

**Міністерство освіти і науки України**  
**Національний технічний університет України «Київський політехнічний**  
**інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи №6 з дисципліни  
«Економіка ІТ-індустрії та підприємництво»

**„КОНСТРУКТИВНА МОДЕЛЬ ВАРТОСТІ СОСОМО”**

**Виконала:** *ІП-11 Дякунчак Ілона*

**Перевірив:** *Родіонов П.Ю.*

Київ 2024

**Мета:** навчитися використовувати інструменти за моделлю COCOMO для розрахунку економічних показників розробки програмного забезпечення.

### ЗАВДАННЯ

1. Розрахувати трудомісткість розробки програмного застосунку використовуючи за базовою та проміжною моделями COCOMO. Для виконання роботи брати проекти, що містять більше 25000 рядків коду.
2. Проаналізувати програмний застосунок на основі моделі COCOMO II (попередня та детальна оцінка).
3. Дослідити вплив розміру програмного коду (SIZE) на трудомісткість (PM) та час розробки проєкту (TM) для різних моделей COCOMO II.
4. Отримати значення PM та TM по всім моделям для одного й того ж значення параметра SIZE, обравши номінальний (середній) рівень складності проєкту, що має високу ступінь новизни.
5. Обов'язково навести проведені розрахунки з поясненням вибору всіх параметрів. Якщо параметр не використовувався (або дорівнює нулю) – вказати причину невикористання.

### ВИКОНАННЯ

Для виконання практичної роботи я обрала проєкт за посиланням: <https://github.com/google-pay/shop-paydemo>. Застосунок написаний на JS та являє собою інтернет-магазин одягу. Розмір проєкту: 26962 рядків програмного коду.

#### Завдання 1:

*Обчислимо трудомісткість за базовою моделлю:*

З огляду на складність застосунку і досвід, необхідний для розробки такого проєкту, він підпадає під категорію "Semi-Detached" в моделі COCOMO.

$$PM = ai * (SIZE)^{bi} = 3 * (26,962)^{1,12} = 120,1 \text{ люд.} * \text{міс.}$$

Де:

PM (People × Month) – трудомісткість (люд. × міс.),

SIZE – обсяг програмного продукту в тисячах рядків вихідного тексту (KSLOC).

*Обчислимо трудомісткість за проміжною моделлю:*

Вибір значень атрибутів вартості з поясненням – нижче у таблиці 1:

Таблиця 1 – Вибір значень атрибутів вартості

Атрибути вартості, CDk	Рейтинг	Причина
Характеристики продукту		
1. Необхідна надійність ПЗ	Високий (1,15)	Забезпечення надійності транзакцій та обробки платіжних даних.
2. Розмір БД додатка	Середній(1)	Використання стандартних баз даних для збереження даних транзакцій та клієнтів.
3. Складність продукту	Високий (1,15)	Інтеграція з Google Pay API та функціонал кошика додають складності.
Характеристики апаратного забезпечення		
4. Обмеження швидкодії при виконанні програми	Середній(1)	Стандартні вимоги до часу відгуку для веб-застосунку.
5. Обмеження пам'яті	Середній(1)	Використання стандартних засобів оптимізації пам'яті.
6. Нестійкість оточення віртуальної машини	Дуже низький (n/a)	Не потребує віртуальної машини
7. Необхідний час відновлення	Середній(1)	Час відновлення не є критично важливим, але швидкість все ж важлива для забезпечення хорошого користувацького досвіду (наприклад, для безперервності під час транзакцій через Google Pay).

Характеристики персоналу		
8. Аналітичні здібності	Високий (0,86)	Потребуються високі аналітичні здібності для інтеграції платіжних систем і створення користувацького інтерфейсу.
9. Досвід розробки	Середній(1)	Використання поширених технологій, таких як Node.js і Firebase.
10. Здібності до розробки ПЗ	Високий (0,86)	Для цього проєкту потрібна кваліфікована команда для роботи із транзакційними системами.
11. Досвід використання віртуальних машин	Дуже низький (1,21)	Використання віртуальних машин не потрібне.
12. Досвід розробки на мовах програмування	Високий (0,95)	Node.js використовується, тому потрібен досвід роботи з цією мовою програмування.
Характеристики проєкту		
13. Застосування методів розробки ПЗ	Високий (0,91)	Для забезпечення якості та складності проєкту потрібні формальні методи розробки.
14. Використання інструментарію розробки ПЗ	Середній(1)	Стандартний набір інструментів, таких як IDE, GitHub.
15. Вимоги дотримання графіку розробки	Середній(1)	Немає жорстких термінів для завершення проєкту.

