



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

Практична робота №6

Економіка ІТ-індустрії

Тема: Конструктивна модель вартості COSOMO

Виконав

студент групи ІП-11:

Панченко С. В.

Перевірив:

Родіонов П. В.

Київ 2024

ЗМІСТ

1 Мета.....	3
2 Завдання.....	4
3 Виконання.....	5
3.1 Вибір проекту.....	5
3.2 Базова модель COCOMO.....	5
3.2.1 Визначення SIZE.....	5
3.2.2 Тип проекту.....	6
3.2.3 Розрахунки вихідних значень.....	7
3.3 Проміжна модель COCOMO.....	7
3.3.1 Фактори витрат.....	7
3.3.2 Розрахунки вихідних значень.....	11
3.4 COCOMO 2.....	12
3.4.1 Фактори масштабу.....	12
3.4.2 Попередня оцінка.....	16
3.4.2.1 Визначення множників трудомісткості.....	16
3.4.2.2 Розрахунок.....	18
3.4.3 Детальна оцінка.....	19
3.4.3.1 Визначення множників трудомісткості.....	20
3.4.3.2 Розрахунок.....	24
3.5 Залежність ТМ та РМ від SIZE для моделей COCOMO 2.....	24
3.6 РМ та ТМ для всіх моделей.....	29
3.6.1 Базова модель COCOMO.....	29
3.6.2 Проміжна модель COCOMO.....	30
3.6.3 COCOMO 2.....	30
3.6.3.1 Попередня модель.....	30
3.6.3.2 Детальна модель.....	31
Висновок.....	32

1 МЕТА

Навчитися використовувати інструменти за моделлю СОСОМО для розрахунку економічних показників розробки програмного забезпечення.

2 ЗАВДАННЯ

Розрахувати трудомісткість розробки програмного застосунку використовуючи за базовою та проміжною моделями COCOMO. Для виконання роботи брати проєкти, що містять більше 25000 рядків коду.

Проаналізувати програмний застосунок на основі моделі COCOMO II (попередня та детальна оцінка).

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

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.

Обов'язково навести проведені розрахунки з поясненням вибору всіх параметрів. Якщо параметр не використовувався (або дорівнює нулю) – вказати причину невикористання.

3 ВИКОНАННЯ

3.1 Вибір проекту

Для аналізу я обрав `cppfront`. `cppfront` — це експериментальний проєкт, створений для вивчення нових підходів до програмування на C++. Його метою є розробка сучасного фронтенду, який дозволяє писати код на більш спрощеній і високорівневій версії C++, який потім перекладається у стандартний C++ для компіляції. Цей проєкт досліджує можливості покращення продуктивності та зручності використання мови.

3.2 Базова модель COCOMO

Базова модель COCOMO дозволяє оцінити трудомісткість розробки програмного забезпечення (PM), час розробки (TM), середню чисельність персоналу (SS) та продуктивність (P) на основі розміру програмного коду (SIZE) і типу проєкту.

3.2.1 Визначення SIZE

За допомогою скрипта `count_lines.py` підрахуємо кількість рядків коду в директоріях `source` та `include`:

```
1 import os
2 from pathlib import Path
3
4 SCRIPT_PATH = Path(os.path.abspath(__file__))
5 SCRIPT_DIR_PATH = SCRIPT_PATH.parent
6 EXTERNAL_PATH = SCRIPT_DIR_PATH / 'external'
7 SUFFIXES: list[str] = ['.h', '.hpp', '.c', '.cpp', '.cpp2', '.h2', '.cxx', '.sh']
8
9 def main() -> None:
10     total_lines = 0
11     for cppfront_dir in ['source', 'include']:
12         for dir, _, src_file_names in os.walk(EXTERNAL_PATH / 'cppfront' / cppfront_dir):
13             for src_file_name in src_file_names:
```

```

14     if any(src_file_name.endswith('.') + suffix) for suffix in SUFFIXES):
15         src_file_path = Path(dir) / src_file_name
16         with open(src_file_path, 'r') as src_file:
17             total_lines += len(src_file.readlines())
18     print(total_lines)
19
20 if __name__ == '__main__':
21     main()

```

У результаті отримуємо результат:

```

1 (base) sideshowbobgot@localhost:~/university/economics_seventh_semester/lab_6$ python
count_lines.py
2 39463

```

Маємо, кількість рядків коду — це 39463. Переведемо LOC у KLOC за розрахунками (3.1):

$$\text{SIZE} = \frac{39463}{1000} = 39.463 \text{ KSLOC} \quad (3.1)$$

3.2.2 Тип проекту

У таблиці 3.1 описано значення коефіцієнтів базового рівня моделі в залежності від типу проекту.

Таблиця 3.1 Коефіцієнти базового рівня моделі в залежності від типу проекту

Тип проекту	a_1	b_1	c_1	d_1
Розповсюджений	2.4	1.05	2.5	0.38
Напівнезалежний	3	1.12	2.5	0.35
Вбудований	3.6	1.2	2.5	0.32

Проект має напівнезалежний (Semi-Detached) тип. Це обґрунтовано тим, що він розробляється з певною взаємозалежністю між компонентами, але не є повністю вбудованим (зав'язаний на певному обладнанні) чи поширеним (вільно доступний).

