

К. Е. ПЕТРОВ, О. В. ІВАНЕНКО, І. В. КОБЗЕВ

УДОСКОНАЛЕННЯ МЕТОДУ UCP ДЛЯ ОЦІНКИ ТРУДОВИТРАТ ПРИ РЕАЛІЗАЦІЇ ІТ-ПРОЄКТІВ

Проведено аналіз існуючих методів оцінки трудовитрат, необхідних для успішного виконання ІТ-проєкту. Запропоновано модифікований метод UCP, який дозволяє обчислювати оцінку обсягу програмного забезпечення проєкту на основі інтервальних оцінок факторів технічної складності та складності зовнішнього середовища. Це робить оцінку трудовитрат точнішою в порівнянні з оригінальними методом UCP, що, в свою чергу, підвищує ефективність процесів планування та управління проєктом. Наведено результати комп'ютерного моделювання, які демонструють працездатність та ефективність запропонованого методу.

1. Вступ

У сучасному світі, де інформаційні технології відіграють ключову роль в багатьох сферах життя, ефективне планування та управління ІТ-проєктами стає все важливішим. Одним з основних аспектів планування виконання ІТ-проєктів є оцінка трудомісткості розробки програмного забезпечення (ПЗ). Точна оцінка трудомісткості на ранньому етапі життєвого циклу проєкту може значно вплинути на успіх або провал проєкту, оскільки вона суттєво впливає на розподіл ресурсів, планування часу та бюджетних витрат.

Оцінка трудовитрат при виконанні ІТ-проєкту [1] є складним процесом, який залежить від багатьох чинників. Перш за все, це такі характеристики проєкту, як його обсяг і складність, які суттєво впливають на точність оцінки.

На теперішній час для оцінки трудомісткості розробки ПЗ використовується велика кількість різноманітних методів, які умовно можна розділити на: алгоритмічні; неалгоритмічні та методи, орієнтовані на навчання. Однак багато з цих методів мають обмеження щодо їх використання або не надають достатньої точності при оцінці трудомісткості, особливо на ранніх стадіях розробки ПЗ. Важливо розуміти, що від даних методів не слід очікувати абсолютної точності.

Одним з методів, який викликає особливий інтерес, є відомий метод Use Case Points (UCP), заснований на застосуванні варіантів використання (use cases). Він належить до алгоритмічних методів, що робить його достатньо надійним та передбачуваним у використанні. Цей метод також дозволяє врахувати функціональні вимоги до системи, що є особливо корисним на ранніх стадіях реалізації проєкту. Крім того, UCP не прив'язаний до конкретних технологій або мов програмування, що свідчить про його універсальність і гнучкість у використанні.

Актуальність даного дослідження обумовлена необхідністю вдосконалення існуючих методів оцінки трудомісткості розробки програмного забезпечення. Зокрема, методу UCP, який, незважаючи на свою надійність та універсальність, має деякі обмеження, які можуть впливати на точність оцінки. Тому виникає потреба в модифікації цього методу з метою покращення його точності та надійності.

2. Аналіз існуючих методів оцінки трудовитрат, необхідних для виконання ІТ-проєктів, та визначення проблеми дослідження

Для оцінки трудовитрат, необхідних для успішної реалізації ІТ-проєктів, використовується велика кількість моделей та методів, серед яких можна умовно виділити алгоритмічні (або параметричні), неалгоритмічні (непараметричні) та методи, орієнтовані на навчання.

Розглянемо ці методи детальніше.

Алгоритмічні моделі та методи використовують математичні моделі для оцінки трудовитрат на розробку проекту, які ґрунтуються на дослідженнях історичних даних і використовують деякі вхідні дані, наприклад, кількість рядків коду, кількість функцій для виконання, а також деякі фактори витрат, такі як мова програмування, методологія дизайну, рівень кваліфікації, оцінки ризиків тощо.

Модель життєвого циклу програмного забезпечення (Software Life Cycle Management, SLIM) [2] є однією з перших алгоритмічних моделей оцінки трудовитрат на розробку ПЗ. Вона заснована на використанні функції Нордена-Рейлі [3] і відома як модель макрооцінки. SLIM може записувати та аналізувати дані з раніше завершених проєктів, які потім використовуються для калібрування. Після цього можна відповісти на низку питань, щоб отримати значення нарощування трудовитрат у існуючій базі даних. Модель SLIM не підходить для малих проєктів, оскільки вона передбачає використання лінійного програмування та інші складні методи оцінки, які можуть бути зайвими для малих обсягів робіт.

Модель COCOMO (Constructive Cost Model) [4] є математичною моделлю для оцінки вартості, трудовитрат та термінів виконання проєктів з розробки ПЗ. COCOMO II є покращеною версією моделі COCOMO 81 та створена для вирішення проблем масштабування, функціональності та передбачення нових вимог та технологій, що виникли після випуску COCOMO 81.

COCOMO II включає в себе три підмоделі: композиційну, ранньої розробки проєкту та постархітектурну і враховує 17 параметрів витрат [4]. Модель може бути застосована для оцінки різних типів проєктів та різних етапів їх життєвого циклу, але для отримання надійних оцінок необхідно мати точні дані щодо розміру та складності проєкту, які іноді доволі важко забезпечити.

