

МЕТОД КАЛІБРУВАННЯ МОДЕЛІ СОСОМО ШЛЯХОМ РЕДУКЦІЇ ОСНОВНОГО РІВНЯННЯ

Д.В. Баценко

Національний авіаційний університет, 03058, проспект Космонавта Комарова 1

Розглядаються методи калібрування моделей оцінки вартості програмного забезпечення та особливості калібрування параметрів вартості; пропонується новий метод для більш точного калібрування моделі СОСОМО шляхом варіювання кількості параметрів вартості, робляться висновки про доцільність застосування запропонованого методу.

This article reviews calibration methods used for tuning software cost estimation models, peculiarities of calibration; a new method is proposed for higher accuracy calibration of the COCOMO model based on changing the quantity of effort multipliers; and the conclusions are drawn on the expediency of application of the proposed method.

Вступ

Зростання функціональних можливостей програмного забезпечення (ПЗ) збільшує час розробки та фінансові витрати і потребує розвитку методів точного прогнозування, або оцінки на ранніх етапах, кінцевої вартості ПЗ.

На даний час існує багато методів оцінки вартості ПЗ [1–7]. За принципом оцінки методи поділяються на алгоритмічні та неалгоритмічні [7, 8]. Алгоритмічні методи базуються на математичних моделях, тому їхні результати є передбачуваними та повторюваними, але в багатьох випадках універсальні алгоритмічні моделі (такі як СОСОМО) мають достатньо низьку точність оцінки. Тому для підвищення точності оцінки постійно розробляються нові методи калібрування алгоритмічних моделей, які мають на меті підвищити точність оцінки у кожному конкретному застосуванні, або на певному підприємстві.

У статті розглядається новий підхід до калібрування моделі СОСОМО шляхом на зменшення кількості параметрів моделі, який дозволяє підвищити точність та повторюваність результатів оцінки вартості ПЗ.

Ефективність цього підходу тестується як основна гіпотеза дослідження: "Для будь-якого конкретного домену та організаційної ситуації існує така модель СОСОМО з редукованою кількістю параметрів, якій відповідає вище значення PRED та менша дисперсія, ніж у випадку загальної моделі СОСОМО".

З практичної точки зору, усунення змінних параметрів вартості сприяє підвищенню ефективності моделі вартості, яка підлаштована до існуючих в організації історичних даних та поточних проектів. Однак, якщо організація вносить значні зміни до методики роботи, які стосуються видалених із моделі змінних параметрів вартості (наприклад, перехід від розробки звичайних бізнес-застосувань до розробки систем реального часу та критичних з точки зору безпеки систем), це буде значним ризиком отримання похибок оцінки. Тому метод буде випробуваний із наступним припущенням: "Прогнози щодо точності оцінки правдиві лише для проектів, аналогічних тим, що представлені у калібрувальному масиві даних".

1. Природа проблеми

Точні моделі оцінки вартості ПЗ можуть значно допомогти менеджерам програмних проектів: керівники проектів зможуть приймати обґрунтовані рішення з управління ресурсами, управління та планування проекту, і як наслідок зможуть завершити проект вчасно та в рамках запланованого бюджету, що є проблемою сьогодні (див. табл. 1). Однак, якщо менеджери використовують неточні моделі, то неправильно обґрунтовані рішення можуть фактично привести до краху проекту.

Таблиця 1. Результати дослідження точності оцінки

Дослідження	Jenkins	Phan	Heemstra	Lederer	Bergeron	Standish
Перевитрата коштів	34% (Median)	33% (Mean)			33% (Mean)	89% (Mean)
Проект завершено з перевитратою бюджету	61%		70%	63%		84%
Проект завершено з недовитратою бюджету	10%			14%		
Збільшення графіка розробки	22%					
Проект завершено після запланованого графіка	65%		80%			84%
Проект завершено до запланованого графіка	4%					

Точність програмних моделей вартості широко вивчена, водночас ефекти нестабільності результатів і систематичної похибки оцінки досліджуються досить рідко. Тому це дослідження звертається до можливостей отримання кращих програмних оцінок від параметричних моделей оцінки вартості, шляхом поєднання різних статистичних методів. У рамках дослідження проаналізовано точність і нестабільність результатів, систематичну похибку оцінки, і граничний діапазон значення трудовитрат при оцінці вартості. Результати показують, що ці моделі досягають високої точності за низької мінливості результатів в експериментах (навчальний набір і експериментальний набір даних). Такі результати досягаються шляхом скорочення кількості параметрів у параметричних моделях, тобто редукування їхньою кількістю в базовому рівнянні моделі.

Кращі моделі оцінки вартості ПЗ забезпечують не лише високу точність, а й низьку мінливість, і низьку тенденційність результатів. Точність моделей оцінки вартості ПЗ визначається наближенням отриманої оцінки до реальної вартості завершених проєктів, і часто вимірюється величиною PRED (N), яка розраховується за допомогою величини відносної похибки (MRE), що є абсолютною величиною відносного розміру різниці між фактичною і розрахованою вартістю проєкту [9–11]. PRED (N) показує середній відсоток оцінок, які знаходяться в межах N відсотків від фактичних значень. Наприклад, PRED (30) = 50 означає, що половина оцінки знаходиться в межах 30% від реальної вартості проєкту. Однак міра точності моделі не може відповісти на наступні питання:

- яка вірогідність того, що ця оцінка знаходиться в межах P відсотків;
- як "далеко" від N може знаходитись ця оцінка, якщо не знаходиться в межах P відсотків;
- наскільки впевнено ми можемо сказати, що оцінка має саме вказане значення PRED.