3.2.3 Розрахунки вихідних значень

За формулами (3.2), (3.3), (3.4), (3.5) описані обчислення трудомісткості (PM), часу розробки (TM), середньої чисельності персоналу (SS), продуктивності (P) відповідно.

$$PM = a_1 \cdot (SIZE)^{b_1} \quad (3.2)$$

$$TM = c_1 \cdot (PM)^{d_1} \quad (3.3)$$

$$SS = \frac{PM}{TM} \quad (3.4)$$

$$P = \frac{SIZE}{PM} \quad (3.5)$$

За розрахунками (3.6), (3.7), (3.8), (3.9) описані обчислення вихідних параметрів з пункту 3.2.3.

$$PM = 3 \cdot 39.463^{1.12} = 184.02 \text{ люд.} - \text{місяців} \quad (3.6)$$

$$TM = 2.5 \cdot 184.02^{0.35} = 15.51 \text{ календарних місяців} \quad (3.7)$$

$$SS = \frac{184.02}{15.51} = 11.86 \text{ осіб} \quad (3.8)$$

$$P = \frac{39.463}{184.02} = 0.214 \frac{\text{KSLOC}}{\text{люд.-місяць}} \quad (3.9)$$

3.3 Проміжна модель COSOMO

У цьому розділі будуть оцінені вихідні характеристики з допомогою проміжної моделі COSOMO.

3.3.1 Фактори витрат

У таблиці 3.1 наведено коефіцієнти факторів витрат в залежності від рівня.

Таблиця 3.2 Атрибути вартості

Атрибути вартості	Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
Необхідна надійність ПЗ	0.75	0.88	1	1.15	1.4	n/a
Розмір БД	n/a	0.94	1	1.08	1.16	n/a
Складність продукту	0.7	0.85	1	1.15	1.3	1.65
Обмеження швидкодії	n/a	n/a	1	1.11	1.3	1.66
Обмеження пам'яті	n/a	n/a	1	1.06	1.21	1.56
Нестійкість оточення	n/a	0.87	1	1.15	1.3	n/a
Необхідний час відновлення	n/a	0.87	1	1.07	1.15	n/a
Аналітичні здібності	1.46	1.19	1	0.86	0.71	n/a
Досвід розробки	1.29	1.13	1	0.91	0.82	n/a
Здібності до розробки ПЗ	1.42	1.17	1	0.86	0.7	n/a
Досвід використання VM	1.21	1.1	1	0.9	n/a	n/a
Досвід розробки мовами	1.14	1.07	1	0.95	n/a	n/a

Атрибути вартості	Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
Застосування методів розробки	1.24	1.1	1	0.91	0.82	n/a
Використання інструментарію	1.24	1.1	1	0.91	0.83	n/a
Вимоги до графіку	1.23	1.08	1	1.04	1.1	n/a

У таблиці 3.3 описані оцінки атрибутів вартості та обґрунтування.

Таблиця 3.3 Оцінки атрибутів вартості

Атрибут	Значення	Обґрунтування
Необхідна надійність ПЗ (RELY)	1.15	Висока надійність необхідна для компілятора, оскільки помилки можуть критично вплинути на код користувачів
Розмір БД (DATA)	1	Номінальний рівень, оскільки проект не потребує складної бази даних
Складність продукту (CPLX)	1.15	Висока складність через специфіку домену (компілятор, обробка складних синтаксичних конструкцій)

Атрибут	Значення	Обґрунтування
Обмеження швидкодії (TIME)	1.11	Високі вимоги до продуктивності, оскільки компілятор має працювати швидко
Обмеження пам'яті (STOR)	1	Номінальні вимоги до пам'яті, сучасні системи мають достатньо ресурсів
Нестійкість оточення (VIRT)	1	Номінальна стабільність середовища, проект працює на стандартних платформах
Необхідний час відновлення (TURN)	1	Номінальний час відновлення, не критичний для інструменту розробки
Аналітичні здібності (ACAP)	0.86	Потрібні високі аналітичні навички через складність компіляторів
Досвід розробки (AEXP)	0.91	Необхідний високий досвід у розробці компіляторів та C++
Здібності до розробки ПЗ (PCAP)	0.86	Потрібні високі навички програмування через складність проекту
Досвід використання VM (VEXP)	1	Номінальний досвід, не критичний для проекту
Досвід розробки мовами (LEXP)	1	Номінальний рівень, хоча важливе знання C++
Застосування методів розробки (MODP)	0.91	Високий рівень застосування інженерних методів для забезпечення якості

Атрибут	Значення	Обґрунтування
Використання інструментарію (TOOL)	0.91	Активне використання інструментів розробки та тестування
Вимоги до графіку (SCED)	1.04	Помірні обмеження по часу, проект експериментальний

3.3.2 Розрахунки вихідних значень

Обчислимо EAF (Effort Adjustment Factor) за формулою (3.10) та за розрахунком (3.11).

$$EAF = \prod_{k=1}^{15} CD_k \quad (3.10)$$

$$EAF = 1.15 \cdot 1.00 \cdot 1.15 \cdot 1.11 \cdot 1.00 \cdot 1.00 \cdot 1.00 \cdot 0.86 \cdot 0.91 \cdot 0.86 \cdot 1.00 \cdot 1.00 \cdot 0.91 \cdot 0.91 \cdot 1.04 = 0.8509 \quad (3.11)$$

У таблиці 3.4 наведено значення коефіцієнтів проміжного рівня залежно від типу проекту.