Метод функціональних точок (Function point, FP) [4], [5] є методом оцінки розміру і складності ПЗ на основі аналізу його функціональних можливостей, що надаються користувачам. Оцінювання ПЗ здійснюється шляхом представлення кожної його функції як множини, елементи якої належать до п'яти таких типів: входи, виходи, запити, внутрішні та зовнішні інтерфейси. Кожну з цих функцій класифікують за складністю та присвоюють їм певну кількість функціональних балів на основі їх складності. Основними недоліками методу FP є те, що атрибути якості, час розробки та людські ресурси не враховуються. Він не здатен працювати з гібридними системами.

Метод COSYSMO (Constructive Systems Engineering Cost Model) [6] використовує параметричну модель оцінки вартості програмного забезпечення, яка оцінює трудовитрати та час, що необхідні для виконання завдань системної інженерії. Модель містить 14 мультиплікаторів трудовитрат та 4 коефіцієнти розміру, загалом 18 параметрів. COSYSMO може бути адаптований до різних типів проєктів шляхом коригування параметрів, однак точність моделі залежить від наявності достатньо надійних проєктних даних. Крім того, модель вимагає розуміння та оцінки численних факторів, що може бути складним завданням.

SEER-SEM (Software Evaluation and Estimation of Resources – Software Estimating Model) [4], [6] є моделлю оцінювання програмного проєкту, яка використовує підхід на основі параметрів і базується на використанні моделі Дженсена. SEER-SEM використовує параметри розміру, персоналу, складності, середовища та обмежень для оцінки вартості, ризиків та графіку проєкту. Вона охоплює весь життєвий цикл проєкту – від фази його планування до фази його реалізації, і працює з декількома конфігураціями середовища і додатків. До основних недоліків SEER-SEM можна віднести високу складність завдяки врахуванню великої кількості вхідних параметрів, пов'язаних з різними факторами проєкту, та специфічних деталей проєкту, особливо при ідентифікації нелінійних зв'язків між параметрами входу і виходу.

Метод Use Case Points (UCP) [7] призначений для специфічних впорядкованих систем і вимог до систем, що записані за допомогою варіантів використання (use cases), які є частиною систем Unified Modeling Language (UML). На основі компонентів варіантів використання системи визначається значення UCP, яке використовується для кількісного визначення обсягу ПЗ і яке потім використовується для оцінки трудовитрат проекту. Метод може бути використаний на ранній стадії життєвого циклу проекту і отримані оцінки будуть близькими до реальних, але лише для тих програмних проектів, специфікація яких може бути виражена через варіанти використання.

Неалгоритмічні методи використовують деяку інформацію про вже виконані проекти, схожі на поточний, і зазвичай процес оцінки вартості в цих методах здійснюється відповідно до аналізу попередніх наборів даних. У більшості випадків неалгоритмічні методи оцінки базуються на консенсусі, тобто на спроможності одного або групи експертів працювати разом для оцінки трудовитрат, необхідних для реалізації проекту.

Метод експертного оцінювання [8], [9] є одним з традиційних методів, які використовуються для оцінки трудовитрат на ранніх етапах розробки проекту. Він передбачає консультації з експертом або групою експертів, які мають значний досвід роботи з аналогічними проектами. Ці спеціалісти можуть надати цінну інформацію про потреби проекту, включаючи конкретні задачі та ресурси, необхідні для його успішного виконання. Такий підхід дозволяє враховувати практичний досвід і експертні знання для точнішої оцінки трудовитрат.

Розглянемо основні підходи, що використовуються для отримання експертних оцінок.

Процес оцінки трудовитрат згідно з методом Delphi починається з того, що керівник проекту надає експертам специфікацію та форму оцінки, які вони заповнюють анонімно. Потім проводиться групова зустріч для обговорення питань оцінки. Керівник проекту готує резюме оцінки трудовитрат та розсилає його експертам для повторного заповнення форм. Ці кроки повторюються до досягнення консенсусу щодо оцінки трудовитрат на виконання проекту [6].

Метод Planning Poker [10] формує оцінку трудовитрат шляхом об'єднання думок декількох експертів. На початку сесії планування команда представляє користувачі історії. Кожен учасник отримує колоду карток з оцінками та обирає ту, що відповідає його оцінці. Після цього всі одночасно відкривають свої картки, і якщо оцінки відрізняються, вони обговорюються. Цей процес повторюється до досягнення консенсусу.

Перевагою методів експертного оцінювання є те, що не потрібно витратити час на складний аналіз або збір історичних даних, оскільки експерти можуть швидко надати кваліфіковані та обґрунтовані оцінки.

Однак отримані оцінки є суб'єктивними та обмеженими областю компетенції експертів, що може призвести до неоднозначних результатів та відсутності консенсусу.

Метод Top to Bottom [4], [5] або «згори вниз» передбачає, що спочатку визначається загальна вартість проекту на основі глобальних характеристик програмного продукту. Після цього проект розбивається на окремі компоненти і детально оцінюється кожна частина проекту. Отримана оцінка загальної вартості проекту пропорційно розподіляється між різними компонентами. Цей метод є особливо корисним, коли доступна лише обмежена інформація про проект і потрібно отримати приблизну оцінку. Основним недоліком цього методу є його відносна неточність.

