Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій

МЕТОДИЧНІ ВКАЗІВКИ

до практичних робіт з дисципліни "Економіка проєктів в комп'ютерній інженерії"
для студентів спеціальності "Комп'ютерна інженерія"

2025

Методичні вказівки до виконання практичних робіт з дисципліни «Економіка проєктів в комп’ютерній інженерії”/ Н.Я. Савка / Під ред. Л.О. Дубчак. Тернопіль: ЗУНУ, 2025. 33 с.

Укладачі: Н.Я. Савка, к.т.н., доцент кафедри комп’ютерної інженерії, Західноукраїнський національний університет

Відповідальний за випуск: Дубчак Л.О. к.т.н., доцент, завідувач кафедри комп’ютерної інженерії, Західноукраїнський національний університет

Рецензенти: Мудрик І.Я., доктор філософії, доцент кафедри програмної інженерії, Тернопільський національний технічний університеит ім. Івана Пулюя

Гончар Л.І., к.е.н., доцент кафедри комп’ютерних наук, Західноукраїнський національний університет

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри комп’ютерної інженерії протокол № 4 від 28 жовтня 2025 р.

ЗМІСТ

ВСТУП

Економічне обгрунтування проєктів в комп'ютерній інженерії являє комплексний передпроектний документ, базується на аналізах і розрахунках різних показників, що характеризують ефективність розробки. На підставі всіх розрахунків і аналітичних даних формують висновки про економічну доцільність реалізації проєкту, дають оцінку перспектив його впровадження.

Основною метою економічного обґрунтування розроби проектів в комп'ютеній інженерії – дати фінансову оцінку передбачуваних витрат та одержуваного ефективного результату, а також оцінити прибутковість проєкту і, в кінцевому підсумку, економічну доцільність його розробки та впровадження.

Варто зазначити, що обов'язковою складовою будь-якого інжинірингового проєкту є фінансові витрати на різних етапах виконання робіт. Відповідно, важливо вірно здійснити фінансову оцінку передбачуваних витрат, продуктивність, корисність та, в результаті, економічну ефективність проєкту.

Наукоємні розробки та дослідження, на відміну від корпоративних, не завжди мають за мету отримання прибутку або ж іншої матеріальної вигоди. В багатьох випадках, проєкти наукового спрямування не є економічно вигідними. Однак, вони є рушійною силою прогресу, дослідженням незвіданих галузей та проблем, які, у свою чергу, завжди впливають на майбутні різнопланові розробки. Зважаючи на вищезазначене, досить часто наукова діяльність стимулюється зовнішніми інвестиціями та підтримкою держаних інститутів, міжнародних грантів. Із використанням результатів досліджень та на основі отриманих моделей можна робити прогнози по впровадженню нових методик та принципів організації роботи різних установ.

В той же час, для отримання відмінних результатів експериментів та доцільності розробки комп'ютерних систем потрібні відповідні затрати на дослідження та розробку, що становить основу витрат, які будуть здійснені протягом підготовки, виконання та реалізації відповідного проєктного рішення. Враховуючи залежність якості кінцевого продукту від кваліфікації розробників, варто сконцентрувати увагу на якості та результативності відповідної розробки. Для цього необхідно провести ґрунтовний аналіз предметної області, залучити найновіші та інноваційні технології, провести ґрунтовне тестування та оцінку результатів розробки.

ПРАКТИЧНА РОБОТА № 1
МОДЕЛІ ЦІНОУТВОРЕННЯ ПРИ УКЛАДАННІ УГОД НА РОЗРОБКУ ІТ ПРОДУКТІВ

Мета: оволодіти теоретичними та практичними знаннями, що стосуються моделей ціноутворення на ІТ продукти.

Завдання

1. Ознайомитися із основними моделями ціноутворення, що існують на ринку ІТ продуктів.
2. Провести порівняльний аналіз моделей, виокремити основні їх переваги та недоліки.
3. Обгрунтувати модель ціноутворення на розробку, на яку поступає замовлення у ІТ компанію.

Теоретичні відомості

Сучасна сфера ІТ послуг з кожним днем стає все ширшою. Разом з нею відбувається активний ріст кількості людей, що потребують допомоги ІТ спеціалістів. Починаючи з молодих стартаперів, які мають на меті реалізувати власні ідеї, та закінчуючи досвідченими представниками бізнесу, що знаходяться в процесі постійного пошуку нових шляхів вдосконалення власних проектів та рішень. Усі вони щоденно стикаються із необхідністю взаємодії з представниками ІТ сфери на предмет створення нових комп'ютерних систем.

Варто зазначити, що у переговорах щодо розробки будь-якої комп'ютерної системи беруть участь дві сторони – замовник та виконавець (підрядна організація, підрядник). Одним із основних та найважливіших аспектів будь-яких перемовин щодо розробки ІТ продукту є вибір моделі фінансової взаємодії. Найбільш популярними та усталеними моделями оплати, що нині використовуються як великими компаніями, так і звичайними фріланс-командами, є Fixed Price та Time and Material [6]. Також існують такі моделі оплати як Fixed Budget та оплата послуг Dedicated Teamn. Надалі охарактеризуємо кожну модель окремо, щоб з'ясувати їх основні відмінності.

1.1 Fixed Price – модель оплати, за якою замовник сплачує фіксовану суму за результат роботи. Доцільно та ефективно може використовуватись у невеликих за обсягом та складністю проектах, де всі процеси, ризики та необхідні ресурси відомі і розраховані ще на початковому етапі. Тож, як наслідок, може бути сформована та виставлена фіксована ціна.

Випадки застосування такої моделі:

- замовник має чіткі вимоги до підрядника, власне бачення та розуміння продукту (інакше кажучи: клієнт знає, який конкретний результат йому потрібен);

- підрядник має великий досвід з організації та управління процесами, що виникають на шляху реалізації проекту, та існує можливість об’єктивно оцінити весь обсяг робіт, включаючи потенційні ризики;

- обидві сторони мають однакове уявлення щодо змісту та обсягу робіт.

Зазвичай підрядник встановлює найвищі ставки, а також закладає додаткові витрати до загального бюджету, щоб застрахувати себе від непередбачених обставин, тому при використанні такої моделі, замовник також платить за можливі ризики.

Слід врахувати і негативні аспекти вказаного підходу: Fixed Price потребує деталізованого технічного завдання (ТЗ); Fixed Price – це завжди компроміс якості та економії, оскільки саме наявність чітких вимог та специфікацій на початковому етапі зумовлюють відсутність гнучкого підходу та необхідності постійної взаємодії між сторонами, що може призвести до неочікуваних для замовника результатів, а також потребує внесення постійних змін до ТЗ.

З метою задоволення обох сторін, весь процес роботи над IT продуктом можна поділити на фази (етапи або майлстоуни), кожна з яких має окрему, чітко встановлену суму. У такому випадку кожна фаза обговорюється окремо, обсяг робіт формується за результатами попереднього етапу, що значно зменшує ризики.

1.2 Time & Material (T&M) – це тип оплати, в умовах якого замовник сплачує за процес розробки проекту: за час роботи, який формується виходячи з погодинних ставок членів команди підрядника, та за поточні ресурси, які використовуються в процесі. Така модель використовується в складних та динамічних проектах, де на початковому етапі досить важко повністю встановити майбутній функціонал та бажаний результат. У випадку використання T&M, процес розробки проекту стає цілком прозорим та зрозумілим для замовника. Він здійснює повний контроль та керування процесом на кожному з етапів, маючи при цьому можливість постійно вносити зміни у технічні завдання та змінювати пріоритети. Внаслідок того замовник самостійно несе усі потенційні ризики, пов’язані з якістю продукту та його кінцевим виглядом.