Таблиця 3.4 Значення коефіцієнтів проміжного рівня залежно від типу проекту.

Тип проекту	a_1	b_1
Розповсюджений	3.2	1.05
Напівнезалежний	3	1.12
Вбудований	2.8	1.2

Обчислимо трудомісткість за формулою (3.12) та розрахунком (3.13).

$$PM = EAF \cdot a_1 \cdot (SIZE)^{b_1} \quad (3.12)$$

$$PM = 0.8509 \cdot 3.0 \cdot (39.463)^{1.12} = 156.58 \text{ люд.} - \text{місяців} \quad (3.13)$$

За формулами та розрахунками (3.3), (3.4), (3.5), (3.14), (3.15), (3.16) описані обчислення інших вихідних параметрів.

$$TM = 2.5 \cdot (156.58) \cdot 0.35 = 14.66 \text{ календарних місяців} \quad (3.14)$$

$$SS = \frac{156.58}{14.66} = 10.68 \text{ осіб} \quad (3.15)$$

$$P = \frac{39.463}{156.58} = 0.252 \frac{\text{KSLOC}}{\text{люд.-місяць}} \quad (3.16)$$

3.4 COCOMO 2

У цьому розділі буде застосована модель COCOMO 2 для оцінки вихідних параметрів.

3.4.1 Фактори масштабу

У таблиці 3.1 визначені фактори масштабу, у таблиці — значення факторів масштабу, у таблиці 3.7 — обране значення та обґрунтування.

Таблиця 3.5 Фактори масштабу

SFj	Опис	Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
PREC (Precedentedness)	Прецедентність, наявність досвіду аналогічних розробок	продукт і платформа відсутній	продукт і платформа не дуже знайомі	деякий досвід в продукті і платформі присутній	продукт і платформа в основному відомі	продукт і платформа великою мірою знайомі	продукт і платформа повністю знайомі
FLEX (Development Flexibility)	Гнучкість процесу розробки	процес строго детермінований	допускаються деякі компроміси	значна жорсткість процесу	відносна жорсткість процесу	незначна жорсткість процесу	визначені тільки загальні цілі
RESL (Architecture/Risk Resolution)	Архітектура і дозвіл ризиків	ризики відомі // проаналізовані на 20%	ризики відомі // проаналізовані на 40%	ризики відомі // проаналізовані на 60%	ризики відомі // проаналізовані на 75%	ризики відомі // проаналізовані на 90%	ризики дозволені на 100%

TEAM (Team Cohesion)	Спрацьованість команди	формальна взаємодія	важка взаємодія до деякої міри	частіше колективна робота	переважно основному колективна робота	висока міра взаємодії	повна довіра, взаємозаміна і взаємодопомога
PMAT (Process Maturity)	Зрілість процесів	CMM Рівень 1 (нижче середнього)	CMM Рівень 1 (вище середнього)	CMM Рівень 2	CMM Рівень 3	CMM Рівень 4	CMM Рівень 5

Таблиця 3.6 Значення факторів масштабу

Чинник масштабу (SFj)	Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
PREC	6.2	4.96	3.72	2.48	1.24	0
FLEX	5.07	4.05	3.04	2.03	1.01	0
RESL	7.07	5.65	4.24	2.83	1.41	0
TEAM	5.48	4.38	3.29	2.19	1.1	0
PMAT	7.8	6.24	4.68	3.12	1.56	0

Таблиця 3.7 Обрані фактори масштабу

Чинник масштабу (SFj)	Рівень	Значення	Обґрунтування
PREC	Середній	3.72	Частковий досвід у подібних проєктах експериментальний характер.
FLEX	Високий	2.03	Гнучкий підхід до процесу розробки визначені тільки основні цілі.
RESL	Високий	2.83	Аналіз ризиків і архітектури виконаний на рівні 75%..
TEAM	Середній	3.29	Базовий рівень співпраці та розподілу обов'язків у команді.
PMAT	Середній	4.68	Дотримання базових принципів керування розробкою середній рівень зрілості.

3.4.2 Попередня оцінка

Виконаємо попередню оцінку: визначимо множники трудомісткості, а потім — вихідні характеристики.

3.4.2.1 Визначення множників трудомісткості

Для цієї оцінки необхідно оцінити для проєкту рівень семи множників трудомісткості:

- параметри персоналу:
 - PERS (Personnel Capability) – кваліфікація персоналу (Extra Low – аналітики і програмісти мають нижчу кваліфікацію, плінність більше 45%; Extra High – аналітики і програмісти мають вищу кваліфікацію, плінність менше 4%);
 - PREX (Personnel Experience) – досвід персоналу (Extra Low – нове застосування, інструменти і платформа; Extra High – застосування, інструменти і платформа добре відомі);
- параметри продукту:
 - RCPX (Product Reliability and Complexity) – складність і надійність продукту (Extra Low – продукт простий, спеціальних вимог по надійності немає, БД маленька, документація не потрібна; Extra High – продукт дуже складний, вимоги по надійності жорсткі, БД надвелика, документація потрібно в повному обсязі);
 - RUSE (Developed for Reusability) – розробка для повторного використання (Low – не вимагається; Extra High – передбачається повторне використання в інших продуктах);
- параметри платформи:
 - PDIF (Platform Difficulty) – складність платформи розробки (Extra Low – спеціальні обмеження по пам'яті і швидкодії відсутні, платформа стабільна; Extra High – жорсткі обмеження по пам'яті і швидкодії, платформа нестабільна);
- параметри проєкта:

- FCIL (Facilities) – обладнання (Extra Low – інструменти найпростіші, комунікації ускладнені; Extra High – інтегровані засоби підтримки життєвого циклу, інтерактивні мультимедіа комунікації);
- SCED (Required Development Schedule) – необхідний виконання графіка робіт (Very Low – 75% від номінальної тривалості; Very High – 160% від номінальної тривалості).

У таблиці 3.8 описано значення множника трудомісткості від рівня, у таблиці 3.9 — значення обраних коефіцієнтів.

Таблиця 3.8 Значення множників трудомісткості

Множник трудомісткості (EMi)	Екстра низький	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Екстра високий
PERS	2.12	1.62	1.26	1	0.83	0.63	0.5
PREX	1.59	1.33	1.22	1	0.87	0.74	0.62
RCPX	0.49	0.6	0.83	1	1.33	1.91	2.72
RUSE	n/a	n/a	0.95	1	1.07	1.15	1.24
PDIF	n/a	n/a	0.87	1	1.29	1.81	2.61
FCIL	1.43	1.3	1.1	1	0.87	0.73	0.62
SCED	n/a	1.43	1.14	1	1	n/a	n/a

Таблиця 3.9 Значення обраних коефіцієнтів

Множник трудомісткості	Рівень	Значення	Обґрунтування
PERS	Високий	0.83	Команда має високу кваліфікацію низький рівень плинності кадрів.

PREX	Високий	0.87	Команда має добрий досвід роботи з інструментами та платформою.
RCPX	Дуже високий	1.91	Продукт складний існують жорсткі вимоги до надійності.
RUSE	Номінальний	1	Немає особливих вимог до повторного використання продукту.
PDIF	Високий	1.29	Є деякі обмеження щодо продуктивності та стабільності платформи.
FCIL	Високий	0.87	Наявні адекватні інструменти та засоби комунікації.
SCED	Номінальний	1	Звичайні вимоги до виконання графіка робіт.

3.4.2.2 Розрахунок

Обчислимо значення E за формулою (3.17) та розрахунком (3.18).

$$E = B + 0.01 \cdot \sum_{j=1}^5 SF_j, \quad (3.17)$$

де $B = \text{const} = 0.91$.

$$E = 0.91 + 0.01 \cdot (3.72 + 2.03 + 2.83 + 3.29 + 4.68) = 1.0755 \quad (3.18)$$

Обчислимо добуток множників трудомісткості за формулою (3.19) та розрахунком (3.20).

$$EAF = \text{PERS} \cdot \text{PREX} \cdot \text{RCPX} \cdot \text{RUSE} \cdot \text{PDIF} \cdot \text{FCIL} \cdot \text{SCED} \quad (3.19)$$

$$EAF = 0.83 \cdot 0.87 \cdot 1.91 \cdot 1.00 \cdot 1.29 \cdot 0.87 \cdot 1.00 = 1.6796 \quad (3.20)$$

Обчислимо PM за формулою (3.21) та розрахунком (3.22).

$$PM = EAF \cdot A_{\text{попередня оцінка}} \cdot (SIZE)^E, \quad (3.21)$$

де $A_{\text{попередня оцінка}} = 2.94$.

$$PM = 1.6796 \cdot 2.94 \cdot (39.463)^{1.0755} = 241.95 \text{ люд.} \cdot \text{міс.} \quad (3.22)$$

Обчислимо PM_{NS} за формулою (3.23) та розрахунком (3.24).

$$PM_{NS} = \frac{PM}{SCED} \quad (3.23)$$

$$PM_{NS} = \frac{241.95}{1} = 241.95 \quad (3.24)$$

Обчислимо TM за формулою (3.25) та розрахунком (3.26).

$$TM = SCED \cdot C \cdot PM_{NS}^{D+0.2 \cdot (E-B)} \quad (3.25)$$

де $C = 3.67$;

$D = 0.28$.

$$TM = 1.00 \cdot 3.67 \cdot (241.95)^{0.28+0.2 \cdot (1.0755-0.91)} = 17.73 \text{ місяців} \quad (3.26)$$

Обчислимо SS та P за формулами (3.4), (3.5) та розрахунками (3.43), (3.44) відповідно.

$$SS = \frac{241.95}{17.73} \approx 13.65 \text{ осіб} \quad (3.27)$$

$$P = \frac{39.463}{241.95} \approx 0.163 \frac{\text{KSLOC}}{\text{люд.} \cdot \text{міс.}} \quad (3.28)$$

3.4.3 Детальна оцінка

Виконаємо детальну оцінку: визначимо множники трудомісткості, а потім — вихідні характеристики.

3.4.3.1 Визначення множників трудомісткості

У таблиці 3.10 описано значення множника трудомісткості від рівня, у таблиці — значення обраних коефіцієнтів.