Метод Bottom to Top [4] або «знизу вгору» передбачає оцінку трудовитрат всіх компонентів ПЗ для отримання загального кошторису проекту. Цей підхід направлений на формування оцінки системи на основі накопичених знань про окремі компоненти ПЗ та їх взаємодію. Для застосування цього методу необхідна наявність початкового проектного

дизайну, який чітко визначає структурне розбиття компонентів системи. Такий підхід дозволяє розглядати трудовитрати детальніше, що зазвичай робить оцінки точнішими, порівняно з іншими методами [10]. Однак процес оцінювання може зайняти багато часу, що здорожує отримання оцінки.

В методі Price-to-win або Cost-to-win [5] витрати на програмування оцінюються як найкраща запропонована вартість для перемоги у конкурсі на виконання проєкту. Недоліком такого підходу є те що, замість того, щоб зосередитися на функціональності ПЗ, він більше орієнтований на бюджет та можливості замовника. Точність оцінки значно варіюється залежно від бюджету клієнта, тому цей метод дає малу точність [5].

Метод оцінки за аналогіями [7] полягає в порівнянні проєкту з раніше завершеними схожими проєктами. Фактичні дані з завершених проєктів екстраполюються для оцінки трудовитрат запропонованого проєкту. Метод аналогії може бути використаний як на рівні системи, так і на рівні компонентів.

Цей метод є достатньо простим і швидким у використанні, але може бути ненадійним через обмежену кількість схожих проєктів та виникнення складностей у оцінці відмінностей між проєктами.

Методи, орієнтовані на навчання, використовують різноманітні моделі машинного навчання для оцінки трудовитрат. Ці методи стають все популярнішими через їхню здатність передбачати точні результати на основі аналізу даних, отриманих в ході виконання попередніх проєктів.

Наприклад, в [4] описується успішне застосування рекурентних та радіально-базисних штучних нейронних мереж (ШНМ) для оцінки вартості ПЗ.

Однак, варто враховувати, що ШНМ схильні до перенавчання, а також вимагають використання значних обчислювальних ресурсів та великих обсягів даних для свого навчання.

В [7] також розглядаються еволюційні методи оцінки трудовитрат, необхідні для розробки ПЗ, які базуються на використанні генетичних алгоритмів.

В [4], [7] представлені методи, в яких використовуються системи нечіткого логічного виведення для оцінки трудовитрат, необхідних для реалізації проєктів.

Нечітка логіка дозволяє лінгвістично представити вхідні та вихідні дані моделі, що є толерантними до неточності. Вона особливо підходить для оцінки трудовитрат, оскільки багато атрибутів ПЗ вимірюються за номінальною або порядковою шкалою, що є окремим випадком лінгвістичних змінних [7].

Методи оцінювання трудовитрат, засновані на використанні Bayesian networks (Байєсівських мереж) [6], базуються на застосуванні ймовірнісних орієнтованих ациклічних графових моделей для представлення залежностей між змінними.

Оцінка трудовитрат на ранніх стадіях розробки ПЗ відіграє критично важливу роль для успішного виконання проєкту. Втім, через недостатність наявної інформації, прогнозні значення оцінки трудовитрат мають високий рівень невизначеності.

В ході аналізу існуючих методів оцінки трудовитрат при реалізації ІТ-проєктів було виявлено їх основні недоліки:

- алгоритмічні методи мають велику залежність від початкових даних, які важко отримати на початковому етапі, або вимагають великої кількості кроків для обчислення оцінки трудовитрат;
- неалгоритмічні методи обмежені рівнем кваліфікації експертів та суб'єктивністю їхніх оцінок, що суттєво впливає на остаточну оцінку трудовитрат, необхідних для успішного завершення проєкту;
- методи, орієнтовані на навчання, є складними у проєктуванні та мають обмеження

щодо використання даних недостатньої якості та обсягу, а також можливості надмірної адаптації до навчального набору даних, що може призвести до перенавчання та недооцінки нових даних.

Тому актуальним завданням є розробка нових або вдосконалення існуючих методів оцінки трудовитрат для успішного виконання проєктів з точки зору підвищення їх точності та універсальності використання.

3. Мета і задачі дослідження

Метою дослідження є вдосконалення методу Use Case Points (UCP) оцінки трудовитрат, необхідних для успішного виконання ІТ-проєкту. Модифікація класичного методу UCP дозволить отримувати точніші прогностичні оцінки трудовитрат при реалізації ІТ-проєктів, що суттєво підвищить ефективність процесів планування та управління проєктами.

Для досягнення мети необхідно вирішити такі основні завдання:

– вдосконалити метод UCP оцінки трудовитрат, необхідних при реалізації ІТ-проєкту, що дозволить враховувати інтервальні оцінки факторів технічної складності та зовнішніх чинників, які впливають на оцінку обсягу ПЗ і, відповідно, на оцінку трудомісткості проєкту;

– провести експериментальну перевірку працездатності запропонованого методу та оцінити ефективність його використання.