Модель може бути застосована у таких випадках:

- замовник не має можливості сформулювати чіткі, детальні вимоги до продукту та готовий до внесення корективів у початковий план проекту в подальшому;

- відсутня можливість передбачити всі вимоги, ризики та ресурси на початковому етапі;

- проект пов’язаний з ринками, що розвиваються, новими технологіями або неперевіреними об’єктами, де кожного дня курси та темпи розвитку можуть змінюватися, а отже і виникати нові потреби у функціоналі продуктів;

- бажання замовника контролювати процес розробки та наявність постійного потоку задач, що розкидані у часі і не можуть бути передбачені заздалегідь.

Незважаючи на прозорість та взаємовигідність використання такого інструменту взаємодії сторін, відсутність чіткого бюджетного плану та “дедлайнів” можуть призвести до зайвих витрат з боку замовника та недостатньої мотивації для вчасного закінчення проекту з боку підрядника.

Отже, як можна побачити, кожен з підходів має як позитивні, так і негативні аспекти, в залежності від конкретних обставин. Але чи можна об’єднати окремо взяті особливості кожної з моделей та запропонувати альтернативний варіант? Звичайно, що так. Наприклад, цікавим рішенням для ситуацій, коли проект має обмежений бюджет, але замовник бажає постійно взаємодіяти з розробниками та впливати на процес реалізації свого продукту, є Fixed budget модель.

1.3 Використовуючи **Fixed budget**, замовник має змогу постійно вдосконалювати функціонал власного продукту та управляти запланованим обсягом робіт в процесі його виконання, не виходячи при цьому за рамки фіксованої ціни та строків проекту. Що стосується підрядника, то при взаємодії за таким підходом він наперед знає, скільки часу витратить на роботу. Його завдання зводиться до тісної співпраці із замовником, що полягає у правильному встановлені пріоритетів та балансуванні процесом реалізації проекту таким чином, аби побудувати максимально якісний продукт у встановлених грошових та часових рамках. Таким чином, планування та взаєморозуміння сторін відіграє вирішальну роль у досягненні бажаного результату, а фінансові ризики зводяться до мінімуму. Зазначена модель варта уваги нових та динамічних проектів, що оперують незначними сумами, але мають в своєму арсеналі багато цікавих ідей.

1.4 Dedicated Team – “аутстаф-модель”, згідно якої під проект замовника виділяється окрема команда підрядника, з урахуванням його особистих вимог та потреб, яка концентрується лише на проекті замовника. Особливістю цього підходу є те, що замовник сплачує фіксовану суму кожного місяця та особисто несе відповідальність не тільки за навантаження, але й за простій членів команди. Таким чином, замовник отримує повний управлінський контроль над проектом і командою, а підрядник виконує функцію рекрутера персоналу та адміністративної підтримки. Модель переважно використовується у великих та об’ємних проектах зі створення продуктів, що потребуватимуть підтримки та розвитку в майбутньому.

Ця модель використовується з метою економії фінансових та трудових ресурсів, коли:

- замовник бажає отримати вмотивованих та зацікавлених у власному проекті спеціалістів, досвід та особисті якості яких відіграють вирішальне значення у формуванні та забезпеченні продукту, але не має можливості та ресурсів самостійно підбирати та утримувати команду у власному приміщенні;

- замовнику потрібна лояльна команда зовнішнього персоналу, з якою клієнт може встановлювати ті ж робочі відносини і правила, що і з основним персоналом (команда розділяє корпоративну культуру, стиль управління та методологію замовника).

Основним недоліком цієї моделі є те, що при її використанні витрати замовника значно більші, а ніж при використанні інших варіантів. Ще одним негативним фактом виступає потенційна можливість відкладання початку проекту через надто тривалий підбір команди, в той час, як при використанні T&M моделі робота може початися швидше.

Проте важливо розуміти, що згуртована та стабільна команда є головним активом при розробці продукту. При цьому потрібно брати до уваги вірогідність ситуації, за якої відносини з розробниками вже завершено, а продукт, розроблений в умовах Fixed Price або T&M моделей, терміново потребує подальшого вдосконалення або ж раптом відмовляється функціонувати належним чином. За таких умов наявність дійсної “Dedicated Team”, тобто такої, де спеціалісти не лише виконують усі умови технічного завдання замовника, але й знайомі з усім циклом побудови продукту та в силі підтримати його протягом усього часу його існування, стає очевидною.

Для наочного розуміння істотних відмінностей між проаналізованими моделями ціноутворення розроблено таблицю 1.1, у якій наведено особливості кожної з них.

Таблиця 1.1 – Порівняльний аналіз моделей ціноутворення на IT продукти

Модель ціноутворення	Fixed Price	T&M	Fixed budget	Dedicated Team
Статичні показники	Обсяг задач, ціна, терміни	Якість	Ціна, якість, терміни	Ціна, якість
Гнучкі показники	Якість	Обсяг задач, ціна, терміни	Обсяг задач	Обсяг задач, терміни
Ризики	Рівень ціни роботи впливає на якість	Збільшення обсягів роботи впливають на ціну	Необхідний високий рівень взаєморозуміння; пріоритетність задач	Високий рівень витрат; тривалий підбір команди
Бюджет замовника	Фіксований з прогнозованими потребами	Плаваючий	Обмежений	Фіксований
Витрати замовника	Поетапно / по завершенню проекту	Погодинна ставка + використані ресурси	Поетапно / по завершенню проекту	Щомісячно
Ступінь залученості замовника	Мінімальна	Максимальна	Максимальна	Максимальна
Особливість моделі	Потребує деталізованого	Оплата тільки фактично	Можливість управління	Концентрація команди

	ТЗ	виконаної роботи	процесом розробки при фіксованому бюджеті	підрядника лише на проекті замовника

Варто звернути увагу на те, що шалений попит на ІТ послуги вносить свої корективи до існуючих моделей ціноутворення. З'являється усе більше змішаних альтернатив, оскільки самі відносини передбачають гнучкість, унікальність кожної ситуації та налаштованість сторін на досягнення кінцевого результату. Вибір фінансової моделі взаємодії залежить від багатьох факторів, серед яких: динаміка проекту, зрілість та стабільність бізнесу, необхідність взаємодії між замовником та підрядником, наявність чітких вимог до продукту та багато іншого. Важливо ретельно підходити до питання вибору правильної моделі, оскільки легковажність може призвести до плачевних наслідків. Саме від обраного підходу будуть залежати фінансові показники обох сторін, якість та ефективність майбутнього продукту.

Хід роботи

1. Ознайомитися із особливостями ринку ІТ послуг та шляхів формування конкурентоздатних цін та ІТ проекти.
2. Охарактеризувати особливості розробки ІТ продукту (приклад обрати індивідуально), скласти технічне завдання.
3. Обґрунтувати модель ціноутворення при укладанні угоди на розробку ІТ продукту.
4. Проаналізувати основні переваги та недоліки обраної моделі.

Питання для самоконтролю

1. Що таке модель ціноутворення на ІТ продукт?
2. Які моделі оплати за розробку у галузі ІТ ви знаєте?
3. За яких умов доцільно застосувати модель Time & Material?
4. Коли ефективною є модель Dedicated Team?
5. Які негативні аспекти моделі Fixed Price?

ПРАКТИЧНА РОБОТА №2
ОСНОВНІ ЕКОНОМІЧНІ ПОКАЗНИКИ РОЗРОБКИ ІТ-ПРОЄКТІВ

Мета: навчитися проводити розрахунки основних економічних показників розробки ІТ-проєктів й на їх основі обґрунтовувати доцільність впровадження.