Таблиця 3.10 Значення множників трудомісткості

Множник зусиль	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Екстра високий
АСАР (Можливість аналітика)	1.42	1.29	1	0.85	0.71	n/a
АЕХР (Досвід застосування)	1.22	1.1	1	0.88	0.81	n/a
РСАР (Можливості програміста)	1.34	1.15	1	0.88	0.76	n/a
РСОН (Неперервність персоналу)	1.29	1.12	1	0.9	0.81	n/a
РЕХР (Досвід платформи)	1.19	1.09	1	0.91	0.85	n/a
LTEx (Досвід мови та інструментів)	1.2	1.09	1	0.91	0.84	n/a
RELY (Необхідна надійність програмного забезпечення)	0.84	0.92	1	1.1	1.26	n/a
DATA (Розмір бази даних)	n/a	0.23	1	1.14	1.28	n/a
CPLX (Складність програмного продукту)	0.73	0.87	1	1.17	1.34	1.74

RUSE (Необхідна можливість повторного використання)	n/a	0.95	1	1.07	1.15	1.24
DOCU (Відповідність документації)	0.81	0.91	1	1.11	1.23	n/a
TIME (Обмеження часу виконання)	n/a	n/a	1	1.11	1.29	1.63
STOR (Основне обмеження зберігання)	n/a	n/a	1	1.05	1.17	1.46
PVOL (Змінність платформи)	n/a	0.87	1	1.15	1.3	n/a
TOOL (Використання програмних засобів)	1.17	1.09	1	0.9	0.78	n/a
SCED (Необхідний графік розробки)	1.43	1.14	1	1	1	n/a
SITE (Багатосайтовий розвиток)	1.22	1.09	1	0.93	0.86	0.8

Таблиця 3.11 Значення обраних коефіцієнтів

Множник зусиль	Рівень	Значення	Обґрунтування
АСАР (Можливість аналітика)	Високий	0.85	Аналітики мають високу кваліфікацію забезпечуючи якісний аналіз вимог.
АЕХР (Досвід застосування)	Високий	0.88	Команда має досвід у схожих проєктах але не є провідними експертами.

PCAP (Можливості програміста)	Високий	0.88	Програмісти мають достатню кваліфікацію для розробки складних рішень.
PCON (Неперервність персоналу)	Номінальний	1	Немає суттєвої плинності кадрів у команді.
PEXP (Досвід платформи)	Номінальний	1	Досвід роботи з платформою є середнім.
LTEX (Досвід мови та інструментів)	Високий	0.91	Команда добре знайома з мовами програмування та використовуваними інструментами.
RELY (Необхідна надійність програмного забезпечення)	Високий	1.1	Високі вимоги до надійності через важливість продукту.
DATA (Розмір бази даних)	Номінальний	1	Продукт не вимагає значного обсягу бази даних.
CPLX (Складність програмного продукту)	Дуже високий	1.34	Продукт складний із великою кількістю модулів та взаємозв'язків.
RUSE (Необхідна можливість повторного використання)	Номінальний	1	Вимоги до повторного використання продукту відсутні.
DOCU (Відповідність документації)	Номінальний	1	Документація відповідає стандартним вимогам проєкту.
TIME (Обмеження часу виконання)	Високий	1.11	Високі вимоги до швидкодії програмного забезпечення.

STOR (Основне обмеження зберігання)	Номінальний	1	Немає жорстких обмежень щодо використання пам'яті.
PVOL (Змінність платформи)	Високий	1.15	Стабільність платформи не є абсолютною допускаються зміни середовища.
TOOL (Використання програмних засобів)	Високий	0.9	Команда використовує інструменти високого рівня автоматизації.
SCED (Необхідний графік розробки)	Номінальний	1	Звичайні вимоги до виконання графіка.
SITE (Багатосайтовий розвиток)	Високий	0.93	Робота виконується на кількох сайтах із добре налагодженими комунікаціями.

3.4.3.2 Розрахунок

Обчислимо добуток множників трудомісткості за формулою (3.29) та розрахунком (3.30).

$$EAF = ACAP \cdot AEXP \cdot PCAP \cdot PCON \cdot PEXP \cdot LTEX \cdot RELY \cdot DATA \cdot CPLX \cdot RUSE \cdot DOCU \cdot TIME \cdot STOR \cdot PVOL \cdot TOOL \cdot SCED \cdot SITE \quad (3.29)$$

$$EAF = 0.85 \cdot 0.88 \cdot 0.88 \cdot 1.00 \cdot 1.00 \cdot 0.91 \cdot 1.10 \cdot 1.00 \cdot 1.34 \cdot 1.00 \cdot 1.00 \cdot 1.11 \cdot 1.00 \cdot 1.15 \cdot 0.90 \cdot 1.00 \cdot 0.93 = 1.0052 \quad (3.30)$$

Обчислимо РМ за формулою (3.31) та розрахунком (3.32).

$$PM = EAF \cdot A_{\text{детальна оцінка}} \cdot (SIZE)^E, \quad (3.31)$$

де $A_{\text{детальна оцінка}} = 2.45$.

$$PM = 1.0052 \cdot 2.45 \cdot (39.463)^{1.0755} = 120.46 \text{ люд.} \cdot \text{міс.} \quad (3.32)$$

Обчислимо PM_{NS} за формулою (3.23) та розрахунком (3.33).