4. Розробка модифікованого методу оцінки трудовитрат при реалізації ІТ-проєкту

4.1. Опис оригінального методу Use Case Points

Метод Use Case Points (UCP) [11]-[13] оцінки трудовитрат на розробку ПЗ базується на аналізі функціональних вимог системи через використання варіантів або прецедентів використання (use cases). За допомогою цього методу можна оцінити обсяг роботи, необхідної для розроблення програмного продукту, і використовувати цю інформацію при управлінні проєктами та плануванні ресурсів.

В основі методу лежить поняття «use case» – концепція, яку використовують при розробленні ПЗ, дизайні продукту тощо для опису можливих сценаріїв використання системи для досягнення конкретних цілей або завдань. У ньому описується взаємодія між користувачами або учасниками (акторами) та системою для досягнення конкретного результату [11].

Основне призначення варіанту використання – допомагати донести стратегію до зацікавлених сторін і усунути розрив між бізнес-обґрунтуванням і технічними вимогами.

Основними складовими прецедентів використання є актори, система, мета, базові та альтернативні потоки.

При використанні методу UCP можливі два основних підходи до представлення варіантів використання: візуальне представлення за допомогою відповідних діаграм або їх текстовий опис. Обидва підходи мають свої переваги, і вибір між ними залежить від конкретних потреб та обставин.

На основі компонентів варіантів використання системи обчислюється значення бальної оцінки UCP, яка використовується для кількісного визначення обсягу ПЗ, а потім це значення використовується для оцінки трудовитрат проєкту.

Загальна формула розрахунку значення оцінки UCP має вигляд:

$$UCP = UUCP \times TCF \times ECF, \quad (1)$$

де *UUCP* (Unadjusted Use Case Points) – нескориговані точки варіантів використання; *TCF* (Technical Complexity Factor) – фактор технічної складності; *ECF* (Environmental Complexity Factor) – фактор складності зовнішнього середовища.

В свою чергу, $UUCP$ розраховуються за формулою:

$$UUCP = UAW + UUCW, \quad (2)$$

де UAW (Unadjusted Actor Weight) – нескоригована вага актору; $UUCW$ (Unadjusted Use Case Weights) – нескориговані ваги варіантів використання.

Таким чином, можна виділити такі етапи розрахунку значення бальної оцінки обсягу проекту з використанням методу UCP.

Етап 1. Обчислення нескоригованої ваги актора (UAW). На цьому етапі визначають усіх акторів системи. Актори бувають трьох типів: прості, середні і складні. Після визначення для кожного актора встановлюється вага, у відповідності до його типу. Докладний опис типів акторів та значень відповідних вагових коефіцієнтів наведено у [11], [12].

Обчислення UAW відбувається за формулою:

$$UAW = \sum_{i=1}^3 a_i w_i, \quad (3)$$

де a_i – кількість акторів i -го типу; w_i – вага для кожного типу.

Етап 2. Обчислення нескоригованих ваг варіантів використання ($UUCW$). Кожен варіант використання класифікується як простий, середній або складний. Основою для цього рішення є кількість транзакцій у варіанті використання, включаючи альтернативні шляхи. Для цього транзакція визначається як атомарний набір дій, який або виконується повністю, або не виконується взагалі.

$UUCW$ розраховується за формулою:

$$UUCW = \sum_{i=1}^3 uc_i w_i, \quad (4)$$

де uc_i – кількість варіантів використання в i -му типі варіантів використання; w_i – вага для кожного типу.

Значимо, що відповідні значення вагових коефіцієнтів для варіантів використання на основі транзакцій детально розглянуті у [11], [12].

Етап 3. Обчислення нескоригованих точок варіантів використання ($UUCP$). Нескориговані ваги варіантів використання ($UUCW$) і нескоригована вага актора (UAW) разом дають нескоригований розмір (обсяг) системи, який називають нескоригованими точками варіанта використання. $UUCP$ розраховується за формулою (2).

Етап 4. Обчислення фактору технічної складності (TCF). Оцінка технічних чинників використовується для визначення складності архітектури застосунку. Кожен з тринадцяти факторів оцінюють за шкалою від 0 до 5. Оцінка «0» означає, що фактор не має значення для цього проекту, оцінка «5» означає, що він є важливим. Перелік технічних факторів та відповідних вагових коефіцієнтів детально описано у [11], [12].

Показник TCF в методі UCP обчислюється за формулою:

$$TCF = 0,6 + 0,01 \times \sum_{i=1}^{13} t_i w_i, \quad (5)$$

де t_i – точкове значення (бальна оцінка) i -го технічного фактору; w_i – вага i -го фактору.

Етап 5. Оцінка фактору зовнішніх чинників (ECF). Цей вид оцінки використовується для визначення коефіцієнту впливу організаційних ризиків на розробку. Кожен з восьми

факторів оцінюють за шкалою від 0 до 5. Оцінка «0» означає, що фактор не має значення для цього проекту, оцінка «5» означає, що він є важливим. Обчислення проводяться аналогічно розрахунку оцінки фактору технічної складності. Перелік відповідних технічних факторів та їх відповідних вагових коефіцієнтів детально описані у [11], [12].

Показник ECF в методі UCP розраховується таким чином:

$$ECF = 1,4 - 0,03 \times \sum_{i=1}^8 e_i w_i, \quad (6)$$

де e_i – точкове значення(бальна оцінка) фактору зовнішнього чинника i ; w_i – вага фактору i .