Значення мінливості оцінок моделей дають відповіді на поставлені вище питання і показують, в якому саме діапазоні знаходяться результати оцінки. Стандартне відхилення використовується для виявлення діапазону значень мінливості. Менше значення мінливості підвищує впевненість щодо результатів оцінки.

Для оцінки якості самої моделі оцінки вартості важливо перевірити її здатність якісно оцінювати рівень зусиль при розробці нових програмних проєктів. Калібрувальні дані зазвичай покривають обмежений набір доменів. Технології, процеси і люди (знання і досвід) швидко змінюються в галузі розробки ПЗ, водночас фактори, що впливають на продуктивність та моделі поведінки в галузі розробки ПЗ, та їхній вплив змінюються.

В опублікованих дослідженнях представлено багато підходів для підвищення точності моделей оцінки вартості ПЗ [10–12]. Тим не менше, значна неточність оцінок вартості ПЗ є давньою проблемою. Є джерела [13], в яких наведено приклади того, що найбільш значною проблемою моделей оцінки вартості ПЗ є статистичні недоліки, присутні у більшості моделей, а також відсутність широкого тестування цих моделей. Дослідження встановило, що багато моделей, які базуються на показниках оцінюваного розміру, ресурсів і характеристиках проєкту, більше використовують розумну інтуїцію експертів або користувачів, ніж статистичні методи [9, 10, 14]. Навіть при використанні статистичних методів шум і слабка дисперсія даних, а також низька кореляція змінних можуть сильно спотворювати вихідні дані.

Щоб об'єднати переваги і уникнути недоліків існуючих підходів розроблено підхід, показаний на рис. 1, для вирішення розглянутих проблем. Вхідні дані отримуються з бази даних COCOMO 81 [1], бази даних COCOMO II [5], та інших доступних баз даних, таких як, наприклад, база даних проєктів NASA COCOMO 81.

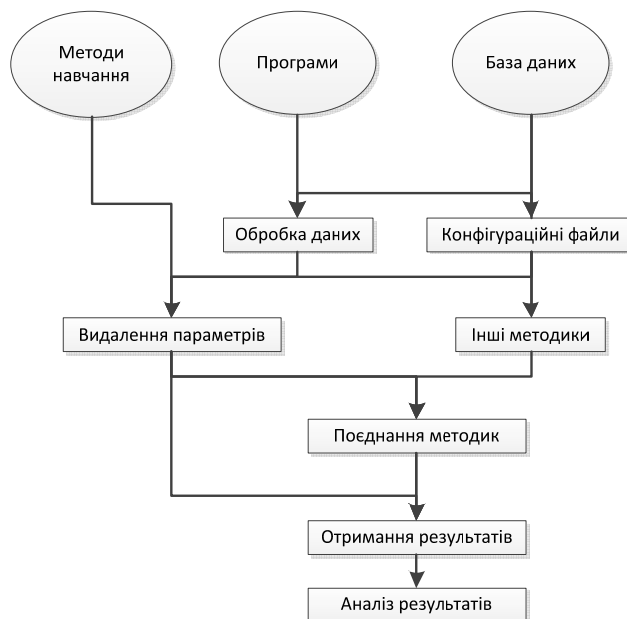


Рис. 1. Інтегрований підхід дослідження

Історія калібрування. В своїй роботі [15] Richard Stutzke показав, що історія технологій оцінки ПЗ почалася у 1960-х роках з концепції параметричної оцінки, розробленої Frank Freiman. Пізніше для оцінки апаратного забезпечення була розроблена модель PRICE. Оцінка ПЗ стала дуже активним напрямом протягом 1970-х років. Для оцінки вартості розробки ПЗ, дослідники спробували виявити основні фактори, що впливають на показник зусиль за допомогою регресивних методів, і включити результати цих досліджень у модель. Можливості моделі PRICE розширені для оцінки програмного забезпечення, і ця модель отримала назву PRICE S. Модель SLIM (модель життєвого циклу) [16, 17] отримана після проведення теоретичних і аналітичних досліджень. Наприкінці 1970-х років з'явилася модель COCOMO (конструктивна модель вартості) [1]. У 1980-х кращі моделі продовжували удосконалюватися і розвиватися. Протягом другої половини 1980-х років з'явилося кілька комп'ютеризованих засобів оцінки, які реалізували модель COCOMO [18]. Модель Randall Jensen [19] була розширена за рахунок усунення деяких небажаних аспектів моделі SLIM розробленої Lawrence Putnam, а потім продавалася як SEER-SEM (оцінка системи і ресурсів – модель оцінки ПЗ). У 1990-х роках, моделі вартості ПЗ продовжували покращуватись, водночас з'являлися різні нові методи і моделі [4, 20, 21].