Завдання

1. Провести розрахунок трудомісткості розробки ІТ-проєкту.

2. Розрахувати витрати на розробку.

3. Розрахувати можливу ціну проєкту.

4. Провести економічне обґрунтування вибору комплексу технічних і програмних засобів.

5. Дослідити аналоги та порівняти їх із поточним проєктом.

6. Здійснити оцінку економічної ефективності розробки ІТ-проєкту.

Теоретичні відомості

Економічне обґрунтування розробки ІТ-проєктів включає етапи [5]:

– опис технологічного процесу розробки із зазначенням трудомісткості кожної операції;
– визначення витрат на оплату праці розробників;
– визначення матеріальних витрат;
– розрахунок транспортних витрат;
– визначення амортизаційних відрахувань;
– визначення накладних витрат;
– складання кошторису та визначення собівартості розробки;
– розрахунок прогнозованої ціни;
– обґрунтування вибору комплексу технічних і програмних засобів;
– оцінка показників економічної ефективності розробки ІТ-проєкту.

2.1 Визначення стадій технологічного процесу. Для визначення загальної тривалості етапів розробки ІТ-продукту доцільно сформувати таблицю 2.1.

Таблиця 2.1 – Стадії розробки ІТ продукту

№ п/п	Назва операції (стадії, етапи)	Виконавець, посада	Середній час виконання операції, год.
1	Підготовка, складання ТЗ (ініціалізація ідеї, аналітика та планування, формування вимог)		
2	Проєктування		
3	Розробка		
4	Тестування		
5	Впровадження		
6	Підтримка та експлуатація		
Разом			

2.2 Розрахунок трудомісткості розробки. Для визначення трудомісткості насамперед складається перелік всіх основних етапів і видів

робіт, які повинні бути виконані (див. табл. 2.1). При цьому особлива увага приділяється логічному впорядкуванню послідовності окремих видів робіт і виявленні можливостей їх паралельного виконання, що дозволяє суттєво скоротити загальну тривалість розробки. Форму поділу робіт по етапах із зазначенням трудомісткості їх виконання наведена в таблиці 2.2.

Таблиця 2.2 – Розподіл робіт по етапах і видах розробки проєжкту та оцінка їх трудомісткості

Етап розробки КС	Вид роботи	Трудомісткість розробки, людино × год.
Разом трудомісткість розробки КС		

2.3 Розрахунок витрат на розробку. Визначення витрат на розробку проєкту здійснюється шляхом складання відповідного кошторису, який включає:

- витрати на оплату праці;
- матеріальні витрати;
- податкові відрахування;
- амортизація основних фондів;
- накладні витрати;
- транспортні витрати;
- інші витрати.

До матеріальних витрат включають витратні матеріали, та комплектуючі необхідні для функціонування ІТ-проєкту. Загальну суму витрат на матеріали B_M визначають за формулою:

$$B_M = \sum_{i=1}^{n} K_i \cdot Ц_i, \qquad (2.1)$$

де K_i – кількість *i*-го типу матеріалу;

$Ц_i$ – ціна за одиницю *i*-го типу матеріалу;

i – тип матеріального ресурсу;

n – кількість матеріалів.

На основі цього формуємо таблицю 2.3.

Таблиця 2.3 – Витрати на матеріали

Найменування	Виробник (модель)	Одиниця вимірювання	Кількість	Ціна за одиницю, грн	Сума, грн.
Разом					

Розрахунок витрат на оплату праці

До цієї статті відносяться витрати на заробітну плату праці всіх розробників (керівник, розробник, програміст, тестувальник, монтажник і т.п.). Витрати на оплату праці розробників проєкту визначають за формулою:

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij} \,, \qquad (2.2)$$

де n_{ij} – чисельність спеціалістів i-ої спеціалізації j-ї кваліфікації, осіб;

t_{ij} – затрачений час на розробку проекту спеціалістом i-ої спеціалізації j-ї кваліфікаці, год;

C_{ij} – годинна ставка працівника i-ої спеціалізації j-ї кваліфікаці, грн.

Витрати на оплату праці заносимо у таблицю 2.4.

Таблиця 2.4 – Витрати на оплату праці

№ п/п	Посада виконавця	Час розробки, год.	Погодинна заробітна плата, грн	Витрати на оплату праці, грн
РАЗОМ витрати на заробітну плату				

Податкові відрахування

До цієї статті включають витрати на оплату єдиного соціального внеску (ЄСВ), передбаченого чинним законодавством України у розмірі 22 %, а також податку з доходів фізичних осіб (ПДФ) у розмірі 18 % та військовий збір – 5 %. Ці відрахування здійснюють від суми нарахованої зарплати. Якщо розробник проєкту є фізична особа підприємець (ФОП), то податкові відрахування здійснюють, залежно від того, до якої групи ФОП належить особа-розробник. Суму податкових відрахувань можна розрахувати за формулою:

$$B_{П} = k \cdot B_{ОП}, \qquad (2.3)$$

де k – коефіцієнт податкових відрахувань.

До амортизаційних відрахувань включають суму відрахувань від вартості обладнання і приладів, що використовуються при розробці ІТ-проєктів:

$$B_{АМ} = \sum_{i=1}^{n} \frac{B_i \cdot H_i \cdot T_i}{100 \cdot T_{ЕФi}} \,, \qquad (2.4)$$

де B_i – балансова вартість i-го устаткування, *грн.;*

H_i – річна норма амортизації i-го устаткування, %;

T_i – час роботи i-го устаткування за весь період розробки, год.;

$Т_{ЕФi}$ – ефективний фонд часу роботи i-го устаткування за рік, год / рік;

i – тип устаткування;

n – кількість устаткування.

Балансову вартість устаткування розраховуємо за формулою:

$$B_{Б} = Ц_{р} * (1 + K_{УН}), \quad (2.5)$$

де $Ц_{р}$ – ринкова вартість устаткування,

$K_{УН}$ – коефіцієнт, що враховує витрати на установку й налагодження устаткування (приймається рівним 12%).

Усі розрахунки заносимо у таблицю 2.5.

Таблиця 2.5 – Амортизація основних фондів

Найменування устаткування	Вартість устаткування, грн.	Річна норма амортизації,%	Ефективний фонд часу роботи обладнання, год / рік	Час роботи обладнання для розробки проєкту, год	Сума, грн.
Разом амортизаційні відрахування					

Річні норми амортизації обладнання визначаються за довідником (60 % для комп'ютерної техніки) або виходячи з можливого терміну корисного використання устаткування:

$$H_{i} = \frac{100}{T_{Hi}}, \quad (2.6)$$

де T_{Hi} – можливий термін використання i-го устаткування, рік. Визначається залежно від обладнання (для комп'ютерної техніки становить 5 років).

Накладні витрати пов'язані із утриманням апарату управління підприємства (фірми) та створення необхідних умов праці. В залежності від організаційно-правової форми діяльності суб'єкта, накладні витрати можуть становити 60–100 % від суми заробітної плати працівників.

$$B_{H} = 0{,}7 \cdot B_{ОП}, \quad (2.7)$$

Кошторис витрат на розробку ІТ-проєкту. Загальні витрати ($B_{КС}$) розраховують за формулою:

$$B_{КС} = B_{ОП} + B_{П} + B_{М} + B_{АМ} + B_{Н}. \quad (2.8)$$

Результати проведених розрахунків заносимо у таблицю 2.6.