Розрахунки було здійснено за наступною формулою:

$$EAF = \prod_{k=1}^{15} CD_k$$

$$EAF = 1,023$$

$$PM = EAF * a * (SIZE)^b = 1,023 * 3 * (26,962)^{1,12} = 122,87 \text{ люд.} * \text{міс.}$$

Де:

EAF (Effort Adjustment Factor) – добуток обраних атрибутів вартості PM (People × Month) – трудомісткість (люд. × міс.);

SIZE – обсяг програмного продукту в тисячах рядків вихідного тексту (KSLOC);

## Завдання 2:

*Попередня оцінка:*

Фактори масштабу у табличному вигляді (табл. 2):

Таблиця 2 – Фактори масштабу

Чинник масштабу	Значення	Причина
PREC (Precedentedness)	Nominal(3,72)	Команда має досвід роботи з подібними застосунками, але не на глибокому рівні.
FLEX (Development Flexibility)	High (2,03)	Деяка гнучкість у процесі, але визначено основні цілі.
RESL (Architecture/Risk Resolution)	Nominal(4,24)	Основні ризики відомі та проаналізовані до 75%.
TEAM (Team Cohesion)	High (2,19)	Колектив переважно скоординований із достатньою взаємодією.
PMAT (Process Maturity)	Low(6,24)	Процеси мають початкову зрілість (СММ рівень 1).

Множники трудомісткості у табличному вигляді (попередня оцінка) наведено у табл. 3:

Таблиця 3 – Множники трудомісткості (попередня оцінка)

Множник трудомісткості	Значення	Причина
PERS (Personnel Capability)	Nominal (1)	Рівень кваліфікації програмістів середній.
PREX (Personnel Experience)	Low(1,22)	Невеликий досвід у застосуванні аналогічних платформ.
RCPX (Product Reliability and Complexity)	Nominal (1)	Складність програми середня, вимоги до документації помірні.
RUSE (Developed for Reusability)	Nominal (1)	Повторне використання коду планується частково.
PDIF (Platform Difficulty)	Nominal (1)	Платформа стабільна з незначними обмеженнями.
FCIL (Facilities)	High(0,87)	Використовуються інтегровані середовища.
SCED (Required Development Schedule)	Nominal (1)	Графік розробки збалансований без надлишкової напруги.

$$E = 0,91 + 0,01 * 18,42 = 1,094$$

$$EAF = 1,06$$

$$PM = EAF * A * (SIZE)^E = 1,06 * 2,94 * (26,962)^{1,094} = 114,5 \text{ люд.} * \text{міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0.2(E-B)} = 1 * 3,67 * (114,5)^{0,28+0,2(1,094-0,91)} =$$

$$16,48$$

Де:

PM (People × Month) – трудомісткість (люд. × міс.);

TM (Time at Month) – час розробки в календарних місяцях;

SIZE – обсяг програмного продукту в тисячах рядків вихідного тексту (KSLOC);

*Детальна оцінка:*

Множники трудомісткості (детальна оцінка) наведено у таблиці 4:

Таблиця 4 – Множники трудомісткості (детальна оцінка)

Множник трудомісткості	Значення	Причина
Analyst Capability (ACAP)	Nominal (1,29)	Припустимо, команда має кваліфікованих аналітиків із помірним досвідом.
Applications Experience (AEXP)	High (0,88)	Проект демонструє досвід у розробці вебзастосунків, включаючи інтеграцію з платіжними системами, такими як Google Pay.
Programmer Capability (PCAP)	High (0,88)	Висока якість коду і використання сучасних практик свідчать про високий рівень програмістів.
Personnel Continuity (PCON)	Nominal (1,12)	Немає інформації про плинність кадрів, тому номінальне значення є безпечним припущенням.
Platform Experience (PEXP)	High (0,91)	Команда має значний досвід у роботі з вебплатформами, необхідними для роботи з Google Pay API.
Language and Tool Experience (LTEX)	High (0,91)	Проект реалізовано на JavaScript , що вказує на досвідченість команди у цих технологіях.
Required Software Reliability (RELY)	High (1,26)	Враховуючи, що система працює з фінансовими транзакціями, надійність є критично важливою.
Database Size (DATA)	Nominal (0,95)	База даних незначна, основні дані зберігаються у стандартних обсягах.
Software Product Complexity (CPLX)	Nominal (0,87)	Проект інтегрує платіжні API та кошик покупок, що додає складності.

