics. The study demonstrates practical implementation through a comprehensive case study involving automated testing integration within a startup environment focused on developing a complex fitness-tracking mobile application with seamless wearable device integration capabilities. The application required real-time data processing from various sensors including heart rate monitors, accelerometers, and GPS devices, presenting unique challenges characteristic of embedded systems development.

The research methodology employed intermediate COCOMO modeling techniques specifically adapted for embedded projects, incorporating fifteen different cost driver factors that account for product characteristics, hardware constraints, personnel qualifications, and organizational environment variables. Quantitative results demonstrate significant improvements in project economics, including a substantial 17% reduction in overall effort, translating to approximately 82 person-months in cost savings, representing potential financial savings exceeding \$240,000 under typical market conditions.

Additionally, the implementation achieved a notable 1.7-month decrease in total project duration, providing competitive advantages through accelerated time-to-market positioning. The automated testing framework implementation resulted in measurable reductions in defect rates, regression testing cycles, and long-term maintenance costs, while simultaneously improving software reliability and user satisfaction metrics. The research also explores the transformational potential of integrating modern DevOps practices, specifically continuous integration and continuous deployment (CI/CD) pipelines, for achieving additional cost reductions of 5-10% beyond the initial automation benefits.

This study confirms the effectiveness of COCOMO-based mathematical modeling approaches for complex projects involving hardware integration requirements, real-time processing constraints, and resource-limited environments, ultimately supporting informed management decision-making processes and enhancing organizational competitiveness in rapidly evolving technology markets.

Ключові слова: *модель COCOMO, оцінка вартості, вбудовані системи, продуктивність, автоматизоване тестування, програмне забезпечення, програмування, проектування, DevOps, моделювання, прогнозування.*

Keywords: *COCOMO model, cost estimation, embedded systems, productivity, automated testing, software, programming, design, DevOps, modeling, forecasting.*

Вступ. У сучасних умовах цифрової трансформації проблема ефективного управління вартістю програмного забезпечення набула особливої актуальності. Зростаюча складність програмних продуктів, поява нових технологій, а також постійне скорочення бюджетів ІТ-проектів вимагають точних та обґрунтованих методів оцінки витрат. Особливо це стосується вбудованих інформаційно-вимірювальних систем, де поєднуються високі вимоги до надійності, функціональності та апаратного середовища.

Одним із найбільш визнаних підходів до моделювання вартості є конструктивна модель COCOMO (Constructive Cost Model), запропонована Баррі Боемом ще у 1981 році [1]. Попри свій вік, модель залишається актуальною, адже дозволяє враховувати як базові параметри (кількість рядків коду), так і низку критичних факторів витрат (Cost Drivers). У її адаптованому вигляді COCOMO успішно застосовується для вбудованих проєктів, де необхідна точна оцінка трудомісткості розробки в умовах обмежених ресурсів та складних середовищ.

Модель складається з ієрархії трьох послідовно деталізованих та уточнюючих рівнів (режимів). На кожному рівні всі проєкти розбиваються на три групи за рівнем складності:

1. Розповсюджений або органічний тип (organic projects).
2. Напівнезалежний або напіврозподілений тип (semidetached projects).
3. Вбудований тип (embedded projects).

Розповсюджений або органічний (organic) рівень звичайно класифікується як платіжна відомість, опис або наукове обчислення. Напівнезалежний або напіврозподілений (semidetach) рівень типізується прикладними системами, наприклад, компіляторами, системами баз даних або редакторами. Вбудований (embedded) рівень характеризується режимами реального часу, наприклад, системами контролю повітряного руху, мережами АТМ або воєнними системами. Середовище розробки в цьому випадку складається з багатьох складних інтерфейсів, включаючи ті з них, які поставляються замовникам разом із апаратним забезпеченням. Тип тієї чи іншої групи можна розглядати як один з параметрів моделі COCOMO.

У статті розглядається застосування моделі COCOMO до проєктів зі створення вбудованих інформаційно-вимірювальних систем, аналізуються ключові фактори, що впливають на вартість програмного забезпечення, та досліджується взаємозв'язок між витратами і продуктивністю розробки. Для демонстрації практичної ефективності математичного підходу до моделювання витрат проаналізовано кейс впровадження автоматизованого тестування у стартапі з розробки мобільного застосунку.