$$PM_{NS} = \frac{120.46}{1} = 120.46 \quad (3.33)$$

Обчислимо ТМ за формулою (3.25) та розрахунком (3.34).

$$TM = 1.00 \cdot 3.67 \cdot 120.46^{0.28 + 0.2 \cdot (1.0755 - 0.91)} = 14.09 \text{ місяців} \quad (3.34)$$

Обчислимо SS та P за формулами (3.4), (3.5) та розрахунками Error: Reference source not found, Error: Reference source not found відповідно.

$$P = \frac{120.46}{14.09} \approx 8.55 \text{ осіб} \quad (3.35)$$

$$P = \frac{39.463}{120.46} \approx 0.328 \frac{\text{KSLOC}}{\text{люд.} \cdot \text{міс.}} \quad (3.36)$$

3.5 Залежність ТМ та РМ від SIZE для моделей COCOMO 2

Для опису залежності напишемо скрипт.

```
1 import math
```

```
2
```

```
3 SCALE_FACTORS = {
```

```

4  "PREC": 3.72, # Precedentedness
5  "FLEX": 2.03, # Development Flexibility
6  "RESL": 2.83, # Architecture/Risk Resolution
7  "TEAM": 3.29, # Team Cohesion
8  "PMAT": 4.68 # Process Maturity
9  }
10
11 # Значення множників трудомісткості для детальної оцінки
12 DETAILED_MULTIPLIERS = {
13     "ACAP": 0.85, # Analyst Capability
14     "AEXP": 0.88, # Applications Experience
15     "PCAP": 0.88, # Programmer Capability
16     "PCON": 1.00, # Personnel Continuity
17     "PEXP": 1.00, # Platform Experience
18     "LTEX": 0.91, # Language and Tool Experience
19     "RELY": 1.10, # Required Software Reliability
20     "DATA": 1.00, # Database Size
21     "CPLX": 1.34, # Product Complexity
22     "RUSE": 1.00, # Required Reusability
23     "DOCU": 1.00, # Documentation Match to Lifecycle Needs
24     "TIME": 1.11, # Execution Time Constraint
25     "STOR": 1.00, # Main Storage Constraint
26     "PVOL": 1.15, # Platform Volatility
27     "TOOL": 0.90, # Use of Software Tools
28     "SITE": 0.93, # Multisite Development
29     "SCED": 1.00 # Required Development Schedule
30 }
31
32 PRELIMINARY_MULTIPLIERS = {
33     "PERS": 0.83, # Personnel Capability
34     "PREX": 0.87, # Personnel Experience
35     "RCPX": 1.91, # Product Reliability and Complexity
36     "RUSE": 1.00, # Required Reusability
37     "PDIF": 1.29, # Platform Difficulty

```

```

38  "FCIL": 0.87, # Facilities
39  "SCED": 1.00 # Required Development Schedule
40 }
41
42 B = 0.91
43 C = 3.67
44 D = 0.28
45
46 def calculate_estimates(
47     size_kloc: float,
48     em_values: dict[str, float],
49     detailed: bool
50 ) -> tuple[float, float]:
51     A = 2.45 if detailed else 2.94
52     sf_sum = sum(SCALE_FACTORS.values())
53     E = B + 0.01 * sf_sum
54     eaf = math.prod(em_values.values())
55     pm = eaf * A * (size_kloc ** E)
56     tm = C * (pm ** (D + 0.2 * (E - B)))
57     return pm, tm
58
59 from typing import Iterable
60 import matplotlib.pyplot as plt
61
62 def plot_graphs(
63     preliminary: Iterable[float],
64     detailed: Iterable[float],
65     klocs: Iterable[float],
66     y_label: str
67 ) -> None:
68     plt.plot(klocs, preliminary, label=f'preliminary')
69     plt.plot(klocs, detailed, label=f'detailed')
70     plt.title(f'{y_label} vs. KLOC')
71     plt.xlabel('KLOC')

```

```

72     plt.ylabel(y_label)
73     plt.legend(title='Model type', loc='upper right')
74     plt.grid(True)
75     plt.savefig(fname=f'{y_label}.svg', format='svg')
76     plt.show()
77
78 import numpy as np
79 from array import array
80
81 klocs = np.arange(0, 1000)
82 preliminary_pm_array: array[float] = array('d')
83 preliminary_tm_array: array[float] = array('d')
84 detailed_pm_array: array[float] = array('d')
85 detailed_tm_array: array[float] = array('d')
86
87 for kloc in klocs:
88     preliminary_pm, preliminary_tm = calculate_estimates(
89         kloc, PRELIMINARY_MULTIPLIERS, False
90     )
91     detailed_pm, detailed_tm = calculate_estimates(
92         kloc, DETAILED_MULTIPLIERS, False
93     )
94     preliminary_pm_array.append(preliminary_pm)
95     preliminary_tm_array.append(preliminary_tm)
96     detailed_pm_array.append(detailed_pm)
97     detailed_tm_array.append(detailed_tm)
98
99 plot_graphs(preliminary_pm_array, detailed_pm_array, klocs, 'PM')
100 plot_graphs(preliminary_tm_array, detailed_tm_array, klocs, 'TM')

```

На рисунках 3.1, 3.2 зображені залежності РМ та ТМ від SIZE відповідно.