Етап 6. Остаточний підрахунок значення UCP. На останньому етапі обчислень оцінюють загальну кількість варіантів за формулою (1) [13].

4.2. Модифікація оригінального методу Use Case Points

Модифікація оригінального методу UCP полягає у зміні підходу для розрахунків оцінок фактору технічної складності (TCF) та фактору зовнішніх чинників (ECF), тобто етапів 4 та 5 оригінального методу UCP.

Наведемо деякі міркування з цього приводу.

Як було відзначено вище, метод UCP відрізняється своєю алгоритмічною природою, що забезпечує стабільність та прогнозованість при оцінці трудовитрат. Він також враховує функціональні вимоги до системи, що робить його незамінним інструментом на початкових стадіях проекту.

Слід зазначити, що метод UCP ідеально підходить для проектів, що використовують об'єктно-орієнтоване програмування, завдяки його здатності до адаптації до різних мов програмування. Це свідчить про його універсальність і гнучкість у використанні.

Проте, не дивлячись на ці переваги, метод UCP має і певні обмеження, зокрема, не завжди дає високу точність при оцінці трудовитрат. Аналіз літературних джерел показав високу зацікавленість дослідників у розробці нових підходів, заснованих на використанні варіантів використання, які б давали точніші оцінки трудовитрат на виконання проектів. З цієї причини було запропоновано багато методів, які є модифікаціями UCP. Більшість з них зосереджені лише на зміні значення нескоригованої точки використання ($UUCP$), яке є вагою, що присвоюється акторам і варіантам використання.

Однак, крім $UUCP$, як видно з формули (1), на точність оцінки впливають ще два інші коригувальні фактори – значення оцінок TCF та ECF . При визначенні значень цих коригувальних коефіцієнтів виникають певні труднощі, оскільки часто відсутній базовий рівень для порівняння. Керівники проектів постійно стикаються з необхідністю інтерпретувати кожен фактор та згадувати інші проекти для порівняння з поточним. Так деякі з коригувальних коефіцієнтів можуть бути неоднозначними. Наприклад, фактор «Паралельність» може включати в себе паралельну обробку, паралельне програмування або взаємодію системи з іншими додатками. Відсутність чітких вказівок в методі UCP щодо того, що саме цей фактор повинен вимірювати, може призвести до неточностей при його оцінці [13], [14].

Таким чином, можна зробити висновок, що TCF та ECF можуть значно впливати на точність оцінки і, отже, потребують додаткового аналізу. Незначна зміна їх значень може суттєво вплинути на оцінку обсягу ПЗ і, відповідно, на оцінку трудомісткості проекту.

Слід зазначити, що бальна оцінка факторів TCF та ECF здійснюється керівником проекту та відображає його суб'єктивну думку. У класичній реалізації методу UCP вона задається у вигляді точкового значення (константи від 0 до 5 балів).

Гнучкішим рішенням в цьому випадку є представлення цієї оцінки в інтервальної формі для ситуацій, коли оцінювання проводиться групою експертів або коли менеджер

проекту не достатньо впевнений у своїй оцінці.

Для отримання значення очікуваної оцінки важливості фактору при її завданні в інтервальній формі пропонується використати апроксимацію інтервальних значень за допомогою бета-розподілу, що дозволить дещо знизити суб'єктивізм оцінок, тобто їх невизначеність, завдяки врахуванню різних можливих сценаріїв оцінки впливу конкретного фактору на проект – оптимістичного, песимістичного та найімовірнішого. Це дозволить отримати точнішу і збалансованішу оцінку. Завдяки такому підходу, менеджери проекту можуть визначити діапазон (інтервал) можливих оцінок для кожного фактора, що дозволяє обчислити очікувану оцінку, замість простого призначення оцінки (деякого детермінованого значення).

Аналогічний підхід успішно використовується в методі PERT (Project Evaluation and Review Technique) [15], [16] для врахування можливих ризиків при плануванні часу виконання окремих операцій, з яких складається проект.

Показник TCF в модифікованому методі UCP пропонується обчислювати за формулою:

$$TCF = 0,6 + 0,01 \times \sum_{i=1}^{13} t_i w_i, \quad (7)$$

де t_i – очікуване значення i -го технічного фактору; w_i – вага i -го фактору.

Очікуване значення технічного фактору t_i розраховується таким чином:

$$t_i = \frac{O_i + 4M_i + P_i}{6}, \quad (8)$$

де O_i – оптимістична оцінка i -го фактору; M_i – найімовірніша оцінка i -го фактору; P_i – песимістична оцінка i -го фактору.

Результати розрахунку очікуваних значень оцінки факторів технічної складності TCF з використанням формули (8) наведено у табл. 1.

Таблиця 1

Результати розрахунку очікуваних значень оцінки TCF

Фактор	Назва	Ваговий коефіцієнт w_i	Оцінка фактору			
			O_i	M_i	P_i	t_i
T2	Висока продуктивність	2	3	4	5	4
T3	Ефективність роботи кінцевого користувача	1	2	3	5	3,17
T6	Легкість інсталяції	0,5	1	3	4	2,83

Показник ECF у модифікованому методі UCP обчислюється за формулою (6), в якій e_i – це очікуване значення i -го фактору зовнішнього чинника, що розраховується за формулою (8) аналогічно тому, як це зроблено для TCF .