Аналіз літературних даних і постановка проблеми. Оцінка вартості програмного забезпечення є важливою складовою життєвого циклу будь-якого ІТ-проєкту. У науковій літературі ця проблема розглядається в контексті прогнозування ресурсів, управління ризиками, вибору стратегії розробки та контролю за якістю продукту. У низці досліджень [2–4] підкреслюється, що недооцінка або переоцінка витрат може призвести до зриву проєкту, перевитрати бюджету або отримання нерентабельного продукту.

Серед класичних моделей оцінки витрат особливе місце посідає COCOMO – конструктивна модель, яка дозволяє обчислювати трудомісткість та тривалість проєкту залежно від розміру коду, типу системи та додаткових факторів.

Первинна версія моделі була презентована у 1981 році Баррі Боемом [1], а у 2002 році була розширена до версії COSOMO II [5], яка враховує сучасні типи проєктів та процеси розробки.

Однак традиційна модель COSOMO має обмеження щодо точності в оцінці проєктів зі складною апаратною інтеграцією, високими вимогами до реального часу або специфічними середовищами – саме такими є вбудовані інформаційно-вимірювальні системи. У таких системах необхідно враховувати не лише програмну складову, але й взаємодію з сенсорами, контролерами, інтерфейсами обміну даними та іншими апаратними компонентами. Це створює додаткові виклики для моделювання вартості та продуктивності.

У сучасних публікаціях [6,7] пропонуються різні модифікації COSOMO для специфічних умов, зокрема для веб-систем, мобільних застосунків або проєктів з використанням Agile. Проте адаптація COSOMO саме до вбудованих систем вимагає глибшого аналізу та експериментального обґрунтування.

Сучасна еволюція моделі COSOMO представлена у вигляді COSOMO II, яка значно розширює базові можливості початкової версії. Зокрема, модель містить три рівні деталізації: ранній дизайн (early design), післяархітектурний рівень (post-architecture) та рівень композиції застосунків (application composition), що дозволяє більш точно адаптувати модель до різних фаз проєкту [5]. Крім того, COSOMO II підтримує оцінювання effort'у з урахуванням повторного використання компонентів, сучасних інструментів розробки (CASE-засобів), об'єктно-орієнтованих підходів та гнучких методологій.

Дослідники також пропонують модифікації моделі, що поєднують COSOMO з Agile-практиками, наприклад, через інтеграцію story points у формулу оцінки [6]. Проте ці підходи найчастіше орієнтовані на інформаційні системи або мобільні додатки, не враховуючи специфіку вбудованого програмного забезпечення, де чинниками стають обмеження по енергоспоживанню, реальний час реакції, обробка сигналів тощо.

Альтернативні методи, такі як Function Point Analysis (FPA) або Use Case Points, часто використовуються на ранніх етапах проєкту, коли специфікації

програмного забезпечення ще не деталізовані. Однак для вбудованих систем, де архітектурні та технічні деталі відіграють ключову роль у визначенні обсягу робіт, ці методи демонструють нижчу точність. FPA, наприклад, не враховує складність обробки в реальному часі або взаємодію з апаратними пристроями, що знижує її придатність у цьому контексті [7].

Таким чином, незважаючи на широкий спектр наявних моделей, існує потреба в адаптації методів математичного моделювання вартості до проєктів зі специфічними характеристиками, включаючи високу апаратну інтегрованість, суворі обмеження по часу виконання, та складну архітектуру, що й характерно для інформаційно-вимірювальних систем.

Мета і завдання дослідження. Метою даної роботи є дослідження можливостей адаптації моделі COSOMO для точного математичного моделювання вартості програмних систем у сфері вбудованих інформаційно-вимірювальних рішень.