Методи модельно-орієнтованої оцінки ПЗ включають алгоритми з математичними моделями, які калібрують за допомогою певної кількості даних відомих проектів; експертні підходи базуються на оцінках одного або декількох експертів з відповідним досвідом оцінювання; навчально-орієнтовані методи використовують методи штучного інтелекту і машинного навчання для аналізу попередніх проектів, що дозволяє краще оцінювати нові проекти; динамічні методи дозволяють явно адаптувати зміни програмного проекту протягом його життєвого циклу для підвищення точності оцінки; регресійні методи застосовують статистичний підхід до побудови моделей на основі великого обсягу проектних даних; змішані методи використовують більше однієї методики з різних категорій. Наприклад, байєсовський підхід, який поєднує експертний та регресійний підходи, був застосований для калібрування моделі COCOMO II і дозволив значно підвищити точність оцінки. Боем [22] надав ретельні огляди 38 підходів і різних моделей, розглянув їхні сильні та слабкі сторони і прийшов до наступних висновків: жодна методика не може бути пріоритетною для усіх ситуацій; поєднання різних методів у більшості випадків дає хороші результати; метод використання нейронних мереж та динамічні методи не є достатньо зрілими на момент дослідження. В даному дослідженні методи машинного навчання і регресійні методи використовуються для певних задач або їхні елементи вже інтегровані у моделі.

2. Моделі оцінки вартості ПЗ

Методи і моделі оцінки вартості ПЗ можна поділити на дві групи: неалгоритмічні методи та алгоритмічні моделі. До неалгоритмічних методів належать Price-to-win, оцінка за Паркінсоном, експертна оцінка, оцінка за аналогією [7, 8]. До алгоритмічних моделей належать SLIM і COCOMO [6-8].

2.1. Моделі оцінки вартості ПЗ. Модель оцінки вартості ПЗ базується на одній або декількох функціях, які описують залежність між характеристиками проекту і витратами на його реалізацію. Моделі поділяють за типом використовуваних функцій на лінійні, мультиплікативні, степеневі; і за використанням історичних даних на емпіричні та аналітичні. Найбільш часто використовуваними і добре документованими моделями є модель Lawrence Putnam (степенева, аналітична) і модель COCOMO (степенева, емпірична)

2.2. Модель COCOMO. Сімейство моделей COCOMO створено у 1981 році на основі бази даних про проекти консалтингової фірми TRW [4]. COCOMO представляє собою три моделі, орієнтовані на використання в трьох фазах життєвого циклу ПЗ: базова (Basic) застосовується на етапі вироблення специфікацій, вимог; розширена (Intermediate) – після визначення вимог до ПЗ; поглиблена (Advanced) – використовується після закінчення розробки проекту ПЗ. У загальному вигляді, рівняння моделей має вигляд $E = a \cdot S^b \times EAF$, де E – витрати на проект (у людино-місяцях), S – розмір коду (у KLOC), EAF – фактор уточнення витрат (effort adjustment factor). Параметри a і b залежать від виду додатку, що розроблюється, який може бути наступним:

– відносно простий проект, робота над яким ведеться однорідною командою розробників, вимоги носять рекомендаційний характер, відсутня заздалегідь вироблена вичерпна специфікація (наприклад, нескладне прикладне ПЗ);

– проект середньої складності, робота над яким ведеться змішаною командою розробників, вимоги до проекту визначаються специфікацією, однак можуть змінюватися в процесі розробки проекту (наприклад, ПЗ системи управління банківським терміналом);

– проект, який має бути реалізований у жорстких рамках заданих вимог чи специфікацій (наприклад, ПЗ системи управління польотами).

У базовій моделі фактор EAF приймається рівним одиниці. Для визначення значення цього фактора у розширеній моделі використовується таблиця, яка містить ряд параметрів, що визначають вартість проекту. При використанні поглибленої моделі, спочатку проводиться оцінка з використанням розширеної моделі на рівні компонента, після чого кожен параметр вартості оцінюється в чотирьох фазах ЖЦ ПЗ: визначення вимог, проектування, кодування і тестування компонентів, інтеграції та тестування готового продукту [4].

COCOMO II також є сімейством моделей і представляє собою розвиток базової (Basic) моделі COCOMO. COCOMO II складається з трьох моделей – створення додатків (Application Composition Model, ACM), раннього етапу розробки (Early Design Model, EDM) і пост-архітектурної (Post Architecture Model, PAM).

ACM використовується на ранньому етапі реалізації проекту для того, щоб оцінити основні елементи проекту: інтерфейс користувача, взаємодію з системою, продуктивність. За початковий розмір приймається

кількість екранів, звітів і 3GL-компонентів. Якщо припустити, що у проекті буде використано r відсотків об'єктів з раніше створених проектів, кількість нових об'єктних точок у проекті (Object Points, OP) можна розрахувати, як:

$$OP = (\text{object points}) \cdot (100 - r) / 100$$

Тоді витрати можна розрахувати за формулою

$$E = OP / PROD,$$

де PROD – табличне значення [4].

EDM – це високорівнева модель, якій потрібна порівняно невелика кількість вхідних параметрів. Вона призначена для оцінки доцільності використання тих чи інших апаратних і програмних засобів у процесі розробки проекту. Для визначення розміру використовується не скоригована функціональна точка (Unadjusted Function Point). Для її перетворення у LOC використовуються дані з таблиці [4]. Рівняння моделі раннього етапу розробки має вигляд: $E = a \cdot LOC \cdot EAF$ [4], де a – константа 2.45. EAF визначається так само, як і в оригінальній моделі COSOMO. Параметри для EDM отримуються комбінуванням параметрів пост-архітектурної моделі.