Однак оцінка трудовитрат на цьому не закінчується. Правильний підхід до планування проекту передбачає оцінку його обсягу та визначення тривалості на основі значення UCP (1). Точки варіантів використання застосовуються для оцінки обсягу проекту, але їх недостатньо для безпосередньої відповіді на питання про трудовитрати проекту. Виходячи з оцінки обсягу, необхідно визначити тривалість, враховуючи швидкість роботи команди над варіантами використання.

В різних дослідженнях пропонується різні співвідношення кількості годин, що необхідно витратити на один варіант використання. Зазвичай це значення варіюється від

15 до 30 людино-годин на один варіант використання. Наприклад, для проекту з 500 варіантами використання необхідно від 7500 до 15000 людино-годин роботи.

Однак, замість використання загальних рекомендацій, доцільнішим є розрахунок власних історичних середніх значень для організації на основі даних попередніх проектів. Створення сховища проектних даних для збереження таких показників дозволить точніше оцінювати майбутні проекти.

Використовуючи діапазон годин та кількість варіантів використання, можна визначити очікувану оцінку трудовитрат проекту [13].

Таким чином, остаточно оцінка трудовитрат проекту оцінюється за формулою:

$$Effort = UCP \times ER, \quad (9)$$

де *Effort* – значення трудовитрат в людино-годинах; *ER* – коефіцієнт трудомісткості, який дорівнює кількості людино-годин, необхідної для виконання одного *UCP*.

5. Експериментальна перевірка результатів вдосконалення методу UCP

Для експериментальної перевірки працездатності та ефективності запропонованого модифікованого методу UCP було використано набір даних [17], який знаходиться у відкритому доступі і підтримується Університетом Томаша Баті (Чеська Республіка). Ці дані були отримані від з трьох компаній-розробників ПЗ та є реальним набором інформації. Для аналізу кількості кроків або акторів було використано метод UCP.

Датасет складається з 18 атрибутів:

- project_No – ідентифікатор проекту для цілей ідентифікації;
- simple actors – кількість акторів, класифікованих за UCP як прості;
- average actors – кількість акторів, класифікованих за UCP як середні;
- complex actors – кількість акторів, класифікованих за UCP як складні;
- uaw – нескоригована вага актора, обчислюється за формулою (3);
- simple UC – кількість варіантів використання, класифікованих як прості;
- average UC – кількість варіантів використання, класифікованих як середні;
- complex UC – кількість варіантів використання, класифікованих як складні;
- ucsw – нескоригована вага варіантів використання, обчислюється за формулою (4);
- tcf – фактор технічної складності;
- ecf – фактори складності зовнішнього середовища;
- real_p20 – трудовитрати) в людино-годинах, визначаються за допомогою коефіцієнта продуктивності (PF = 20);
- real_effort_person_hours – трудовитрати в людино-годинах;
- sector – проблемна область проекту;
- language – мова програмування, що використовується для проекту;
- methodology – методологія розробки, що використовується для розробки проекту;
- applicationtype – класифікація типу проекту, надається донором;
- datadonor – анонімізована аббревіатура для донора даних.

Всього в датасеті зібрана інформація про 71 проект.

Для тестування ефективності запропонованого модифікованого методу як ключові були відібрані такі атрибути: project_No, simple actors, average actors, complex actors, uaw, simple UC, average UC, complex UC, tcf, ecf, real_p20 та real_effort_person_hours.

Оскільки в запропонованій модифікації методу UCP передбачається використання інтервальних оцінок для розрахунку *TCF* та *ECF* за формулою (8), необхідно мати інформацію про кількість балів, яку присвоєно кожному конкретному фактору. Однак, зазначений масив даних не містить таких деталей, тому було вирішено розрахувати ці

значення вручну. Хоча ці розрахунки можуть не бути абсолютно точними, вони максимально наближені до реальних показників.

Для тестування було випадковим чином відібрано 21 проєкт з 71 і для кожного з них розраховано значення оцінки UCP за формулою (1).

Після цього проводилося оцінювання точності запропонованого методу та порівняння його із оригінальним UCP. Для цього використовувались такі метрики:

- *MMRE* (Mean Magnitude of Relative Error) – середня величина відносної похибки;
- *PRED(x)* (Percentage of Prediction within x %) – відсоток передбачення в межах x %;
- *MAE* (Mean Absolute Error) – середня абсолютна похибка.

Обчислення значень метрик *MMRE* та *PRED(x)* базується на розрахунку відносної похибки (*MRE*) за формулою:

$$MRE_i = |y_i - \tilde{y}_i| / y_i, \quad (10)$$

де y_i – реальне (фактичне) значення; \tilde{y}_i – прогнозоване значення.

Розрахувати *MMRE* можна за формулою:

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE_i, \quad (11)$$

де n – кількість спостережень.

Обчислення метрики *PRED(x)* відбувається за формулою:

$$PRED(x) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1, & \text{якщо } MRE_i \leq x \\ 0, & \text{в іншому випадку} \end{cases}, \quad (12)$$

де x – порогове значення, яке визначає допустимий рівень відносної похибки.