Для досягнення поставленої мети були сформульовані такі завдання:

1. Проаналізувати існуючі підходи до оцінки вартості програмного забезпечення, зокрема моделі COSOMO.
2. Дослідити специфіку вбудованих систем у контексті оцінки витрат.
3. Адаптувати класичну модель COSOMO для умов проєктування та розробки вбудованих інформаційно-вимірювальних систем.
4. Продемонструвати практичне застосування адаптованої моделі на прикладі стартап-проєкту.
5. Оцінити вплив продуктивності команди на зміну вартості розробки ПЗ.

Методи і матеріали дослідження. У цьому дослідженні було застосовано підхід до моделювання вартості програмного забезпечення на основі проміжного рівня моделі COSOMO (Intermediate COSOMO), який дозволяє враховувати специфіку розробки складних програмних систем із глибокою апаратною інтеграцією. Такий підхід виявився особливо доречним для оцінки проєктів у сфері вбудованих інформаційно-вимірювальних систем, де поряд із

програмною логікою критичне значення мають технічні обмеження, інтерфейси, а також вимоги до надійності та продуктивності в реальному часі.

Оцінювання здійснювалося на основі кількісного аналізу трудомісткості, яка у моделі COCOMO розраховується за формулою (1):

$$PM = a_i * (SIZE)^{b_i} * EAF \quad (1)$$

де PM – трудомісткість (люд.- міс.);

SIZE – це обсяг програмного продукту в тисячах рядків вихідного тексту (Kilo of Source Line of Code – KSLOC);

EAF – це множник коригування зусиль, який враховує до п'ятнадцяти факторів (так званих cost drivers), а коефіцієнти a_i і b_i залежать від типу проєкту.

У даному дослідженні, враховуючи апаратну залежність, жорсткі часові обмеження та складну архітектуру, проєкт класифікується як вбудований (embedded).

Параметри для оцінки зусиль були відібрані відповідно до рекомендацій Боєма, зокрема було враховано характеристики продукту (надійність, складність, обсяг баз даних), апаратного забезпечення (обмеження за часом виконання та пам'яттю), персоналу (кваліфікація, досвід) та організаційного середовища (інструменти розробки, вимоги до графіка). Значення факторів оцінювались на основі середніх або помірно високих значень, характерних для проєктів із участю невеликих, але досвідчених команд.

Практичну частину дослідження проведено на прикладі стартап-проєкту зі створення мобільного додатку для фітнес-трекінгу, що має інтеграцію з носимими пристроями (браслети, пульсометри) через Bluetooth. Проєкт передбачав реалізацію функціоналу реального часу, що потребувало зниження затримок при обробці сенсорних даних. Команда складалась із п'яти розробників, одного інженера з тестування та менеджера проєкту, а загальний обсяг вихідного коду становив орієнтовно 35 тисяч рядків.

Особливу увагу в аналізі було приділено впливу автоматизації тестування на загальні витрати. Автоматизоване тестування, впроваджене за допомогою фреймворку Arrium, дозволило суттєво скоротити час на регресійні перевірки

та зменшити кількість помилок після релізів. Це дало змогу оцінити, наскільки зміна підходів до контролю якості впливає на продуктивність команди й, відповідно, на вартість розробки.

Результати досліджень. Застосування моделі COCOMO у дослідженні дозволило отримати кількісні оцінки вартості розробки програмної системи у контексті вбудованого програмного забезпечення з інтеграцією до апаратної частини. Було проведено розрахунки з використанням вихідних даних кейс-проєкту: 35 KLOC, середній рівень складності системи, а також обрані значення факторів витрат відповідно до специфіки проєкту. Значення коефіцієнтів для типу embedded згідно з оригінальною специфікацією COCOMO склали $a=3,6$ та $b=1,2$ [1].

Для вихідної ситуації, до впровадження автоматизованого тестування, середній коефіцієнт EAF (Effort Adjustment Factor), розрахований на основі оцінок cost drivers (включаючи високі вимоги до надійності, складності, та обмеження за часом виконання), становив приблизно 1,35.