Таблиця 2.6 - Кошторис витрат на розробку проєкту

Статті витрат	Сума, грн.
Матеріальні витрати Витрати на оплату праці Податкові відрахування Амортизаційні відрахування	

Накладні витрати	
РАЗОМ (собівартість розробки)	

2.4 Прогнозована ціни ІТ-проєкту. Величину можливої ціни проєкту визначають з урахуванням ефективності, якості і термінів виконання проєкту на рівні, що відповідає економічним інтересам замовника і виконавця (підрядника) через порівняння із аналогами Для прикладних проектів прогнозовану ціну розраховують за формулою:

$$Ц_{\text{П}} = B_{КС} \cdot \left(1 + \frac{P}{100}\right), \qquad (2.9)$$

P – середній рівень рентабельності проєкту, % (приймається в розмірі 20-30%).

Показник рентабельності характеризує частку чистого приведеного доходу, що припадає на одиницю дисконтованих в період життєвого циклу проєкту інвестиційних вкладень:

$$P = \frac{Прибуток}{Собівартість} \times 100\%.$$

2.5 Оцінка показників економічної ефективності розробки ІТ-проєктів. За міжнародним стандартам для оцінки ефективності розробки проєктув і галузі ІТ застосовують такі показники:

- внутрішня норма дохідності;
- чистий приведений дохід;
- рентабельність;
- термін окупності.

Показник внутрішньої дохідності характеризує величину чистого прибутку (чистого валового доходу), що припадає на одиницю інвестиційних вкладень у кожному часовому інтервалі життєвого циклу проєкту. Розрахунок цього показника виконується за такою формулою:

$$\sum_{i=0}^{T} \frac{Di}{(1+q)^i} - \sum_{i=0}^{T} \frac{Ki}{(1+q)^i} = 0, \qquad (2.10)$$

де D_i – дохід від реалізації у i-му періоді;

K_i – інвестиційні вкладення в i-му періоді з урахуванням інфляційних процесів;

i – періоди виконання і впровадження проєкту;

T – загальний період (тривалість) життєвого циклу проекту;

q – показник внутрішньої норми дохідності.

Показник інвестиційних вкладень (капітальних витрат на розробку) з урахуванням інфляційних процесів обчислюємо за формулою:

$$K_i = \varphi_i * R_i, \qquad (2.11)$$

де φ_i – коефіцієнт інфляції на поточний період;

R_i – інвестиційні платежі в i-му періоді (капітальні вкладення в проєкт).

Дохід від реалізації ІТ-проєкту у *і*-му періоді розраховуємо за формулою:

$$D_i = J_i(Ц_{Пі} - C_i), \quad (2.12)$$

де $Ц_{Пі}$ – ціна продажу програмного продукту в *і*-му періоді;

C_i – собівартість розробки ІТ- продукту;

J_i – кількість проєктів.

У ринкових умовах при ціновій політиці, що змінюється, показник терміну окупності є одним із основних. Він визначається на основі величини капітальних витрат по періодах розробки ІТ-продукту та величини фактичних чи прогнозних доходів:

$$\sum_{i=0}^{T} Ki = \sum_{i=0}^{T} Di, \quad (2.13)$$

де T – термін окупності.

Економічна ефективність полягає у відношенні результату від розробленого ІТ-продукту до затрачених ресурсів:

$$E = \frac{D_i}{B_{КС}}. \quad (2.14)$$

Тоді термін окупності можна розрахувати за такою формулою:

$$T = \frac{1}{E}. \quad (2.15)$$

На основі проведених вище розрахунків формуємо таблицю 2.7.

Таблиця 2.7 – Показники ефективності розробки проєкту

Показник	Значення
Капітальні витрати на розробку, грн	
Плановий прибуток, грн	
Прогнозована ціна, грн.	
Економічна ефективність	
Термін окупності, рік	

Враховуючи основні економічні показники, що стосуються розробки ІТ-проєктів, формують висновок щодо доцільності розробки та впровадження проєкту. Якщо отримано суттєвий економічний ефект від розробки, а термін окупності капітальних вкладень не більший 10 років (залежно від специфіки ІТ-проєкту), то така розробка є економічно вигідною та конкурентоздатною на ринку подібних ІТ продуктів

Хід роботи

1. Обрати приклад ІТ проєкту.

2. Провести розрахунки основних економічних показників прозробки проєкту.

3. Сформувати висновки щодо доцільності розробки та впровадження ІТ-продукту на основі показників економічної ефективності.

Питання для самоконтролю

1. Які етапи включає розробка ІТ-проєкту?
2. Що входить до матеріальних витрат?
3. Які є основні витрати на розробку проєкту?
4. Що таке податкові відрахування?
5. Як і для чого нараховується амортизація?
6. Які показники визначають економічну ефективність розробки проєкту?

ПРАКТИЧНА РОБОТА № 3
ТРУДОМІСТКІСТЬ РОЗРОБКИ МОДЕЛІ КОРПОРАТИВНОЇ МЕРЕЖІ

Мета: навчитися розраховувати трудомісткість розробки моделі корпоративної мережі, проводити аналіз отриманих показників й формувати висновки.

Завдання:

1. Ознайомитися із етапами розробки моделі корпоративної мережі.
2. Визначити трудомісткість процесу розробки моделі корпоративної мережі.

Теоретичні відомості

Трудомісткість – це затрати робочого часу людиною певної кваліфікацій на виробництво одиниці продукції [7]. Трудомісткість розробки моделі корпоративної мережі включає витрати часу на дослідження корпоративної мережі і проведення аудиту на підприємстві, витрати часу на розробку моделі після проведеного аудиту, складання моделі по готовому проекту, на підготовку документації, редагування і оформлення документації. трудомісткість – це затрати робочого часу на виробництво одиниці продукції.

Кожен етап розробки моделі корпоративної мережі вимагає залучення спеціалістів, які мають певну кваліфікацію та відповідний стаж роботи. Вважається, що чим більший стаж роботи, тим більший його досвід у тій чи іншій галузі. Зважаючи на це, у таблиці 3.1 зазначено умовний коефіцієнт кваліфікації розробника, що випливає із стажу його роботи.

Таблиця 3.1 – Кваліфікація розробника

Стаж роботи	Коефіцієнт КР

до 2-х років	0,8
2-3 роки	1
3-5 років	1,1-1,2
5-7 років	1,3-1,4
понад 7 років	1,5-1,6

3.1 Витрати часу на дослідження корпоративної мережі і проведення аудиту на підприємстві розраховуємо за формулою:

$$t_u = \frac{Q * B}{S * KR}, \quad (3.1)$$

де Q – умовна кількість одиниць техніки і програмного забезпечення, яке застосовується для розробки корпоративної мережі;

KR – коефіцієнт, що залежить від кваліфікації розробника;

B – коефіцієнт збільшення витрат, пов'язаний з неповнотою опису і необхідності уточнень і доробок; приймається B = 15;

S – коефіцієнт, що визначається складністю задачі; для цього випадку S = 80.

3.2 Витрати часу на розробку моделі захисту після проведеного аудиту на підприємстві розраховуємо за формулою:

$$t_p = \frac{Q}{S * KR}, \quad (3.2)$$

де S коефіцієн для цього етапу приймається S=20.

3.3 Витрати часу на складання моделі корпоративної мережі по готовому проекту t_n розраховуються за формулою (3.2), проте коефіцієнт, який визначається складністю задачі для цього етапу становить S=50.

3.4 Витрати часу на підготовку документації $t_{док}$ розраховуються за формулою (3.2), проте коефіцієнт, який визначається складністю задачі для цього етапу становить S=20.