Ідеальною вважається модель, яка має низьке значення *MMRE* та високе значення *PRED(x)*. Ці величини допомагають визначити точність оцінки та стійкість моделі. Точна модель прогнозування трудовитрат повинна мати $MMRE \leq 0.25$ (щоб гарантувати, що середня помилка оцінки не перевищує 25 %) і $PRED(25) \geq 0.75$ (щонайменше 75 % прогнозованих значень повинні мати відносну похибку (*MRE*) менше 25 %) [13].

Середня абсолютна похибка (*MAE*) показує ступінь невідповідності між фактичними та прогнозованими значеннями та розраховується за формулою:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i|, \quad (13)$$

Чим ближче значення *MAE* до нуля, тим точніша модель.

В табл. 2 наводиться значення відповідних метрик (11)-(13) для порівняння точності оцінок трудовитрат на виконання ІТ-проєктів оригінального та модифікованого методів UCP.

Таблиця 2

Результати порівняння точності методів оцінки трудовитрат

Метод	<i>MMRE</i>	<i>PRED(25)</i>	<i>MAE</i>
Оригінальний UCP	0,22	63,64%	77,21
Модифікований UCP	0,17	86,36%	60,54

6. Обговорення результатів дослідження

Аналіз отриманих результатів показав, що модифікований метод UCP є точнішим, ніж оригінальний метод UCP. Це підтверджується значеннями метрик *MMRE*, *PRED(25)* і *MAE*

(табл. 2). Зокрема, збільшення точності на 22 % за показником *PRED (25)* може допомогти компаніям ефективніше здійснювати управління проектами.

Для наочної демонстрації результатів дослідження було побудовано діаграму значень середньої абсолютної похибки (*MAE*) для двох методів оцінки трудовитрат: оригінального методу *UCP* та модифікованого методу *UCP* (позначається як *UCP_2*). Діаграму наведено на рис. 1. На горизонтальній осі діаграми відображені номери відповідних проектів.

Побудована діаграма ясно демонструє, що модифікований метод *UCP* (стовпчики справа) забезпечує точніші передбачення оцінок трудовитрат порівняно з оригінальним методом *UCP* (стовпчики зліва). У більшості випадків модифікований метод *UCP* має меншу середню абсолютну похибку (*MAE*), що підтверджує його високу ефективність і надійність у прогнозуванні трудовитрат.

На основі отриманих результатів можна зробити висновок, що модифікований метод *UCP* дає точніші значення оцінок прогнозованих трудовитрат, що позитивно впливає на успішність виконання проектів.

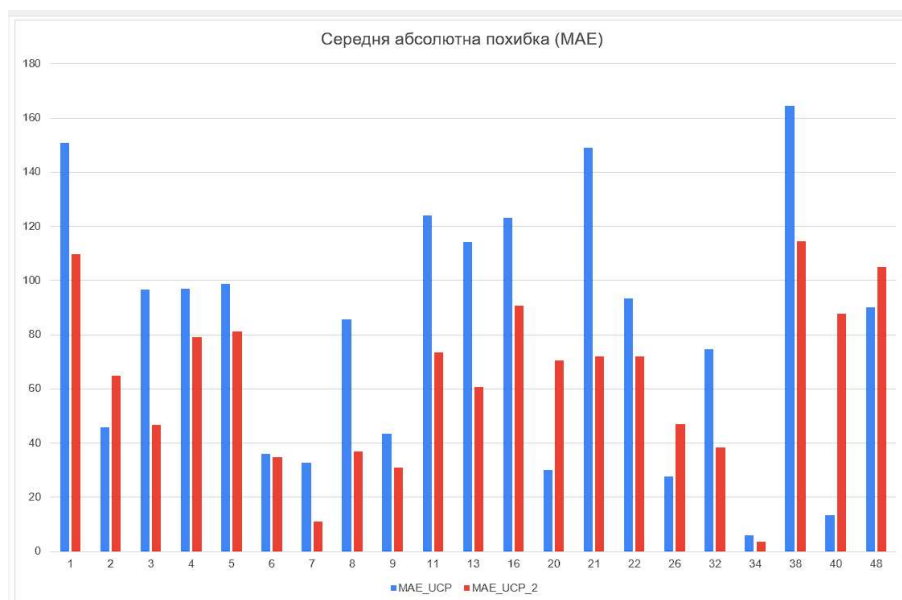


Рис. 1. Діаграма значень *MAE*

7. Висновки

У ході досліджень метод Use Case Points, який використовується для прогнозування трудовитрат, необхідних для успішного виконання ІТ-проекту, було удосконалено шляхом врахування ризиків при оцінці факторів технічної складності (*TCF*) та зовнішнього середовища (*ECF*), які суттєво впливають на значення остаточної оцінки.

Під час вирішення цього завдання було здійснено:

- вдосконалення методу *UCP* для оцінки трудовитрат необхідних при реалізації ІТ-проекту, яке дозволяє враховувати інтервальну природу оцінок факторів *TCF* та *ECF* завдяки використанню виразу (8);

- експериментальну перевірку працездатності і ефективності використання запропонованого методу та порівняння отриманих з його допомогою значень оцінки трудовитрат з оригінальним методом *UCP*.

На основі отриманих результатів, можна зробити висновок, що модифікований метод *UCP* може бути використано для підвищення точності прогнозування трудовитрат, що

суттєво підвищить ефективність процесів планування та управління майбутніми проектами.