Розрахована трудомісткість при цьому складає:

$$PM = 3,6 * (35)^{1.20} * 1,35 \approx 3,6 * 99,5 * 1,35 \approx 483,8 \text{ людино – місяців}$$

Після впровадження автоматизованого тестування відбулося суттєве зменшення показників за деякими cost drivers – зокрема, зменшився вплив фактора SCED (дотримання графіка розробки), а також TOOL (інструментарій розробки) було оцінено на вищому рівні, що знизило EAF до приблизно 1,12.

Повторний розрахунок показав наступне:

$$PM = 3,6 * (35)^{1.20} * 1,12 \approx 3,6 * 99,5 * 1,12 \approx 401,5 \text{ людино – місяців}$$

Таким чином, економія трудових витрат склала близько 82 людино-місяці, що відповідає 17% зменшенню загальних витрат на розробку. Якщо перевести це в умовні фінансові одиниці (наприклад, при середньому бюджеті \$3000/люд.-міс.), скорочення вартості проєкту могло б сягати понад \$240,000.

Окрім зниження трудомісткості, спостерігалось також зменшення тривалості проєкту. Використаємо формулу для часу розробки (коефіцієнти для embedded-прєктів згідно з базовою моделлю) (2):

$$TM = 2,5 * (PM)^{0,38} \quad (2)$$

В такому випадку отримуємо:

1. До автоматизації $TM \approx 2,5 * (483.8)^{0,38} \approx 21.7$ місяців.
2. Після автоматизації $TM \approx 2,5 * (401.5)^{0,38} \approx 20.0$ місяців.

Скорочення тривалості на 1,7 місяця створює додатковий ефект у вигляді швидшого виходу на ринок і зниження операційних витрат у фазі дохідної реалізації.

Обговорення результатів. Отримані результати підтверджують ключові положення теорії програмної інженерії щодо впливу організаційних та технологічних факторів на вартість розробки програмного забезпечення. Зменшення трудомісткості на 17% у результаті впровадження автоматизованого тестування є не лише економічно значущим, але й ілюструє дієвість гнучкого управління факторами витрат у межах моделі COCOMO.

Найбільший ефект був досягнутий завдяки покращенню оцінок за такими параметрами, як TOOL (застосування інструментів автоматизації) та SCED (дотримання графіка). Це демонструє, що навіть відносно прості технологічні зміни можуть мати значний вплив на вартість, особливо у проєктах з високими вимогами до якості та стабільності – таких, як вбудовані інформаційно-вимірювальні системи. З огляду на те, що такі системи зазвичай працюють у критичних режимах (наприклад, медичне обладнання, промислові контролери), мінімізація помилок та зменшення часу на верифікацію є вирішальними для загальної ефективності проєкту.

Варто також відзначити, що в моделі COCOMO кожен окремих cost driver має значний вплив на підсумкову трудомісткість. За даними Chulani, Boehm і Steece [8], навіть одиничне коригування параметра CPLX (складність продукту) або ACAP (здібність аналітиків) може змінити загальний EAF на 10-30%, що суттєво відбивається на загальній вартості проєкту. У даній роботі показовим є те, що покращення лише за двома показниками призвело до зниження загальної трудомісткості на понад 80 людино-місяців.

Такі результати підкреслюють потенціал моделі як інструменту управлінського контролю, а не лише початкового прогнозування. Вони свідчать про необхідність розглядати COSOMO не як статичну формулу, а як частину динамічної системи прийняття рішень у проєктах зі змінними умовами.

Крім того, важливою знахідкою стало спостереження зменшення не лише прямих витрат, але й часових ресурсів. Це може мати прямий вплив на конкурентоспроможність компанії, яка отримує можливість раніше вивести продукт на ринок, зменшуючи time-to-market. У випадку стартапів це часто є вирішальним фактором виживання.

Отже, модель COSOMO в адаптованому вигляді підтверджує свою доцільність для застосування в сучасних умовах. Зокрема, в галузях, де поєднуються традиційна програмна інженерія з апаратною інтеграцією. У цьому контексті особливо цінним є те, що модель дозволяє не лише проводити оцінку «до» початку проєкту, але й здійснювати динамічне коригування прогнозів залежно від змін у процесі розробки.