3.5 Витрати часу на редагування оформлення документації розраховуються за формулою:

$$t_{ред} = 0.75 * t_{док}. \quad (3.3)$$

3.6 Трудомісткість розробки моделі корпоративної мережі обчислюємо за формулою:

$$T_{заг} = t_u + t_p + t_n + t_{док} + t_{ред}. \quad (3.4)$$

З врахуванням складності мережі та технологічних умов, формула 3.4 набуде вигляду:

$$T = T_{заг} \cdot K_c \cdot K_{ТУ}, \quad (3.5)$$

де K_c – коефіцієнт складності мережі (див. таблиця 3.2);

$K_{ТУ}$ – коефіцієнт технологічних умов (див. таблиця 3.3);

Таблиця 3.2 – Коефіцієнти складності мережі

Категорія складності	**Опис**	K_c
Проста	типова структура, невелика кількість вузлів	1.0
Середня	кілька підмереж, базові маршрутизатори, сервер	1.2
Складна	масштабна мережа з VLAN, VPN, балансуванням	1.4–1.6

Таблиця 3.3 – Коефіцієнти технологічних умов

Умови розробки	$K_{ТУ}$
Використання готових шаблонів	0.8
Типові засоби налаштування	1.0
Нове обладнання, власні скрипти	1.2

Отже, при розрахунку показника трудомісткості розробки моделі корпоративної мережі отримуємо загальну кількість людино-годин, яка необхідна для розробки корпоративної мережі.

Хід роботи

1. Розробити проєкт корпоративної комп'ютерної мережі із вказанням кількості одиниць комп'ютерної техніки та необхідного програмного забезпечення.
2. Розрахувати трудомісткість процесу розробки моделі корпоративної мережі.
3. Обґрунтувати ефективність розробки корпоративної мережі на основі показника трудомісткості.

Питання для самоконтролю

1. Що таке трудомісткість розробки?
2. Які етапи розробки моделі корпоративної мережі?
3. Що таке коефіцієнт кваліфікації розробника?
4. Які показники включає трудомісткість розробки моделі захисту корпоративної мережі?

ПРАКТИЧНА РОБОТА № 4
ОЦІНКА ВАРТОСТІ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ НА ОСНОВІ МЕТОДУ ФУНКЦІОНАЛЬНИХ ТОЧОК

Мета: навчитися оцінювати вартість програмної системи методом функціональних точок, вміти класифікувати функцій, розраховувати нефакторовані та скориговані функціональні точки.

Завдання:

1 Ознайомитися із особливостями методу оцінки вартості проєктів на основі функціональних точок.

2. Оцінка функціональних елементів проєкту.

2. Обчислення вартості розробки програмної системи шляхом проведення послідовних розрахунків.

Теоретичні відомості

Метод функціональних точок (Function Point Analysis, FPA) – це стандартизований підхід до оцінки розміру програмного забезпечення, що включає розрахунок трудомісткості та вартості його розробки. Метод запропоновано Аланом Албрехтом (IBM, 1979) і згодом формалізовано міжнародною асоціацією IFPUG [1-4].

На відміну від підрахунку рядків коду, метод функціональних точок базується на оцінці функціональності, яку бачить користувач. Вартість системи визначається не кількістю коду, а кількістю реалізованих функцій та їхньою складністю. Усі функції системи поділяють на п’ять класів.

1. Зовнішні введення (EI) – процеси введення даних у систему (наприклад, форми введення).
2. Зовнішні виведення (EO) – процеси формування вихідних результатів (звіти, файли, повідомлення).
3. Запити (EQ) – процеси, що включають введення і виведення без значної логіки обробки (пошук, прості запити).
4. Внутрішні логічні файли (ILF) – дані, що зберігаються та обробляються всередині системи.
5. Зовнішні інтерфейсні файли (EIF) – файли або дані, що використовуються системою, але підтримуються іншою системою.

Для кожного класу функцій встановлено ваги залежно від складності (низька / середня / висока) (див. таблиця 4.1). Сума вагових коефіцієнтів по всіх функціях формує нефакторовані функціональні точки (UFP):

$$UFP = \sum_{i=1}^{5} FP \qquad (4.1)$$

Таблиця 4.1 – Ваги класів функцій

Клас функції	Низька складність	Середня складність	Висока складність
Зовнішні введення (EI)	3	4	6
Зовнішні виведення (EO)	4	5	7
Запити (EQ)	3	4	6
Внутрішні логічні файли (ILF)	7	10	15
Зовнішні інтерфейсні файли (EIF)	5	7	10

Для врахування особливостей проєкту вводять 14 технічних та середовищних факторів General System Characteristics (GSC)), зокрема:

– комунікаційні можливості (Data Communications) – наскільки система залежить від мережевих технологій;

– розподіленість обробки (Distributed Data Processing) – чи дані обробляються на кількох вузлах;

– продуктивність (Performance) – наскільки високі вимоги до швидкодії;

– інтенсивність використання (Heavily Used Configuration) – навантаження на апаратне та програмне забезпечення;

– обсяг транзакцій (Transaction Rate) – кількість операцій / запитів у системі;

– онлайн введення даних (Online Data Entry) – частка системи, що працює з інтерактивним введенням;

– ефективність користувача (End-User Efficiency) – вимоги до зручності, інтерфейсу, мінімізації дій користувача;

– оновлення даних онлайн (Online Update) – наскільки часто дані змінюються в реальному часі;

– складність процесів (Complex Processing) – рівень алгоритмічної обробки та логічних умов;

– повторне використання програмного забезпечення (Reusability) – чи планується використання компонентів повторно;

– легкість інсталяції (Installation Ease) – вимоги до швидкої інсталяції, налаштування;

– легкість експлуатації (Operational Ease) – вимоги до автоматизації адміністрування та зручності підтримки;

– множинність сайтів / платформ (Multiple Sites) – чи система розгортається на кількох майданчиках або з різними конфігураціями;

– адаптивність (Facilitate Change) – наскільки система має бути гнучкою для майбутніх змін.

Кожен фактор оцінюється від 0 до 5 балів. Сума значень кожного фактора формує коефіцієнт впливу, що використовується для коригування базового функціонального розміру:

$$\text{Total Degree of Influence (TDI)} = \sum_{i=1}^{14} \text{GSC}_i . . \qquad (4.1)$$

Якщо значення TDI дорівнює 0, то програмна система дуже проста, при значенні показника 70 – систему вважають складною із високими вимогами.

Формула для розрахунку скоригованих функціональних має вигляд:

$$\text{Adjusted FP} = \text{UFP} \times (0.65 + 0.01 \times \text{TDI}), \qquad (4.2)$$

де 0.65 – базовий коефіцієнт, 0.01 – надбавка до коефіцієнта TDI за рахунок складності.

Таким чином, AFP враховує не лише функціональність системи, а й загальну складність реалізації,яку ідентифікують 14-ма факторами.

Трудомісткість розробки програмної системи обчислюють на основі формули:

$$L = AFP \times H, \qquad (4.3)$$

де *H* – нормативна кількість годин на 1 функціональну точку (FP) (залежить від типу програмної системи – 10-15 год/FP для невеликого проєкту, для складних систем – 20-25 год/FP).

Вартість проєкту при цьому обчислюють за формулою:

$$C=L\times R+Cother+Crisk \quad (4.4)$$

де *R* – вартість однієї години роботи, *Cother* – інші витрати (тестування, ліцензії, документація, інфраструктура (сервери, хостинг, обладнання), непрямі витрати (адміністративні, транспортні, офіс);

Crisk – резерв на ризики (покриття непередбачуваних витрат), зокрема, затримки виконання, зриви постачання, помилки у проєктуванні, потребу в додаткових ресурсах, зміни вимог замовника.