РАМ є найбільш деталізованою моделлю, яка використовується, коли проект повністю готовий до розробки. Для оцінки вартості ПЗ за допомогою РАМ необхідний пакет опису життєвого циклу проекту, який містить детальну інформацію про параметри вартості і дозволяє провести більш точну оцінку. РАМ використовується на етапі фактичної розробки та підтримки проекту. Для оцінки розміру можуть використовуватися як вихідні рядки коду, так і функціональні точки з модифікаторами, які враховують повторне використання коду. Модель використовує 17 параметрів вартості та 5 коефіцієнтів, що визначають масштаб проекту (у моделі COSOMO масштаб визначався параметрами виду програми). Рівняння РАМ має вигляд: $E = a \cdot LOC^b \cdot EAF$, де a прийнято за 2.55, а $b = 1.01 + 0.01 \cdot \sum W_i$, де W_i – параметри, що відображають властивості проекту, наприклад, схожість з раніше виконаними проектами, ризик вибору архітектури для реалізації, розуміння процесу розробки, досвідченість команди розробників. Значення параметрів є табличними [20].

2.3. Параметри вартості. Параметр вартості (cost driver) [23] – це суб'єктивна величина, яка оцінює різні часові, якісні та ресурсні аспекти розробки ПЗ. Кожен з параметрів може бути відкалібрований. Калібрування параметрів вартості – це коригування значень параметрів, яке впливає на значення трудовитрат, а отже на час і вартість, при оцінці програмного проекту. При калібруванні зазначених далі сімнадцяти параметрів вибирається оціночний рівень (дуже високий, високий, вище номінального, номінальний, нижчий від номінального, низький, дуже низький) параметра. У формулах цей рівень відображений у вигляді коефіцієнта трудовитрат i , таким чином, на кожній стадії розробки проекту впливає на вартість і тривалість тієї або іншої стадії. Виділяють наступні групи параметрів [5] (див. табл. 2): продукту (product factors), платформи (platform factors), персоналу (personnel factors) та проекту (project factors). У табл. 3 подано короткий опис кожного параметра.

Таблиця 2. Групи параметрів моделі COSOMO II

Параметри	Опис
Продукту	Враховують характеристики ПЗ, що розробляється (RELY, DATA, CPLX, RUSE, DOCU)
Платформи	Враховують характеристики програмно-апаратного комплексу, що потрібен для функціонування ПЗ (TIME, STOR, PVOL)
Персоналу	Враховують рівень знань та згуртованості колективу проекту (ACAP, PCAP, PCON, APEX, PLEX, LTEX)
Проекту	Враховуються вплив сучасних підходів та технологій, територіальної віддаленості членів колективу та строки виконання проекту (TOOL, SITE, SCED)

Модель COSOMO II постійно переглядається, оновлюється, і калібрується, щоб бути більш підлаштованою для оцінки нових проектів. Декілька підходів калібрування застосовувалися до моделі COSOMO II [9–12, 14, 24]. Початковий регресійний аналіз для COSOMO II [9] проведений за даними 83 проектів COSOMO 81. Подальший збір даних ідентифікував ще 78 нових проектів. Далі, проводилося опитування за методом Дельфі, коли експерти пропонували свої кращі ідеї щодо факторів, які впливають на витрати з розробки ПЗ. Після чого за допомогою методу калібрування Байєса всі дані опитування експертів за методом Дельфі та дані 83 і 78 проектів були поєднані. З тих часів метод Байєса у поєднанні з методом Дельфі використовуються для калібрування командою COSOMO II. Моделі оцінки зусиль, що витрачаються на розробку ПЗ, такі як COSOMO II, мають бути підлаштовані до особливостей конкретних процесів та колективів. Підлаштовані, тобто відкалібровані, моделі можуть бути більш точними після стратифікації конкретною організацією.

Таблиця 3. Опис параметрів моделі COCOMO II

Параметри	Опис
RELY (Required Software Reliability)	Враховує міру виконання програмою запрограмованих дій протягом певного часу
DATA (Database Size)	Враховує вплив обсягу тестових даних на розробку продукту. Рівень цього параметра розраховується як співвідношення байт в базі даних, що тестується, до кількості SLOC у програмі
CPLX (Product Complexity)	Включає п'ять типів операцій: управління, лічіння, пристрою-залежні, управління даними, управління користувацьким інтерфейсом. Рівень складності це суб'єктивне середньо-зважене значення рівнів типів операцій
RUSE (Developed for Reusability)	Враховує трудовитрати, необхідні додатково для написання компонентів, призначених для повторного використання в цьому або наступних проектах. Використовує такі оціночні рівні: "у проекті", "в програмі", "в лінійці продуктів", "в різних лінійках продуктів". Значення параметра накладає обмеження на наступні параметри: RELY і DOCU
DOCU (Documentation Match To Life-Cycle Needs)	Враховує ступінь відповідності документації проекту його життєвому циклу
TIME (Execution Time Constraint)	Враховує часові ресурси, які використовує ПЗ, при виконанні поставлених завдань
STOR (Main Storage Constraint)	Враховує відсоток використання сховищ даних
PVOL (Platform Volatility)	Враховує термін життя платформи (комплекс апаратного і програмного забезпечення, який потрібен для функціонування ПЗ, що розроблюється)
ACAP (Analyst Capability)	Враховує аналіз, здатність проектувати, ефективність та комунікативні здібності групи фахівців, які розроблюють вимоги і специфікації проекту. Параметр не повинен оцінювати рівень кваліфікації окремо взятого фахівця
PCAP (Programmer Capability)	Враховує рівень програмістів у колективі. При виборі значення для цього параметра слід особливо звернути увагу на комунікативні та професійні здібності програмістів і на командну роботу в цілому
PCON (Personnel Continuity)	Враховує ротацию кадрів в колективі
APEX (Applications Experience)	Враховує досвід колективу при роботі над ПЗ певного типу
PLEX (Platform Experience)	Враховує вміння використовувати особливості платформ, такі як графічний інтерфейс, бази даних, мережевий інтерфейс, розподілені системи
LTEX (Language and Tool Experience)	Враховує досвід програмістів (мови, середовища та інструменти)
TOOL (Use Of Software Tools)	Враховує рівень використання інструментів розробки
SITE (Multisite Development)	Враховує територіальну віддаленість (від офісу до міжнародних офісів) членів команди розробників і використовуваних ними засоби комунікації (від телефону до відео конференц-зв'язку)
SCED (Required Development Schedule)	Враховує вплив тимчасових обмежень, накладених на проект і на значення трудовитрат