Required Reusability (RUSE)	Nominal (0,95)	Елементи коду можуть бути використані для повторного використання в інших інтеграціях.
Documentation Match to Life Cycle Needs (DOCU)	Nominal (0,91)	Відповідна документація полегшує інтеграцію.
Execution Time Constraint (TIME)	Nominal (n/a)	Хоча продуктивність важлива, немає екстремальних обмежень на швидкість виконання.
Main Storage Constraint (STOR)	Nominal (n/a)	Немає великих вимог до пам'яті.
Platform Volatility (PVOL)	Low (n/a)	Node.js є зрілою та стабільною платформою з мінімальними ризиками змін.
Use of Software Tools (TOOL)	Nominal (1,09)	Використовуються стандартні інструменти для веб-розробки.
Multisite Development (SITE)	Nominal (1,09)	Основна команда працює як єдине ціле, без мультисайтової розробки.
Required Development Schedule (SCED)	Nominal (1,14)	Жодного екстремального тиску на строки виконання не передбачено.

$$E = 0,91 + 0,01 * 18,42 = 1,094$$

$$EAF = 1,12$$

$$PM = EAF * A * (SIZE)^E = 1,12 * 2,45 * (26,962)^{1,094} = 100,84 \text{ люд.} * \text{міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0,2(E-B)} = 1 * 3,67 * (100,84)^{0,28+0,2(1,094-0,91)} = 15,82$$

Де:



PM (People × Month) – трудомісткість (люд. × міс.);

TM (Time at Month) – час розробки в календарних місяцях;

SIZE – обсяг програмного продукту в тисячах рядків вихідного тексту (KSLOC);

### **Завдання 3:**

*Дослідити вплив розміру програмного коду (SIZE) на трудомісткість (PM) та час розробки проєкту (TM) для різних моделей COCOMO II:*

Оскільки досліджуваний проєкт мав трохи більше 25 000 рядків програмного коду, зробимо розрахунки для розмірів 50 000, 75 000 та 100 000:

Для цих розрахунків:

$$E = 0,91 + 0,01 * 18,42 = 1,094$$

Для попередньої оцінки: EAF = 1,06

Для детальної оцінки: EAF = 1,12

*Розмір 50 000 рядків програмного коду:*

Попередня оцінка:

$$PM = EAF * A * (SIZE)^E = 1,06 * 2,94 * (50)^{1,094} = 225,07 \text{ люд.} * \text{міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0,2(E-B)} = 1 * 3,67 * (225,07)^{0,28+0,2(1,094-0,91)} = 20,4$$

Детальна оцінка:

$$PM = EAF * A * (SIZE)^E = 1,12 * 2,45 * (50)^{1,094} = 198,18 \text{ люд.} * \text{міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0,2(E-B)} = 1 * 3,67 * (198,18)^{0,28+0,2(1,094-0,91)} = 19,6$$

*Розмір 75 000 рядків програмного коду:*

Попередня оцінка:

$$PM = EAF * A * (SIZE)^E = 1,06 * 2,94 * (75)^{1,094} = 350,7 \text{ люд.} * \text{міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0.2(E-B)} = 1 * 3,67 * (350,7)^{0,28+0,2(1,094-0,91)} = 23,49$$

Детальна оцінка:

$$PM = EAF * A * (SIZE)^E = 1,12 * 2,45 * (26,962)^{1,094} = 198,18 \text{ люд.*міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0.2(E-B)} = 1 * 3,67 * (198,18)^{0,28+0,2(1,094-0,91)} = 19,6$$

*Розмір 100 000 рядків програмного коду:*

Попередня оцінка:

$$PM = EAF * A * (SIZE)^E = 1,06 * 2,94 * (100)^{1,094} = 480,5 \text{ люд.*міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0.2(E-B)} = 1 * 3,67 * (480,5)^{0,28+0,2(1,094-0,91)} = 25,95$$

Детальна оцінка:

$$PM = EAF * A * (SIZE)^E = 1,12 * 2,45 * (100)^{1,094} = 423,04 \text{ люд.*міс.}$$

$$TM = SCED * C * (PM_{NS})^{D+0.2(E-B)} = 1 * 3,67 * (423,04)^{0,28+0,2(1,094-0,91)} = 24,92$$

*Отримати значення РМ та ТМ по всім моделям для одного й того ж значення параметра SIZE, обравши номінальний (середній) рівень складності проєкту, що має високу ступінь новизни.*

Проміжна COCOMO, розмір – 50 000 рядків програмного коду (рис. 1):