Варто також зазначити, що впровадження автоматизованого тестування стало не просто технічним рішенням, а інвестицією в довгострокову продуктивність команди. Як показали результати, покращення процесів тестування призвело до зниження кількості помилок, повторних ітерацій та загального технічного боргу – чинників, які безпосередньо впливають на підсумкову вартість програмного продукту.

Окремий напрям адаптації моделей оцінки вартості стосується врахування нових практик DevOps, зокрема – безперервної інтеграції (CI), безперервного розгортання (CD) та автоматизованого моніторингу. В умовах вбудованих систем це дає змогу істотно зменшити час між зміною коду та її перевіркою на реальному пристрої, що знижує ризики пізнього виявлення дефектів. Деякі дослідження пропонують включати показники DevOps-практик до системи cost drivers, розглядаючи їх як такі, що впливають на продуктивність та якість продукту. Наприклад, автоматичні тести, інтегровані з CI/CD, здатні зменшити загальний EAF на 5-10%, що забезпечує суттєву економію при великих

проєктах. Надалі, удосконалення моделей на кшталт COCOMO шляхом врахування метрик з процесів DevOps може стати одним із ключових напрямів досліджень у сфері оцінки витрат.

Висновки. У межах проведеного дослідження було обґрунтовано ефективність застосування адаптованої моделі COCOMO для оцінки вартості програмного забезпечення у проєктах зі створення вбудованих інформаційно-вимірювальних систем. Модель дозволила врахувати не лише обсяг коду, але й низку критичних факторів, що впливають на трудомісткість та тривалість розробки, зокрема – технічну складність, вимоги до продуктивності, кваліфікацію персоналу та використані інструменти.

Вартість програмного забезпечення є одним з ключових факторів , що визначають успішність проєктів у сфері розробки. Під вартістю слід розуміти не лише фінансові витрати на розробку , але й сукупність ресурсів , часу, інтелектуальних зусиль та інфраструктури, необхідних для створення , підтримки й удосконалення ПЗ протягом усього його життєвого циклу. Ключові характеристики вартості включають комплексність, варіативність залежно від етапу розробки, а також залежність від внутрішніх і зовнішніх факторів (технології, кваліфікації персоналу, ринку тощо). До основних компонентів вартості належать: витрати на аналіз вимог, проектування, розробку, тестування, впровадження, підтримку, а також непрямі витрати (наприклад, навчання чи втрати через помилки). У свою чергу, продуктивність у розробці ПЗ відіграє критично важливу роль, адже дозволяє підвищити ефективність використання ресурсів та скоротити вартість розробки без шкоди якості. Продуктивність вимірюється за допомогою спеціальних метрик, таких як функціональні точки, velocity, lead time, щільність дефектів тощо. Ці показники допомагають об'єктивно оцінити результативність роботи команди та планувати майбутні ітерації . Між вартістю і продуктивністю існує взаємозалежність : підвищення продуктивності зазвичай спричиняє зниження витрат , а належне фінансування й інвестування у якісні процеси , інструменти та навчання – підвищує продуктивність . Практичний приклад із впровадження

автоматизованого тестування показав , що навіть невеликі інвестиції можуть мати значний позитивний ефект : зменшення кількості помилок , підвищення швидкості розробки й, відповідно, зниження загальної вартості. Усе це свідчить про необхідність системного підходу до управління вартістю та продуктивністю ПЗ як основи економічної ефективності будь-якого ІТ-проєкту.

Таким чином, адаптоване застосування моделі COCOMO може бути рекомендоване як ефективний інструмент планування та оптимізації ресурсів у проєктах зі складною апаратною взаємодією. Воно сприяє прийняттю обґрунтованих управлінських рішень і забезпечує прозорість процесу оцінки витрат. Подальші дослідження можуть бути зосереджені на моделюванні вартості в умовах гібридних методологій (Agile + DevOps) та з урахуванням впливу машинного навчання на продуктивність команди.