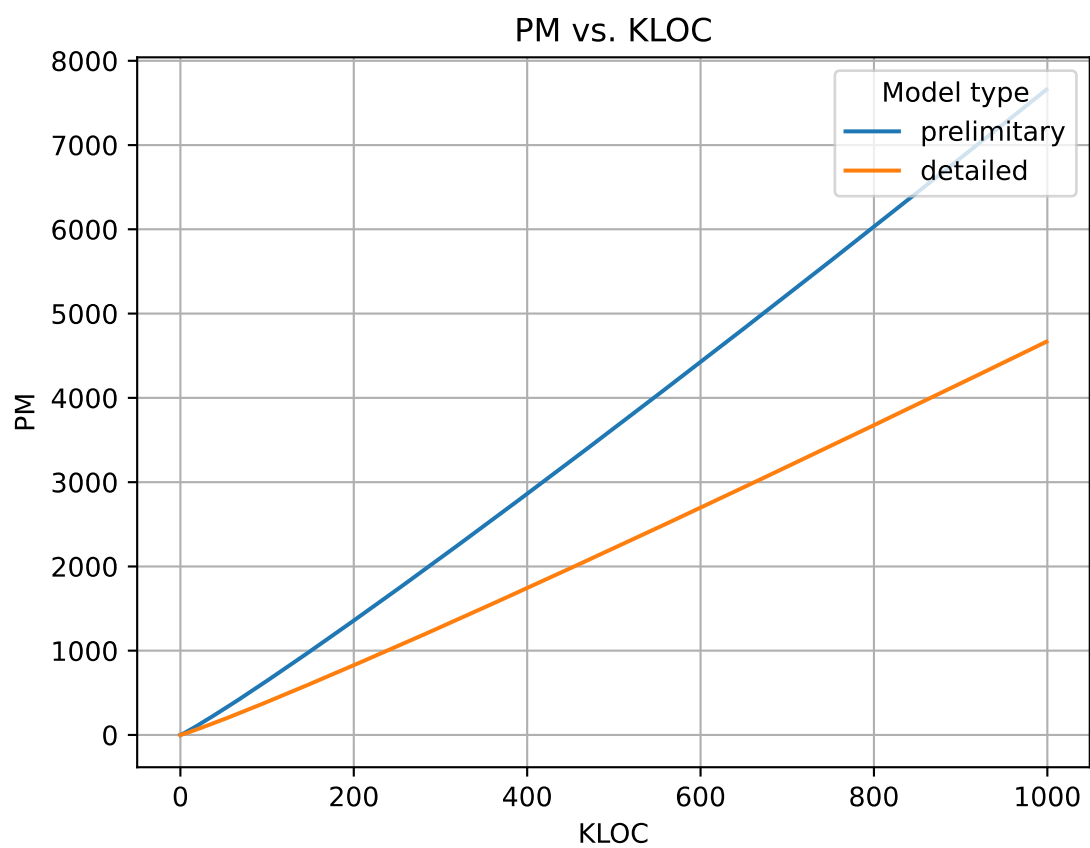


Рисунок 3.1 — Залежність PM від SIZE

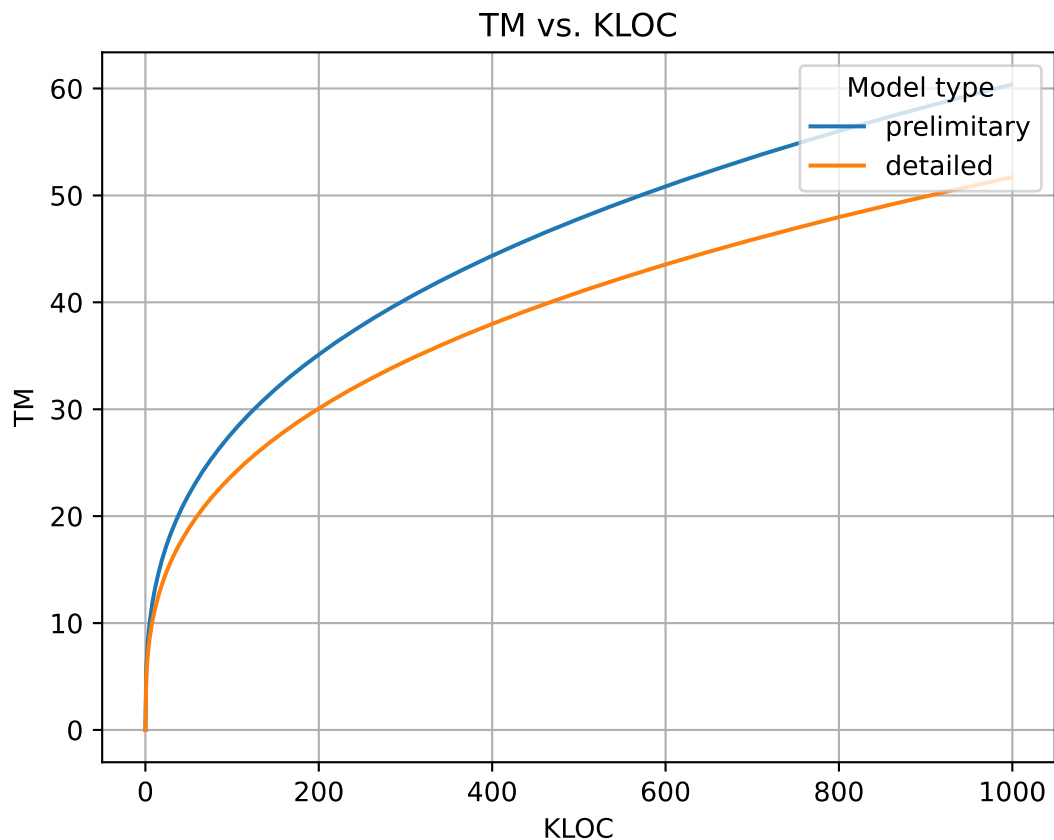


Рисунок 3.2 — Залежність ТМ від SIZE

3.6 РМ та ТМ для всіх моделей

Отримаємо значення РМ та ТМ однакового параметра SIZE, обравши номінальний рівень складності проєкту, що має високу ступінь новизни.

3.6.1 Базова модель COSOMO

Оскільки базова модель двопараметрична, то значення можна взяти з пункту 3.2.3 та записати у таблиці 3.12.

Таблиця 3.12 Вихідні значення

Назва вихідного параметра	Значення
РМ	184.02 люд.-місяців
ТМ	15.51 місяців

3.6.2 Проміжна модель COSOMO

Оскільки усі атрибути вартості мають середній рівень, тобто дорівнюють 1, тоді EAF, який є добутком, теж 1.

Обчислимо трудомісткість та час розробки за формулами (3.12), (3.3) та розрахунками (3.37), (3.38).

$$PM = 1 \cdot 3.0 \cdot (39.463)^{1.12} = 184.02 \text{ люд.} - \text{місяців} \quad (3.37)$$

$$TM = 2.5 \cdot (184.02) \cdot 0.35 = 15.51 \text{ календарних місяців} \quad (3.38)$$

3.6.3 COSOMO 2

Аналогічно до пункту 3.6.2 EAF дорівнює 1. У таблиці 3.13 описані середні значення факторів масштабу.

Таблиця 3.13 Середні значення факторів масштабу

Назва	Значення
PREC	3.72
FLEX	3.04
RESL	4.24
TEAM	3.29
PMAT	4.68

Обчислимо значення E за формулою та розрахунком (3.39).

$$E = 0.91 + 0.01 \times (3.72 + 3.04 + 4.24 + 3.29 + 4.68) = 1.0997 \quad (3.39)$$

3.6.3.1 Попередня модель

Обчислимо PM за формулою (3.21) та розрахунком (3.40).

$$PM = 1 \cdot 2.94 \cdot (39.463)^{1.0997} = 127.85 \text{ люд.} \cdot \text{міс.} \quad (3.40)$$

Обчислимо PM_{NS} за формулою (3.23) та розрахунком (3.41).

$$PM_{NS} = \frac{127.85}{1} = 127.85 \quad (3.41)$$

Обчислимо ТМ за формулою (3.25) та розрахунком (3.42).

$$TM = 1 \cdot 3.67 \cdot 127.85^{0.28 + 0.2 \cdot (1.0997 - 0.91)} = 14.31 \text{ місяців} \quad (3.42)$$