Розшифровуючи інші витрати (*Cother*), формула для обчислення загальної вартості проєкту набуде вигляду:

$$Ctotal=C+Cinfra+Clicense+Csupport+Coverhead+Crisk, \quad (4.5)$$

де *Cinfra* – витрати на інфраструктуру,

Clicense – вартість ліцензій,

Csupport – вартість підтримки,

Coverhead – непрямі витрати (адміністративні, транспортні, оренда приміщення, комунальні послуги, маркетинг, документообіг).

$$Crisk=P\times I, \quad (4.6)$$

де *P* – ймовірність настання ризику (0–1 або у %);

I – потенційний вплив (збитки у грошовому вираженні).

Перевагами такого методу оцінки вартості програмної системи є незалежність від мови програмування, орієнтація на вимоги користувача, можливість застосування на ранніх стадіях проєкту. Однак недоліками при цьому виступають потреба у кваліфікованому аналізі вимог, наявність суб’єктивності при оцінці складності функцій, більш придатний для інформаційних систем, ніж для систем з великою алгоритмічною складовою.

Хід роботи

1. Обрати приклад розробки програмної системи (визначити користувачів та функціонал).
2. Ідентифікація функціональних елементів системи.
3. Розрахунок нефакторованих функціональних точок (UFP).
4. Оцінка впливових факторів та розрахунок TDI.
5. Розрахунок скоригованих функціональних точок (AFP).
5. розрахунок трудомісткості та вартості розробки програмної системи.

Питання для самоконтролю

1. На чому ґрунтується метод функціональнихточок?
2. Переваги та недоліки методу функціональних точок?
3. Які фактори визначають складність проєкту?
4. Що таке трудомісткість розробки програмної системи?

5. Які показники включено у розрахунок вартості розробки проєкту?

6. Методика оцінки ризиків?

ПРАКТИЧНА РОБОТА № 5

ОЦІНКА ВАРТОСТІ ІТ-ПРОЄКТІВ НА ОСНОВІ МОДЕЛЕЙ COCOMO

Мета: навчитися оцінювати вартість програмного забезпечення із застосуванням моделі COCOMO та COCOMO II

Завдання

1. Проаналізувати модель оцінювання трудомісткості розробки проєктів COCOMO.

2. Проаналізувати модель оцінки вартості ІТ-проєктів COCOMO II.

3. Проаналізувати параметри оцінки вартості розробки ПЗ.

4. Здійснити відповідні розрахунки для отримання показників вартості ІТ-проєкту.

Теоретичні відомості

5.1 Моделі оцінювання трудомісткості розробки ПЗ. Виходячи з процесу проектування ПЗ, існуючі моделі оцінювання трудомісткості програмного забезпечення поділяють на дві групи: алгоритмічні, що засновані на підрахунку кількісних характеристик програми у вигляді числа операторів або функціональних точок, та неалгоритмічні, що використовують визначені схеми або принципи [9].

До відомих неалгоритмічних моделей відносяться експертні оцінки. Експертні оцінки в першу чергу застосовуються для проектів, що вирішують інноваційні завдання або засновані на новітніх технологіях та процесах. Часто експертами виступають безпосередньо замовники та розробники, що мають певний досвід. На основі оцінок окремих експертів у відповідності до існуючих методик формується інтегрована консенсусна оцінка.

До загальних недоліків методів експертного оцінювання відносять моменти, пов’язані з використанням складного і часом непрогнозованого «людського» фактору. Важко оцінити ступінь повноти і достовірності інформації, поданої експертами. Немає цілковитої впевненості, що експерти виявили дійсно всі основні проблеми і правильно визначили взаємозв'язки між ними. Відсутність у явному виді аналітичного обґрунтування виявлених проблем, складність у перевірці компетенції тієї або іншої особи через неформалізованість критеріїв об’єктивності представлених даних.

Метод оцінки по аналогії є загальноприйнятим для будь-якого оцінювання і вважається різновидом експертної оцінки. Він використовує у якості оцінки емпіричні дані про вже завершені проекти схожого типу.

Алгоритмічні моделі базуються на математичних моделях і постійно удосконалюються з метою підвищення їх точності оцінювання. Найчастіше реалізованими і добре документованими моделями є модель Путнем (ступенева, аналітична) і модель COCOMO (ступенева, емпірична).

Модель Путнем (SLIM) [10] є найбільш поширеною моделлю аналітичної групи. Вона розроблена для проєктів об'ємом більше 70 000 рядків коду. Модель ґрунтується на твердженні, що витрати на розробку програмного забезпечення розподіляються згідно кривих Нордена-Рейлі, які є графіками функції, що представляє розподіл робочої сили у часі [3]. При цьому необхідно звернути увагу на те, що спочатку дослідження Нордена базувалися не на теоретичній основі, а на спостереженнях за проєктами, в основному не пов'язаними з програмним забезпеченням (машинобудування, будівництво). Тому немає наукового підтвердження того факту, що програмні проєкти вимагають такого ж розподілу робочої сили, а навпаки, часто кількість людино-годин, необхідних для ІТ-проєкту, може різко змінитися й проведена оцінка вартості стає неадекватною.

5.2 Модель COCOMO (Constructive Cost Model). Найпопулярнішою серед алгоритмічних моделей є сімейство моделей COCOMO (Constructive Cost Model), розроблене у 1981 р. [8, 11]. Модель використовує просту формулу регресії з параметрами, визначеними з даних, зібраних по множині проєктів. COCOMO включає моделі у відповідності до фаз життєвого циклу програмного забезпечення:

- базова (Basic) застосовується на етапі напрацювання специфікацій, оцінка лише за розміром коду;
- розширена (Intermediate) – після визначення конкретних вимог до програмного забезпечення, враховує 15 факторів впливу (рівень досвіду команди, використання інструментів, обмеження часу, складність системи);
- поглиблена (Advanced) застосовується після закінчення проектування, додатково враховує різні етапи життєвого циклу проєкту (аналіз, проєктування, кодування, тестування).

При цьому витрати праці на проєкт визначають у людино-місяцях і залежать від розміру коду програми та від її складності, що визначається різновидом програми:

– відносно проста, однорідна команда розробників;

– проект середньої складності, вимоги можуть змінюватися у процесі розроблення;

– складний (вбудований) проєкт, що має бути реалізований у жорстких рамках заданих вимог.

В моделі COCOMO використовуються три режими, за допомогою яких класифікують складність системи, а також середовище розробки.

Органічний режим (Organic), зазвичай, характеризують параметри: невелика команда по розробці проєкту, необхідні невеликі нововведення, є не жорсткі обмеження і кінцевий термін, а середовище розробки є стабільним.

Напіввбудований режим (Semi-detached) типізується прикладними системами, наприклад, компіляторами, системами баз даних або редакторами. Характеристики: невелика команда по розробці проєкту

середнього розміру, необхідні деякі інновації, помірні обмеження і кінцевий термін, а середовище розробки дещо нестабільне.

Вбудований (Embedded) характеризується режимами реального часу, наприклад, системами контролю повітряного руху, мережами АТМ або воєнними системами. Характеристики: велика команда розробників проєкту, великий об'єм необхідних інновацій, жорсткі обмеження і терміни здачі. Середовище розробки в цьому випадку складається з багатьох складних інтерфейсів, включаючи ті, які поставляються замовникам разом з апаратним забезпеченням.

Модель базується на емпіричних даних про програмні проєкти. Основна ідея полягає в тому, що трудомісткість розробки програмного продукту залежить від його розміру (*KLOC* – тисячі рядків коду) та певних коригуючих коефіцієнтів. Для кожного типу використовуються різні коефіцієнти a, b, c, d (див. таблиця 5.1).