3. Метод редукції основного рівняння

Суть запропонованого методу полягає в пошуку функціональної підмножини з меншою мінливістю результатів та вищою точністю, ніж для початкової функціональної множини моделі [9, 11]. В даному випадку до функціональної множини відносяться параметри моделі, що дозволяють її калібрувати.

Точність оцінки моделі визначається величиною PRED (N), яка вираховується зі значення відносної похибки (RE):

$$RE_i = \frac{\text{оцінка}_i - \text{дійсне}_i}{\text{дійсне}_i};$$

$$MRE_i = \text{abs}(RE_i);$$

$$PRED(N)_h = \frac{100}{T} \sum_{i=1}^T \begin{cases} 1, \text{ якщо } MRE_i \leq \frac{N}{100}. \\ 0, \text{ якщо навпаки} \end{cases}$$

Точність обраної функціональної підмножини визначається проведенням тестових експериментів, для кожної серії яких визначається середньоарифметичне значення $PRED(N)$:

$$PRED(N) = \frac{1}{n} \sum_{i=1}^n PRED(N)_h^i.$$

Для покращення моделі застосовуються методи машинного навчання на тестових даних з минулих проектів. Точність результатів підвищується зі збільшенням тестових даних, які були використані для навчання моделі.

Набори даних часто містять кілька сторонніх функцій, які можуть зменшити ефективність алгоритму навчання. Вибір функції підмножини (ВФП) допомагає визначити важливі атрибути та видалити надлишкові. Якби обиралися лише найбільш важливі фактори і передавалися алгоритму навчання це б дозволило отримувати компактніші теоретичні моделі. Це також підвищує розуміння набору даних або домену, що досліджуються. Зменшення розмірності також прискорює процес навчання. Повторюваність результатів поля ВФП проявляється в тому, що ігнорування певних факторів не повинно призводити до погіршення загальної точності результатів класифікатора.

Поєднання підходів та методів у представленому дослідженні показано на рис. 2. Для оцінки вартості ПЗ методи машинного навчання використовуються для розробки процесу і побудови моделі з тренувального набору даних, після чого статистичні методи використовуються для тестування, перевірки та оцінки процесу і моделі на тестовому наборі даних.