Література

1. Boehm B. W. Software engineering economics. IEEE Transactions on Software Engineering. 1984. № 1. С. 4–21. DOI: 10.1109/TSE.1984.5010193 (дата звернення: 02.07.2025).
2. Jørgensen M. A review of studies on expert estimation of software development effort. Journal of Systems and Software. 2004. Vol. 70, № 1–2. С. 37–60. DOI: 10.1016/S0164-1212(02)00156-5 (дата звернення: 02.07.2025).
3. Chirra S. M. R., Reza H. A survey on software cost estimation techniques. Journal of Software Engineering and Applications. 2019. Vol. 12, № 6. С. 226–248. DOI: 10.4236/jsea.2019.126014 (дата звернення: 03.07.2025).
4. Van Heeringen B. Software size measures and their use in software project estimation. ICEAA Professional Development & Training Workshop, 2015. URL: <https://www.iceaaonline.com/wp-content/uploads/2015/06/SW14-Paper-VanHeeringen-Software-Size-Measures.pdf> (дата звернення: 03.07.2025).
5. Boehm B. Cost estimation with COCOMO II : техн. звіт CS 577a. USC Center for Software Engineering, 2002. URL:

https://www.researchgate.net/publication/228600814_Cost_estimation_with_COCO_MO_II (дата звернення: 04.07.2025).

6. Khan J. A., Khan S. U. R., Khan T., Rehman I. U. An amplified COCOMO II based cost estimation model in global software development context. IEEE Access. 2021. Vol. 9. С. 1–1. DOI: 10.22266/ijies2024.0831.70 (дата звернення: 05.07.2025).

7. Morgenshtern O., Raz T., Dvir D. Factors affecting duration and effort estimation errors in software development projects. Information and Software Technology. 2007. Vol. 49, № 8. С. 827–837. DOI: 10.1016/j.infsof.2006.09.006 (дата звернення: 06.07.2025).

8. Chulani S., Boehm B., Steece B. Bayesian analysis of empirical software engineering cost models. IEEE Transactions on Software Engineering. 1999. Vol. 25, № 4. С. 573–583. DOI: 10.1109/32.799958 (дата звернення: 06.07.2025).

References

1. Boehm, B. W. (1984), “Software engineering economics”, IEEE Transactions on Software Engineering, vol. SE-10, no. 1, 4–21, DOI: 10.1109/TSE.1984.5010193.

2. Jørgensen, M. (2004), “A review of studies on expert estimation of software development effort”, Journal of Systems and Software, vol. 70, no. 1–2, pp. 37–60, DOI: 10.1016/S0164-1212(02)00156-5.

3. Chirra, S. M. R. and Reza, H. (2019), “A survey on software cost estimation techniques”, Journal of Software Engineering and Applications, vol. 12, no. 6, pp. 226–248, DOI: 10.4236/jsea.2019.126014

4. Van Heeringen, B. (2015), “Software size measures and their use in software project estimation”, ICEAA Professional Development & Training Workshop, International Cost Estimating and Analysis Association, San Diego, USA, available at: <https://www.iceaaonline.com/wp-content/uploads/2015/06/SW14-Paper-VanHeeringen-Software-Size-Measures.pdf> (Accessed: 3 July 2025).

5. Boehm, B. (2002), “Cost estimation with COCOMO II”, USC Center for Software Engineering Technical Report CS 577a, University of Southern California,

Los Angeles, USA, available at:
https://www.researchgate.net/publication/228600814_Cost_estimation_with_COCO_MO_II (Accessed: 4 July 2025).

6. Khan, J. A., Khan, S. U. R., Khan, T. and Rehman, I. U. (2021), “An amplified COCOMO II based cost estimation model in global software development context”, IEEE Access, vol. 9, no. 1–1, DOI: 10.1109/ACCESS.2021.3089870.

7. Morgenshtern, O., Raz, T. and Dvir, D. (2007), “Factors affecting duration and effort estimation errors in software development projects”, Information and Software Technology, vol. 49, no. 8, pp. 827–837, DOI: 10.1016/j.infsof.2006.09.006.

8. Chulani, S., Boehm, B. and Steece, B. (1999), “Bayesian analysis of empirical software engineering cost models”, IEEE Transactions on Software Engineering, vol. 25, no. 4, pp. 573–583, DOI: 10.1109/32.799958.

Стаття надійшла до редакції 01.09.2025 р.