Таблиця 5.1 – Коригуючі коефіцієнти для різних типів проєктів

Тип проєкту	**a**	**b**	**c**	**d**
Органічний (Organic) – невеликі проєкти, досвідчені команди, зрозумілі вимоги	2.4	1.05	2.5	0.38
Напіввбудований (Semi-detached) – середні проєкти, різнорідні команди, частково нові вимоги	3.0	1.12	2.5	0.35
Вбудований (Embedded) – великі складні системи, жорсткі вимоги, апаратні обмеження	3.6	1.20	2.5	0.32

Формула трудомісткості розробки проєкту у людино-місяцях:

$$E = a(KLOC)^b. \quad (5.1)$$

Тривалість виконання проєкту обчислюється за формулою:

$$D = c(E)^d. \quad (5.2)$$

Вартість проєкту обчислюють за формулою:

$$Cost = E \cdot Rate, \quad (5.3)$$

де *Rate* – середня вартість 1 людино-місяця.

У 1995–2000 рр. розроблено удосконалену модель COCOMO II, яка враховує сучасні особливості розробки:

- об'єктно-орієнтоване програмування;
- повторне використання коду;
- компонентний підхід;
- CASE-засоби та автоматизовані інструменти;
- гнучкі методології управління.

Модель COCOMO II складається з кількох підмоделей [11]:

- Application Composition Model – для швидкої оцінки на ранніх стадіях (прототипи, сценарії, об'єктні пункти);
- Early Design Model – для попередньої оцінки, коли відомі загальні характеристики проєкту;

– Post-Architecture Model – використовується після визначення архітектури системи та технологій, враховує понад 20 факторів впливу.

Основні параметри моделі COCOMO II:

– фактори масштабу (Scale Factors – 5), які визначають експоненту зростання зусиль (наприклад, новизна проєкту, узгодженість команди, зрілість процесу);

– множники зусиль (Effort Multipliers >15), які враховують вимоги до надійності, складність, досвід розробників, використання інструментів, обмеження термінів.

При цьому, трудомісткість виконання проєкту обчислюють за формулою:

$$PM = A \cdot Size^{E} \cdot \prod EM_i , \qquad (5.4)$$

де *PM*– трудомісткість (людино-місяці);

A – константа (≈ 2.94);

Size – розмір проєкту (у KLOC або функціональних точках);

E – експонента масштабу, залежна від 5 факторів масштабу (новизна, гнучкість розробки, опрацьованість архітектури / управління ризиками, злагодженість команди, зрілість процесу розробки);

EMi – множники зусиль (effort multipliers), що враховують складність, досвід, інструменти тощо.

Константа *A* – це нормувальний коефіцієнт, який визначає базовий рівень зусиль для одиниці розміру програмного коду при “середніх” умовах проєкту. *A*=2,94 забезпечує найкраще наближення теоретичних розрахунків до реальних даних проєктів. Вона відображає усереднену базову продуктивність розробника для 1 *KLOC* коду.

Експонента масштабу визначає ступінь нелінійного зростання трудомісткості залежно від розміру проєкту і обчислюється за формулою:

$$E = B + 0{,}01 \times \sum_{j=1}^{5} SF_j , \qquad (5.5)$$

де *B* – базова константа (зазвичай 0,91);

SF_j – значення факторів (Scale Factors) у балах (від 0 до 5) (див. таблицю 5.2).

Константа B 0.91 задає мінімальний рівень експоненти – тобто найкращий можливий випадок, коли усі процеси максимально зрілі (усі SF_j = 0). У такій ситуації E=0.91 і зростання зусиль при збільшенні розміру проєкту відбувається майже лінійно (ефективна розробка). Якщо фактори масштабу мають середні оцінки SF_j=3, то E=0.91+0.01×15=1.06, що відповідає нелінійному зростанню зусиль при збільшенні розміру (типова ситуація для реальних проєктів). Для складних або хаотичних проєктів, де всі SF_j = 5, отримаємо E=0.91+0.01×25=1.16, тобто надпропорційне зростання трудомісткості.

Таблиця 5.2 – Фактори масштабу проєкту

Фактор	Дуже низький – 5	Низький – 4	Номінальний – 3	Високий – 2	Дуже високий – 1	Надзвичайно високий – 0
PREC (Precedentedness – новизна проєкту)	Практично немає подібного досвіду; проєкт повністю новий	Обмежений досвід з подібними системами	Деякий досвід у схожих проєктах	Значний досвід з аналогічними продуктами	Команда багаторазово виконувала подібні проєкти	Повна повторюваність, стандартний процес
FLEX (Development Flexibility – гнучкість розробки)	Надзвичайно жорсткі вимоги; майже без варіантів	Жорсткі вимоги, обмежена свобода	Помірна гнучкість у прийнятті рішень	Більшість вимог узгоджується з командою	Висока гнучкість, замовник довіряє команді	Повна свобода у виборі рішень
RESL (Architecture/Risk Resolution – опрацьованість архітектури та ризиків)	Архітектура не визначена, ризики не аналізуються	Часткове розуміння архітектури	Архітектура окреслена, ризики частково враховані	Добре розроблена архітектура, ризики керовані	Архітектура стабільна, ризики мінімальні	Повністю формалізована архітектура та план ризиків
TEAM (Team Cohesion – злагодженість команди)	Конфлікти в команді, відсутність спільних цілей	Слабка комунікація, різні погляди	Нормальна співпраця, але не повна єдність	Добра взаємодія та довіра в команді	Команда повністю узгоджена, сильна співпраця	Ідеальна злагодженість, відмінна комунікація
PMAT (Process Maturity –зрілість процесу розробки)	Початковий, неструктурований процес (CMMI Level 1)	Частково формалізований (CMMI Level 2)	Повторювані процеси (CMMI Level 3)	Керовані процеси (CMMI Level 4)	Оптимізовані процеси (CMMI Level 5)	Повна зрілість і стандартизація

Основні групи множників зусиль у COCOMO II.

1. Характеристики продукту (Product factors):

– RELY (Required Software Reliability) – вимоги до надійності;

– DATA (Database Size) – розмір бази даних у відношенні до коду;

– CPLX (Product Complexity) – складність продукту (алгоритми, інтерфейси, зв’язки).

2. Характеристики платформи (Platform factors):

– TIME (Execution Time Constraint) – обмеження часу виконання;

– STOR (Main Storage Constraint) – обмеження об’єму пам’яті;

– PVOL (Platform Volatility) – мінливість платформи (частота змін апаратного / програмного середовища).

3. Характеристики персоналу (Personnel factors):

– ACAP (Analyst Capability) – кваліфікація аналітиків;

– PCAP (Programmer Capability) – кваліфікація програмістів;

– PCON (Personnel Continuity) – стабільність складу команди;

– APEX (Applications Experience) – досвід у предметній області;

– PLEX (Platform Experience) – досвід роботи з обраною платформою;

– LTEX (Language and Tool Experience) – досвід використання мов програмування й інструментів.

4. Характеристики проєкту (Project factors):

– TOOL (Use of Software Tools) – використання інструментів підтримки (CASE, IDE, системи контролю версій);

– SITE (Multisite Development) – розподіленість команди, якість комунікації;

– SCED (Required Development Schedule) – стислий графік розробки (наскільки стиснуті терміни у порівнянні з номінальними).

Кожен множник має рівень оцінки (дуже низький, низький, номінальний, високий, дуже високий, надзвичайно високий). Для кожного рівня задається числовий коефіцієнт (див. таблиця 5.3).