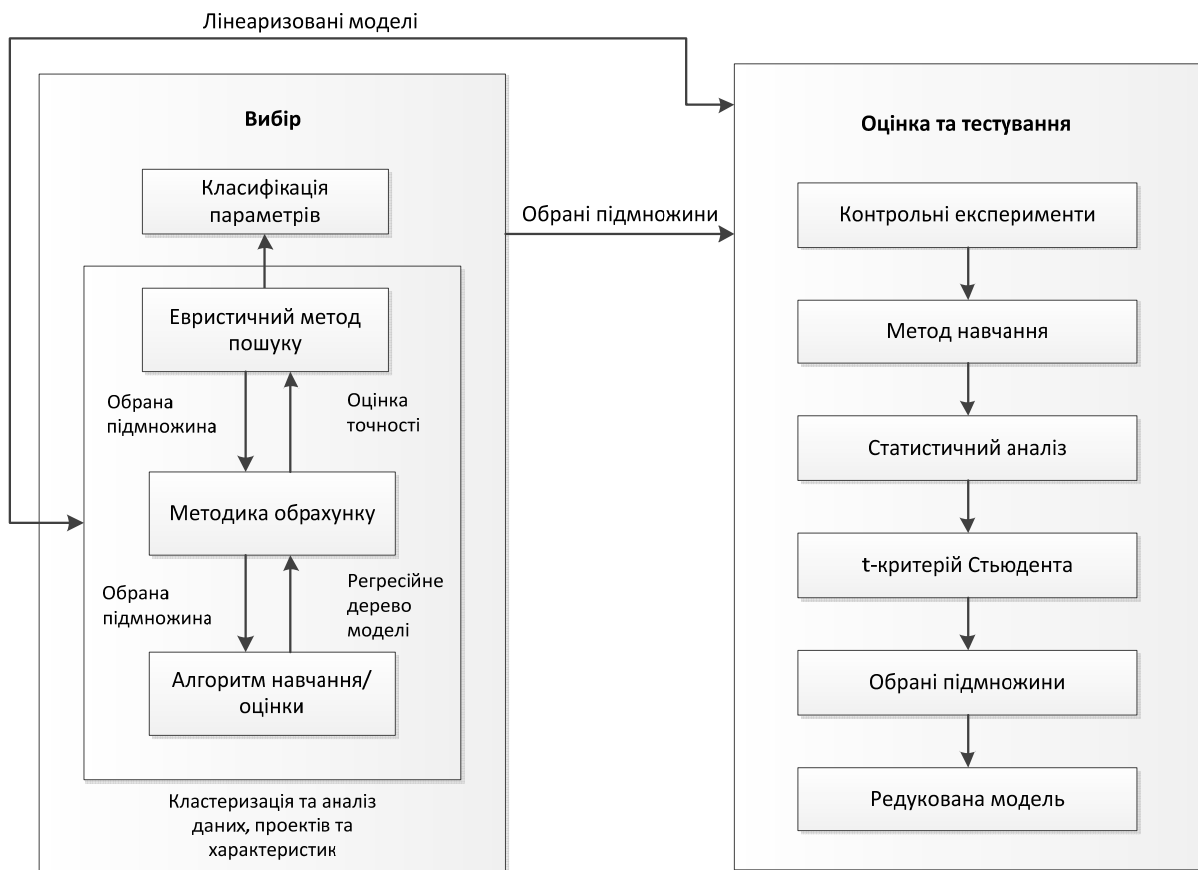


Рис. 2. Поєднання підходів та методів у дослідженні

Лінеаризація. Звичайний метод найменших квадратів та регресійна модель є лінійними методами. Якщо лінійна модель застосовується до нелінійних, результати оцінки моделі погіршуються. СОСОМО 81 і СОСОМО II – експоненціальні моделі, які припускають, що зміни обсягів робіт є швидшими, ніж зміни розміру проекту. Логарифмічне перетворення використовується для перетворення СОСОМО 81 і СОСОМО II в лінійні моделі. Лінеаризована модель СОСОМО 81:

$$\ln(PM) = \beta_0 + \beta_1 * \ln(Size) + \beta_2 * \ln(EM_2) + \dots + \beta_{20} * \ln(EM_{20})$$

Лінеаризована модель СОСОМО II:

$$\ln(PM) = \beta_0 + \beta_1 * \ln(Size) + \beta_2 * 0.01 * SF_1 * \ln(EM_2) + \dots$$

Усі ці перетворення створюють нові параметри, які є математично еквівалентними до оригінальних параметрів, але виражені в різних вимірах.

У запропонованому підході кращі функціональні підмножини на тестовому наборі даних виявлені за допомогою алгоритму ВФП (вибір функціональної підмножини). Оскільки набори даних можуть містити декілька зовнішніх атрибутів, які можуть знизити ефективність моделі, цей підхід допомагає нам ідентифікувати важливі атрибути і видалити надмірні. Якщо лише релевантні атрибути будуть вибраними і дослідженими за допомогою алгоритму вони зможуть допомогти створити компактніші моделі. Це покращує загальне розуміння набору даних або домену, що аналізується. Зниження розмірності також підвищує швидкість процесу дослідження.

Основні технології, використані в цьому дослідженні – ВФП (вибір підмножини особливостей) і ефективний евристичний пошук в підмножинах доступних атрибутів. Мета цього пошуку – знайти підмножину, яка дає подібну або підвищену продуктивність при використанні усіх атрибутів. Є 15 параметрів за винятком SIZE (загалом 16 параметрів в СОСОМО 81). Всебічний пошук в усіх можливих підмножинах досліджує 32768 можливих підмножин. У дослідженні припускається, що тільки 6 найбільш перспективних функціональних підмножин будуть відібрані методом і потрібні 20 секунд для кожного контрольного експерименту (тренувальний набір даних і тестовий набір відокремлені) для кожного PRED (25, 30) на 60 проектних зразках в цьому домені. Загальне число секунд вказує, що потрібно 1.25 року, щоб сформувавши спрощену модель. Ефективність методу ВФП дозволило скоротити час проходження тестів усього лише до 180 секунд.

Контрольні експерименти в цьому підході поєднують найкращі методи в оцінці зусиль калібрувальних експериментів. Контрольні експерименти мають наступний порядок: повторення, навчання, тести. Тобто, на етапі повторення доступні дані поділяються на тренувальний набір і тестовий набір. Методи навчання застосовуються до тренувального набору, щоб навчати модель або калібрувати параметри моделі. Після цього стан моделей та параметри запам'ятовуються і застосовується тестовий набір. Тренувальний та тестовий набори даних не мають спільних значень. Контрольні дослідження перевіряють якість калібрування параметрів, використовуючи дані, що не використовувалися при калібруванні. Якщо калібрування тестується на тренувальній множині, то такі дослідження не можна назвати контрольними. Але такі дослідження корисні для пошуку закономірностей в історичних даних. Проте, якщо мета полягає в створенні перспективної та адаптивної моделі, то теорія повинна бути перевірена на даних, які не використовувалися для її створення.

Тренування моделі відбувається за рахунок тестових даних з попередніх проектів. В даному дослідженні використовується вільнодоступна база даних проектів NASA. Дані в базі поділяються на дві частини (рис. 3): 1) 1/3 даних для тренування моделі; 2) 2/3 даних для перевірки відповідності моделі та точності її оцінок.