3.6.3.2 Детальна модель

Обчислимо РМ за формулою (3.21) та розрахунком (3.43).

$$PM = 1.00 \cdot 2.45 \cdot 39.463^{1.0997} = 106.54 \text{ люд.} \cdot \text{міс.} \quad (3.43)$$

Обчислимо PM_{NS} за формулою (3.23) та розрахунком (3.44).

$$PM_{NS} = \frac{106.54}{1} = 106.54 \quad (3.44)$$

Обчислимо ТМ за формулою (3.25) та розрахунком (3.45).

$$TM = 1 \cdot 3.67 \cdot 106.54^{0.28 + 0.2 \cdot (1.0997 - 0.91)} = 13.38 \text{ місяців} \quad (3.45)$$

ВИСНОВОК

Порівнюючи отримані результати з реальним проектом, можна відзначити, що розрахункові оцінки достатньо точно відображають сумарний час розробки. Хоча проект знаходиться в розробці більше 5 років (що перевищує розрахункові терміни), це пояснюється тим, що він є некомерційним і розробляється паралельно з основною роботою учасників. При цьому фактична кількість контриб'юторів (34 особи) перевищує розрахункову, але через нерівномірність їх участі в проекті та різну інтенсивність роботи, ефективний розмір команди ближчий до розрахованого в моделях.

Підтверджено, що проміжна модель дає більш точні результати за рахунок врахування додаткових атрибутів проекту, на відміну від базової моделі, яка використовує лише розмір коду.

Детальна оцінка виявилася більш оптимістичною порівняно з попередньою через врахування більшої кількості факторів (17 проти 7) та їх детальнішого впливу на проект.

Графіки демонструють нелінійну залежність трудомісткості та часу розробки від розміру коду. При цьому збільшення розміру коду призводить до непропорційного зростання трудовитрат та часу розробки.

Це демонструє, що моделі СОСОМО дають адекватні оцінки трудовитрат для повноцінної розробки, але потребують коригування при застосуванні до проектів з нестандартним режимом розробки.