Таблиця 5.3 – Коефіцієнти множників зусиль

Категорія	Параметр	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Надзвичайно високий
Характеристики продукту	RELY (надійність)	0.82	0.92	1.00	1.10	1.26	–
	DATA (розмір БД)	–	0.90	1.00	1.14	1.28	–
	CPLX (складність продукту)	0.73	0.87	1.00	1.17	1.34	1.74
Характеристики платформи	TIME (обмеження часу виконання)	–	–	1.00	1.11	1.30	1.66
	STOR (обмеження пам’яті)	–	–	1.00	1.06	1.21	1.56
	PVOL (мінливість платформи)	–	0.87	1.00	1.15	1.30	–
Характе	ACAP	1.42	1.19	1.00	0.85	0.71	–

ристики персоналу	(кваліфікація аналітиків)						
	PCAP (кваліфікація програмістів)	1.34	1.15	1.00	0.88	0.76	–
	PCON (стабільність команди)	1.29	1.12	1.00	0.90	0.81	–
	APEX (досвід у предметній області)	1.22	1.10	1.00	0.88	0.81	–
	PLEX (досвід з платформою)	1.19	1.09	1.00	0.91	0.85	–
	LTEX (досвід з мовами та інструментами)	1.20	1.09	1.00	0.91	0.84	–
Характеристики проєкту	TOOL (використання інструментів)	1.17	1.09	1.00	0.90	0.78	–
	SITE (розподіленість команди)	1.22	1.09	1.00	0.93	0.86	0.80
	SCED (графік розробки)	1.43	1.14	1.00	1.00	1.00	–

Значення < 1.0 → зменшує зусилля (сприятливі умови), значення = 1.0 → номінальний вплив, значення >1.0 → збільшує зусилля (ускладнює проєкт).

Таким чином, модель COCOMO II забезпечує більш гнучкий і реалістичний підхід до оцінки вартості ІТ-проєктів, ніж класична COCOMO.

Хід роботи

1. Обрати приклад розробки ІТ-проєкту.
2. Ознайомитися з методикою оцінки вартості проєкту за базовою моделлю COCOMO та COCOMO II.
3. Визначити тип проєкту (органічний, напіввбудований, вбудований).

4. Вибрати вхідні дані:

– оцінку розміру проєкту у KLOC;
– середню ставку розробника (Rate);
– значення факторів для COCOMO II.

5. Виконати обчислення:

– трудомісткість і тривалість та вартість розробки проєкту за базовою моделлю COCOMO;
– трудомісткість за моделлю COCOMO II;
– порівняти отримані результати.

Питання для самоконтролю

1. Що таке модель оцінки трудомісткості розробки ПЗ?
2. Алгоритмічні моделі оцінки трудомісткості розробки ПЗ?
3. Характеристики моделі COCOMO?
4 У чому відмінність між моделлю COCOMO та COCOMO II?
5. Що таке множники зусиль у моделі COCOMO II?
6.Чому модель COCOMO II більш точна для сучасних IT-проєктів?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abran A. Function Points Analysis: An Empirical Study of Its Measurement Processes. IEEE Transactions on Software Engineering, 1996. Vol. 22, No 12. P.895-910.
2. Anou A., Subroto I. M. I., Marwanto A. Estimation of Software Size Using Function Point Analysis on Inventory Information Systems. Journal of Telematics and Informatics (JTI). 2021. Vol.9. No.2. P 58-67.
3. Bilous D., Kozlovskyi A. Using function point analysis for professional service and maintenance IT projects: a tailoring approach for enhanced size and effort estimation. XI International Scientific and Practical Conference «Modern science: theoretical and practical view», April 16-17, 2024, Madrid. Spain. P.82-90.
4. Patwa S., Malviya A.K. A Study of Function Point Analysis: Some Suggestive Modifications. Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE), 2019. P. 634-637.
5. Балазюк О.Ю., Сисоева І.Н., Пилявець В.Н. Комплексна оцінка ефективності інвестиційних проектів із розширення інформаційної системи підприємства // Економіка і суспільство. 2018. Вип. 18. С. 851-861.
6. Горбатенко О. Моделі ціноутворення при укладанні угод у сфері IT: веб-сайт. URL: http://www.moris.com.ua/modeli-tsinoutvorennya-pri-ukladanni-ugod-u-sferi-it/ (дата звернення: 2.02.2019).
7. Гороховатський В.О., Дубницький В.Ю., Кобилін А.М., ЛукінВ.О. та ін. Визначення трудомісткості при розробленні програмних комплексів // Системи обробки інформації. 2014. Вип. 2 (118). С. 92-98.

8. Гудзовата О. О., Костенко А. В., Плеша М. І. Оцінка ефективності впровадження ІТ-проектів // Вісник Львівського торговельно-економічного університету. Економічні науки. 2020. № 60. С.54-60.

9. Косенюк Г.В., Розломій І.О. Методологія економічної ефективності управління ІТ-проектом // Наукові записки Львівського університету бізнесу та права. Серія економічна. Серія юридична. 2021. Вип. 31. 47-53.

10. Кузнєцов М.С. Оцінка ефективності інформаційних систем. Навч.посібник. Дніпропетровськ:НМетАУ, 2007.

11. Шантир А.С., Зінченко В.В., Єльченко С.В., Кравчук П.О.Засади удосконалення моделей оцінки якості програмних систем на прикладі моделей COCOMO та Iso 9126/25010 // Наукові записки ДУТ. 2024. №1(5). С. 94-104.

ДОДАТОК А

Кількість логічних рядків коду на одну функціональну точку для мов програмування

Мова програмування	**КП (кількість логічних рядків коду на одну функціональну точку)**
Basic Assembler	320
Autocoder	320
Netron/CAP	296
Macro Assembler	213
C	128
Пакетні файли DOS	128
Basic	107
Макроси Lotus	107
ALGOL	105
COBOL	105
FORTRAN	105
JOVIAL	105
Змішані мови програмування	105
JCL	96
VPF	95
Pascal	91
COBOL (ANSI 85)	91
APS	86
Slogan	81
RPG	80
Modula-2	80
PL/1	80
Паралельний Pascal	80
Fortran 95	71
Mantis	71
Sabretalk	70
Mapper	69
ColdFusion	68
Datastage	67
Ideal	66
Basic (ANSI)	64
FORTH	64
LISP	64
PROLOG	64
Powerhouse	63
Uniface	61
.NET	60
JSP	59
LOGO	58
C#	58
J2EE	57
Розширений LISP	56
RPG III	56

ASP	56
Java	55
JavaScript	54
C++	53
YACC	53
Culprit	51
Natural	51
KML	50
Visual Basic	50
REXX	50
Ada 95	49
PL/SQL	47
CICS	46
SIMULA	46
Taskmate	45
Focus	45
Web Scripts	44
Pacbase	42
Мови баз даних	40
Clipper DB и dBase III	40
Informix	40
Oracle и SYBASE	40
Openroad	39
Access	38
VBScript	38
Advantage	38
PeopleSoft	37
Cool:Gen/IEF	37
DBase IV	36
Мови підтримки прийняття рішень	35
FoxPro 2.5	34
APL	32
Статичні мови (SAS)	32
Maestro	30
DELPHI	29
Стандартні об’єктно-орієнтовані мови	29
Powerbuilder	28
VB.Net	28
OBJECTIVE-C	27
Lotus Script	23
Oracle Developer /2000	23
Smalltalk	21
awk	21
EIFFEL	21
Shell-сценарії (Perl)	21
Стандартні мови 4-го покоління (4GL)	20
OR3 (4GL)	20
Application Builder	20
CORBA	20

Cristal Reports	20
Datatrieve	20
CLIPPER	19
ABAP (SAP)	18
HTML 3.0	15
Siebel Tools	13
SQL	13
Easytrieve+	13
SQL Forms	11
Excel	6
QUATTRO PRO	6
Мови створення піктограм	4