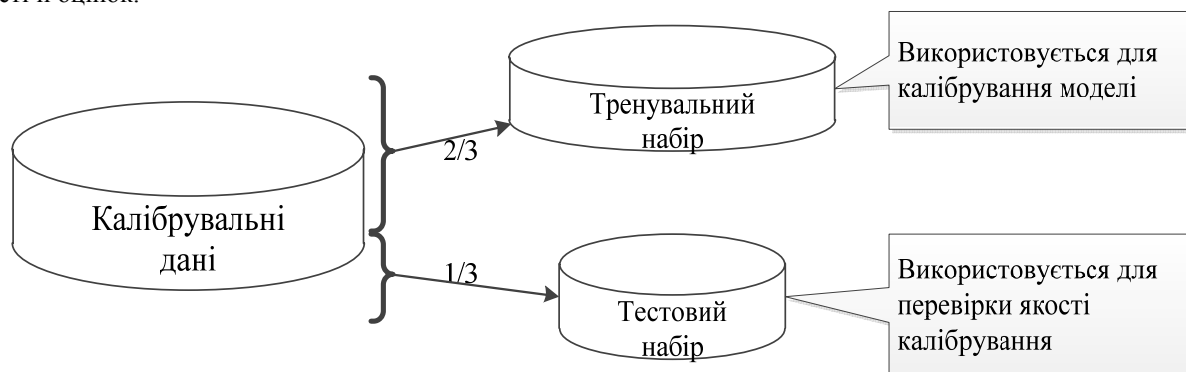


Рис. 3. Контрольні експерименти

У дослідженні використовується звичайний метод найменших квадратів, який є класичним статистичним лінійним регресійним методом моделювання з використанням найменших квадратів. Суть методу полягає у визначенні відношення залежної змінної (результат оцінки моделі) до незалежних змінних (параметри моделі).

Для перевірки середніх значень у вибірках використовується парний *t*-критерій Стьюдента для залежних вибірок. Парний *t*-критерій Стьюдента застосовується також для перевірки наявності суттєвої різниці між середніми значеннями точності функції підмножини в цьому підході. Це статистичний тест, який визначає відмінності між середніми значеннями з певним рівнем точності, припускаючи, що залежна змінна відповідає

закону нормального розподілу. Він використовується в цій моделі, щоб визначити кращу функцію підмножини. Можливі результати на повній множині та підмножинах наведені у табл. 4.

Таблиця 4. Визначення значення PRED(N) підмножин № 1 і № 2

№ експерименту	Підмножина № 1	Підмножина № 2	Відмінності
1	X1	Y1	D1=X1 - Y1
2	X2	Y2	D2=X1 - Y2
3	X3	Y3	D3=X1 - Y3
n	Xn	Yn	Dn=Xn - Yn

Як показали експерименти з даними NASA видалення хоча б одного параметра з оригінальної моделі дозволяє отримати кращу функціональну підмножину. Багато параметрів виявляють високу кореляцію та невідповідність їхніх значень моделі, що збільшує «шум» та невизначеність у вихідних даних моделі. Видалення саме таких параметрів з моделі дозволяє підвищити точність та зменшити мінливість вихідних результатів оцінки.

На рис. 4 можна побачити, що для функціональних підмножин, отриманих за допомогою запропонованого методу, середнє значення PRED(30) є значно вищим, що відповідає вищій точності результатів оцінки. Точки FS01-FS04 відповідають функціональним підмножинам зі зменшеною кількістю параметрів, точка All відповідає повній функціональній множині, а точка justLOC відповідає функціональній підмножині, до якої належить лише один параметр – кількість рядків коду. У всіх п'яти наборах даних, досліджуваних на рис. 4, відкидання щонайменше перших трьох наборів параметрів завжди призводить до істотного статистичного покращення. В деяких випадках, наприклад, ефективність "P02" зростала з 20% до майже 100%.

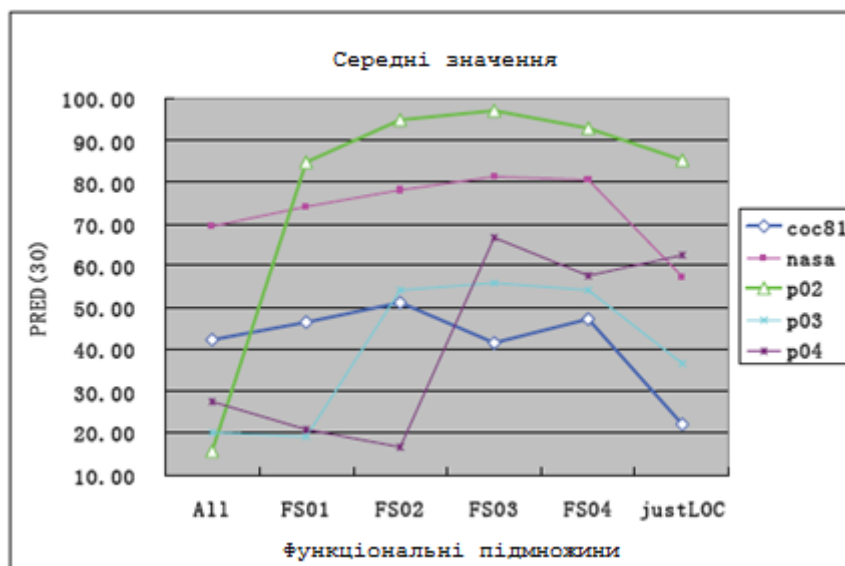


Рис. 4. Значення PRED (30) для різних функціональних підмножин

Висновки

У дослідженні теоретично узагальнено і розв'язано наукову задачу, яка має важливе значення для процесу розробки програмного забезпечення і полягає у розробці методу калібрування моделі оцінки вартості COCOMO з підвищеною точністю, сутність якого полягає у зменшенні кількості параметрів моделі. Основні наукові результати проведених досліджень є наступними:

- запропоновано використання методів математичної статистики для аналізу історичних проектних даних, що дозволяє визначити параметри моделі COCOMO, які підлягають усуненню для підвищення точності оцінки і підвищення повторюваності результатів;

- проведено огляд і дослідження літературних джерел, а також узагальнено ключові висновки дослідників з даної тематики;

- сформульовано гіпотезу, яка підтвердилася в результаті проведеного дослідження і полягає в тому, що для кожної загальної моделі існує модель зі зменшеною кількістю параметрів, яка за даних умов має вищу точність та повторюваність результатів.

1. Boehm B.W. Software engineering economics // Prentice-Hall. – 1981. – 320 p.
2. Shepperd M., Schofield C. Estimating software project effort using analogy // IEEE Trans Software Eng. – 1997. – P. 736–743.
3. David L. Norden-Raleigh Analysis: A Useful Tool for EVM in Development projects. – The Measurable News. – 2002. – 24 p.
4. Kim Johnson. Software cost estimation – Metrics and models. – University of Calgary. – 2001. – 115 p.
5. Boehm B.W. The COCOMO 2.0 Software Cost Estimation Model // American Programmer. – 2000. – 586 p.

6. Василенко Ю.Н. Алгоритмические методы оценки программного обеспечения // Матеріали конференції "Інженерія програмного забезпечення 2005". – НАУ, 2005. – С. 42–51.
7. Сидоров Н.А., Баценко Д.В., Василенко Ю.Н. и др. Методы и средства оценки стоимости программного обеспечения // Проблемы системного підходу в економіці. – НАУ. – 2004. – № 7. – С. 113–118.
8. Сидоров Н. А., Баценко Д. В., Василенко Ю. Н. и др. Модели, методы и средства оценки стоимости программного обеспечения // «УкрПРОГ-2006». – Киев. – 2006. – № 2–3. – С. 290–298.
9. Boehm B., Clark B., Devnani-Chulani S. Calibration Results of COCOMO II.1997, CSE tech report, 1997, <http://sunset.usc.edu/publications/TECHRPTS/1997/usccse97-507/usccse97-507.pdf>
10. Chulani S., Steece B. "COCOMO II Calibration", COCOMO Forum Oct 25 , 2000, <http://sunset.usc.edu/Activities/oct24-27-00/Presentations/Chulani.pdf>
11. Clark B. "Calibration of COCOMO II.2003", 17th International Forum on COCOMO and Software Cost Modeling, <http://sunset.usc.edu/events/2002/cocomo17/Calibration%20fo%20COCOMO%20II.2003%20Presentation%20-%20Clark.pdf>
12. Yang Y., Clark B. "Reducing Local Calibration Bias in COCOMO II 2004 Calibration", 19th International Forum on COCOMO and Software Cost Modeling, October 26-29, 2004, http://sunset.usc.edu/cse/pub/event/2004/COCOMO/files/WedAM/Wed_AM_05.ppt
13. The National Academy of Sciences. Chapter 4 - "Critique of Some Current Applications of Statistics in Software Engineering" in the book of "Statistical Software Engineering", International Standard Book Number 0-309-05344-7, 1996.
14. Yang Y., Clark B. "COCOMO II.2003 Calibration Status", CSE Annual Research Review, March 2003, http://sunset.usc.edu/events/2003/March_2003/COCOMO_II_2003_Recalibration.pdf
15. Stutzke R.D. "Software Estimating Technology: A Survey", CrossTalk. – 1996. – Vol. 9, N 5. – P. 17–22.
16. Putnam L. H. "A General Empirical Solution to the Macro Software Sizing and Estimating Problem" // IEEE Transactions on Software Engineering SE4, July 1978, p. 345–361.
17. Putnam L.H. "Software Cost Estimating and Life-Cycle Control: Getting the Software Numbers, New York", The Institute of Electrical and Electronics Engineers, Inc., 1980.
18. Сидоров Н.А. Утилизация программного обеспечения, экономический аспект // Кибернетика и системный анализ. – 1994. – № 3. – С. 151–167.
19. R. W. Jensen. "A Comparison of the Jensen and COCOMO Estimation Models", Proceedings of the International Society of Parametric Analysts, 1984. – P. 96–106.
20. McGibbon Th. Modern Empirical and Schedule Estimation Tools. – DACS Report. – 1997. – 72 p.
21. Capers J. Applied Software Measurement: Assuring Productivity and Quality. – McGraw-Hill. – 1996. – 590 p.
22. Boehm B. W., Abts C., Chulani S. "Software development cost estimation approaches - A survey", Annals of Software Engineering, Oct., 2000. – P. 177 – 205.
23. Баценко Д.В. Классификация параметров стоимости модели постархитектуры // Матеріали конференції "Інженерія програмного забезпечення 2005". – НАУ. – 2005. – С. 51–56.