Однак слід врахувати, що отримані результати базуються на аналізі лише одного набору даних. Тому, в перспективі, доцільно провести аналіз можливостей практичного застосування і точності запропонованого методу з використанням різних датасетів та більшого обсягу даних.

Перелік посилань:

1. Іваненко О. В. Дослідження моделей і методів оцінки трудовитрат при реалізації ІТ-проектів. *Радіоелектроніка та молодь у XXI столітті: матеріали 27-го Міжнар. молодіж. форуму, 10–12 травня 2023 р.* Харків: ХНУРЕ, 2023. Т. 6. Ч. 1. С. 188–189.
2. Ghafory H., Sahnosh F.A. The review of software cost estimation model: SLIM. *Journal of Advanced Academic Research*. 2020. № 2(4). P. 511–515. <https://doi.org/10.33545/27068919.2020.v2.i4h.447>
3. Pillai K., Sukumaran Nair V. S. A Model for Software Development Effort and Cost Estimation. *IEEE Transactions on Software Engineering*. 1997. Vol. 23. Iss. 8. P. 485–497. <https://doi.org/10.1109/32.624305>
4. Chirra S.M.R., Reza H. A Survey on Software Cost Estimation Techniques. *Journal of Software Engineering and Applications*. 2019. № 12(06). P. 226–248. <https://doi.org/10.4236/jsea.2019.126014>
5. Rashid J., Nisar M. W., Mahmood T., Rehman A., Arafat S. Y. A Study of Software Development Cost Estimation Techniques and Models. *Mehran University Research Journal of Engineering and Technology*. 2020. Vol. 39. No. 2. P. 413–431. <https://doi.org/10.22581/muet1982.2002.18>
6. Tripathi R., Rai Dr. P. K. Comparative Study of Software Cost Estimation Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2016. Vol. 6. P. 323–328.
7. Sinhal A., Bhupendra V. Software Development Effort Estimation: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2013. Vol. 3. P.1120–1135.
8. Petrov K., Kobzev I., Orlov O., Kosenko V., Kosenko A., Vanina Y. Devising a Method for Identifying the Model of Multi-criteria Expert Estimation of Alternatives. *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 4 № 3 (112). P. 56–65. <https://doi.org/10.15587/1729-4061.2021.238020>
9. Petrov K. E., Deineko A. O., Chala O. V., Panforova I. Y. The Method of Alternative Ranking for a Collective Expert Estimation Procedure. *Radio Electronics, Computer Science, Control*. 2020. № 2 (53). P. 84–94. <https://doi.org/10.15588/1607-3274-2020-2-9>
10. Gandomani T. J., Koh T. W., Binhamid A. A Case Study Research on Software Cost Estimation Using Experts. Estimates, Wideband Delphi, and Planning Poker Technique. *International Journal of Software Engineering and Its Applications*. 2014. Vol. 8. № 11. P. 173–182.
11. What Is a Use Case? URL: <https://www.wrike.com/blog/what-is-a-use-case/> (дата звернення: 19.06.2024).
12. What Is a Use Case? How to Write One, Examples, + Template. URL: <https://www.figma.com/resource-library/what-is-a-use-case/> (дата звернення: 19.06.2024).
13. Le Thi Kim Nhung H., Hoc H. T., Van Hai V. An Evaluation of Technical and Environmental Complexity Factors for Improving Use Case. In: Silhavy R., Silhavy P., Prokopova Z. (eds) *Software Engineering Perspectives in Intelligent Systems. CoMeSySo 2020. Advances in Intelligent Systems and Computing*. Springer, Cham, 2020. Vol 1294. P.757–768. https://doi.org/10.1007/978-3-030-63322-6_64
14. Estimating With Use Case Points. URL: <https://www.mountaingoatsoftware.com/articles/estimating-with-use-case-points> (дата звернення: 25.06.2024).
15. PERT: Definition, PERT Formula, PERT Chart, Technique & Example. URL: <https://pmstudycircle.com/pert-program-evaluation-and-review-technique/> (дата звернення: 27.06.2024).
16. Program Evaluation and Review Technique (PERT) Analysis. URL: <https://acqnotes.com/acqnote/tasks/pert-analysis> (дата звернення: 27.06.2024).
17. Silhavy R. Use Case Points Benchmark Dataset. Mendeley Data, 2017. V1. <https://doi.org/10.17632/2rfkjh3cn.1>

Надійшла до редколегії 29.05.2024

Петров Костянтин Едуардович, доктор технічних наук, професор, завідувач кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: kostiantyn.petrov@nure.ua, ORCID: <https://orcid.org/0000-0003-1973-711X> (науковий керівник здобувача вищої освіти Іваненко Ольги Валентинівни).

Іваненко Ольга Валентинівна, здобувач вищої освіти, група УППІТм-22-2, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: olha.ivanenko@nure.ua

Кобзев Ігор Володимирович, кандидат технічних наук, доцент кафедри інформатики та комп'ютерної техніки ХНЕУ ім. Семе́на Кузне́ця, м. Харків, Україна, e-mail: ikobzev12@gmail.com, ORCID: <https://orcid.org/0000-0002-7182-5814>