<b>Product Attributes</b>	
Required Reliability	1.15 (H )
Database Size	1.00 (N )
Product Complexity	1.00 (N )
<b>Computer Attributes</b>	
Execution Time Constraint	1.00 (N )
Main Storage Constraint	1.00 (N )
Platform Volatility	1.00 (N )
Computer Turnaround Time	1.00 (N )
<b>Personnel Attributes</b>	
Analyst Capability	0.86 (H )
Applications Experience	0.91 (H )
Programmer Capability	0.86 (H )
Platform Experience	0.90 (H )
Programming Language and Tool Experience	0.95 (H )
<b>Project Attributes</b>	
Modern Programming Practices	1.00 (N )
Use of Software Tools	0.91 (H )
Required Development Schedule	1.00 (N )
<b>New (Values are probably wrong)</b>	
Required reusability	1.05 (H )
Documentation match to life-cycle needs	1.10 (H )
Personnel continuity	1.00 (H )
Multisite development	1.00 (N )

Рис. 1. Параметри проєкту

Результат виконаних розрахунків:

COCOMO RESULTS for lab6								
MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person-months)	DURATION, (in months)	STAFFING, (recommended)
semi-detached	2.0866368903880503	1.12	2.5	0.35	50.000	166.837	14.988	11.131
Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm). Note: the decimal separator is a period. The final estimates are determined in the following manner: <b>effort</b> = a*KLOC <sup>b</sup> , in person-months, with KLOC = lines of code, (in thousands), and: <b>staffing</b> = effort/duration								

COCOMO II, розмір – 50 000 рядків програмного коду.

Параметри представлено на рисунку 2:

COCOMO II - Constructive Cost Model

Software Size

Sizing MethodSource Lines of Code

SLOC

% Design Modified

% Code Modified

% Integration Required

Assessment and Assimilation (0% - 8%)

Software Understanding (0% - 50%)

Unfamiliarity (0-1)

New

50000

Reused

0

0

Modified

Software Scale Drivers

Precedentedness

Low

Architecture / Risk Resolution

High

Process Maturity

High

Development Flexibility

High

Team Cohesion

High

Software Cost Drivers

Product

Required Software Reliability

High

Data Base Size

Nominal

Product Complexity

Nominal

Developed for Reusability

High

Documentation Match to Lifecycle Needs

High

Personnel

Analyst Capability

High

Programmer Capability

High

Personnel Continuity

High

Application Experience

High

Platform Experience

High

Language and Toolset Experience

High

Platform

Time Constraint

Nominal

Storage Constraint

Nominal

Platform Volatility

Nominal

Project

Use of Software Tools

High

Multisite Development

Nominal

Required Development Schedule

Nominal

Maintenance

Off

Software Labor Rates

Cost per Person-Month (Dollars)

Calculate

Рисунок 2 – Параметри проєкту, що містить більше 50 000 рядків програмного коду

Результат проведених розрахунків:

---

## Results

### Software Development (Elaboration and Construction)

Effort = 107.8 Person-months

Schedule = 15.7 Months

Cost = \$0

Total Equivalent Size = 50000 SLOC

Effort Adjustment Factor (EAF) = 0.58

## ВИСНОВОК

У ході виконання практичної роботи було проведено дослідження трудомісткості розробки програмного продукту за допомогою моделей COCOMO та COCOMO II. Аналіз включав оцінку базової та проміжної моделей COCOMO, а також детальну оцінку за COCOMO II. У процесі дослідження було визначено вплив факторів масштабу (PREC, FLEX, RESL, TEAM, PMAT) на складність та тривалість проєкту, оцінено множники трудомісткості для різних аспектів проєкту, таких як кваліфікація персоналу, досвід роботи з платформою, складність продукту та необхідна повторна використаність коду. На основі отриманих даних було встановлено, що розмір програмного коду (SIZE) є основним чинником, що визначає трудомісткість розробки (PM). Зі збільшенням SIZE трудомісткість зростає в нелінійній залежності через вплив множників і факторів масштабу. Час розробки (TM) також значно залежить від трудомісткості (PM), але ця залежність моделюється з урахуванням типу проєкту та специфіки його реалізації.

Дослідження підтвердило, що використання моделей COCOMO II є ефективним підходом для планування ресурсів, оцінки вартості та тривалості розробки програмних продуктів. Отримані результати демонструють важливість обґрунтованої оцінки факторів масштабу і

множників трудомісткості для підвищення точності прогнозування та управління процесом розробки.