



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.

Перевірів:

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
3.4 Детальна оцінка.....	11
Висновок.....	21

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`:

```
import os
from pathlib import Path

SCRIPT_PATH = Path(os.path.abspath(__file__))
SCRIPT_DIR_PATH = SCRIPT_PATH.parent
EXTERNAL_PATH = SCRIPT_DIR_PATH / 'external'
SUFFIXES: list[str] = ['h', 'hpp', 'c', 'cpp', 'cpp2', 'h2', 'cxx', 'sh']

def main() -> None:
    total_lines = 0
    for cppfront_dir in ['source', 'include']:
        for dir, _, src_file_names in os.walk(EXTERNAL_PATH / 'cppfront' / cppfront_dir):
            for src_file_name in src_file_names:
```

```

if any(src_file_name.endswith('.') + suffix) for suffix in SUFFIXES):
    src_file_path = Path(dir) / src_file_name
    with open(src_file_path, 'r') as src_file:
        total_lines += len(src_file.readlines())
print(total_lines)

if __name__ == '__main__':
    main()

```

У результаті отримуємо результат:

```

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

```

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

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

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 розрахунками (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 Детальна оцінка

Виставимо параметри, наведені у таблицях 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11.

Таблиця 3.5 Software Size (Розмір програмного забезпечення)

Параметр	Значення	Обґрунтування
SLOC	39463	Загальна кількість рядків коду в директоріях source та include
New Code	100%	Експериментальний новий проєкт
Reused Code	0%	Не застосовується - новий проєкт
Modified Code	0%	Не застосовується - новий проєкт

Параметр	Значення	Обґрунтування
Software Understanding	30%	Середньо-високий, оскільки це компілятор C++
Unfamiliarity	0.6	Високий, через експериментальний характер

Таблиця 3.6 Scale Drivers (Драйвери масштабу)

Драйвер	Значення	Обґрунтування
Precedentedness	High	Існують подібні компілятори C++
Architecture/Risk Resolution	High	Добре спроектований Гербом Саттером
Team Cohesion	High	Досвідчена основна команда
Process Maturity	Nominal	Стандартна розробка з відкритим кодом
Development Flexibility	Nominal	Стандартна гнучкість

Таблиця 3.7 Cost Drivers - Product (Фактори вартості - Продукт)

Фактор	Значення	Обґрунтування
Required Software Reliability	High	Критично для компілятора
Database Size	Low	Немає складних баз даних
Product Complexity	Very High	Складність компілятора
Developed for Reusability	High	Проект з відкритим кодом
Documentation Match	Nominal	Стандартна документація

Таблиця 3.8 Cost Drivers - Personnel (Фактори вартості - Персонал)

Фактор	Значення	Обґрунтування
Analyst Capability	Very High	Досвідчена команда
Programmer Capability	Very High	Досвідчена команда
Personnel Continuity	Nominal	Природа відкритого коду
Application Experience	High	Експертиза в C++
Platform Experience	Very High	Глибока експертиза

Фактор	Значення	Обґрунтування
Language/Toolset Experience	Very High	Глибока експертиза

Таблиця 3.9 Cost Drivers - Platform (Фактори вартості - Платформа)

Фактор	Значення	Обґрунтування
Time Constraint	Nominal	Немає жорстких термінів
Storage Constraint	Nominal	Стандартні обмеження
Platform Volatility	Nominal	Стандартна волатильність

Таблиця 3.10 Cost Drivers - Project (Фактори вартості - Проект)

Фактор	Значення	Обґрунтування
Use of Software Tools	Very High	Сучасні інструменти
Multisite Development	High	Розподілена команда
Required Development Schedule	Nominal	Стандартний графік

Таблиця 3.11 Додаткові налаштування

Параметр	Значення	Обґрунтування
Maintenance Mode	Off	Оцінка початкової розробки
Ціна на місяць (Доллари)	0	Проект не є комерційним

На рисунку 3.1 можемо розглянути виставлені параметри.

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

39463

Reused

0

0

0

0

0

Modified

0

0

0

0

0

30

0.6

Software Scale Drivers

Precedentedness

High

Architecture / Risk Resolution

High

Process Maturity

Nominal

Development Flexibility

Nominal

Team Cohesion

High

Software Cost Drivers

Product

Required Software Reliability

High

Analyst Capability

Very High

Time Constraint

Nominal

Data Base Size

Low

Programmer Capability

Very High

Storage Constraint

Nominal

Product Complexity

Very High

Personnel Continuity

Nominal

Platform Volatility

Nominal

Developed for Reusability

High

Application Experience

High

Documentation Match to Lifecycle Needs

Very High

Software Tools

Very High

Development

High

Development Schedule

Nominal

Effort = 50.9 Person-months

Schedule = 12.4 Months

Cost = \$0

Maintenance

Off

Total Equivalent Size = 39463 SLOC

Effort Adjustment Factor (EAF) = 0.35

На рисунку 3.2 можна розглянути результати.

Acquisition Phase Distribution				
Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	3.1	1.6	2.0	\$0
Elaboration	12.2	4.7	2.6	\$0
Construction	38.7	7.8	5.0	\$0
Transition	6.1	1.6	3.9	\$0

Software Effort Distribution for RUP/MBASE (Person-Months)				
Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.4	1.5	3.9	0.9
Environment/CM	0.3	1.0	1.9	0.3
Requirements	1.2	2.2	3.1	0.2
Design	0.6	4.4	6.2	0.2
Implementation	0.2	1.6	13.2	1.2
Assessment	0.2	1.2	9.3	1.5
Deployment	0.1	0.4	1.2	1.8

Рисунок 3.2 Результати

Тепер змінимо SIZE та запишемо ТМ(Time to Develop), РМ(Person-Months) у таблицю .

Таблиця 3.12 Залежність РМ та ТМ від SIZE

SIZE	ТМ	РМ
39463	50.9	12.4
23678	29.6	10.5
31570	40.2	11.5
47356	61.8	13.2
55248	72.8	13.9

ВИСНОВОК