

**Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря
Сікорського Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ
з лабораторної роботи №6
з навчальної дисципліни «Економіка ІТ індустрії та підприємництво»**

Тема:

КОНСТРУКТИВНА МОДЕЛЬ ВАРТОСТІ СОСОМО

Виконав:

Студент 4 курсу кафедри ФІОТ,
Навчальної групи ІІІ-11
Головня О. Р.

Перевірив:

Родіонов П. В.

Київ 2024

I. Мета.

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

II. 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).

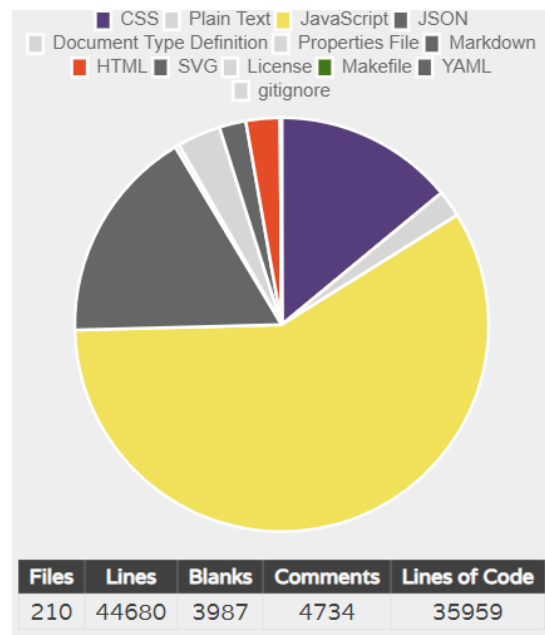
3. Дослідити вплив розміру програмного коду (SIZE) на трудомісткість (PM) та час розробки проєкту (TM) для різних моделей COCOMO II.

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.

III. Результати виконання лабораторної роботи.

Для дослідження було взято проєкт Markdown Here (<https://github.com/adam-p/markdown-here>)



1. Виконаємо обчислення для базової двохпараметричної моделі COCOMO. Модель **Basic COCOMO** – двохпараметрична. Як параметри виступають тип проєкту і обсяг програми (кількість рядків програмного коду).

Рівняння цієї моделі мають вигляд:

$$PM = a_i \times (SIZE)^{b_i},$$

$$TM = c_i \times (PM)^{d_i},$$

$$SS = PM / TM,$$

$$P = SIZE / PM,$$

де PM (People \times Month) – трудомісткість (люд. \times міс.);

TM (Time at Month) – час розробки в календарних місяцях; $SIZE$ – обсяг програмного продукту в тисячах рядків вихідного тексту (KSLOC);

SS – середня чисельність персоналу;

P – продуктивність.

Коефіцієнти a_i, b_i, c_i, d_i вибираються з таблиці (див. таблицю 19).

Таблиця 19 – Значення коефіцієнтів базового рівня моделі COCOMO залежно від типу (моделі) проєкту

Тип проєкту	a_i	b_i	c_i	d_i
Розповсюджений	2,4	1,05	2,5	0,38
Напівнезалежний	3,0	1,12	2,5	0,35
Вбудований	3,6	1,20	2,5	0,32

Оберемо коефіцієнти для напівнезалежного проєкту.

РОЗРАХУНКИ:

$$PM = 3 * (35,9^{1,12}) = 165,5$$

$$TM = 2,5 * (165,5^{0,35}) = 14,94$$

$$SS = 165,5 / 14,94 = 11,07$$

$$P = 35,9 / 165,5 = 0,21$$

Моделі проміжного рівня:

Таблиця 20 – Значення атрибутів вартості залежно від їх рівня

Атрибути вартості, CD _k	Рейтинг					
	Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
Характеристики продукту						
1. Необхідна надійність ПЗ	0,75	0,88	1,00	1,15	1,40	n/a
2. Розмір БД	n/a	0,94	1,00	1,08	1,16	n/a
3. Складність продукту	0,70	0,85	1,00	1,15	1,30	1,65
Характеристики апаратного забезпечення						
4. Обмеження швидкодії при виконанні програми	n/a	n/a	1,00	1,11	1,30	1,66
5. Обмеження пам'яті	n/a	n/a	1,00	1,06	1,21	1,56
6. Нестійкість оточення віртуальної машини	n/a	0,87	1,00	1,15	1,30	n/a
7. Необхідний час відновлення	n/a	0,87	1,00	1,07	1,15	n/a
Характеристики персоналу						
8. Аналітичні здібності	1,46	1,19	1,00	0,86	0,71	n/a
9. Досвід розробки	1,29	1,13	1,00	0,91	0,82	n/a
10. Здібності до розробки ПЗ	1,42	1,17	1,00	0,86	0,70	n/a
11. Досвід використання віртуальних машин	1,21	1,10	1,00	0,90	n/a	n/a
12. Досвід розробки на мовах програмування	1,14	1,07	1,00	0,95	n/a	n/a

1) Необхідна надійність: **Висока**. Оскільки Markdown Here використовується для обробки та відображення тексту у форматі Markdown в електронних листах, надійність є ключовою для коректного відображення форматування.

2) Розмір БД додатка: **Дуже низький**. Бібліотека Markdown Here не використовує

базу даних.

3)Складність продукту: Низька. Markdown Here — це бібліотека, спрощена версія Markdown, і не включає важких функцій.

4)Обмеження швидкодії при виконанні програми: Низьке. Виконання бібліотеки не вимагає значної обчислювальної потужності.

5)Обмеження пам'яті: Низьке. Markdown Here споживає обмежену кількість пам'яті, оскільки він оптимізований для роботи з текстовими даними.

6)Нестійкість оточення віртуальної машини: Низька. Якщо віртуальна машина використовується належним чином, бібліотека має стійкість.

7)Необхідний час відновлення: Низький. Розгортання та відновлення не вимагають значної часової затрати.

8)Аналітичні здібності: Середні. Проект може вимагати аналізу різних варіантів форматування Markdown, але загалом це не є складною задачею.

9)Досвід розробки: Середній. Для внесення внесків у проект потрібен базовий рівень досвіду розробки програмного забезпечення.

10)Здібності до розробки ПЗ: Високі. Проект відкритий для внесення внесків від розробників.

11)Досвід використання віртуальних машин: Середній. Залежить від конкретного розробника, але базовий досвід може бути корисним.

12)Досвід розробки на мовах програмування: Середній. Досвід у JavaScript та HTML корисний для внесення внесків у проект.

13)Застосування методів розробки ПЗ: Високе. Проект слідує стандартам розробки ПЗ та використовує ефективні методи.

14)Використання інструментарію розробки ПЗ: Високе. Використовуються популярні інструменти розробки, такі як Git.

15)Вимоги дотримання графіку розробки: Високі. Проект активно розвивається, і внесення внесків слідує встановленому графіку.

Виконаємо обчислення за формулами:

При визначенні - n/a (not available) - дані відсутні, тобто відповідний рівень не оцінюється

Формула моделі проміжного рівня має вигляд:

$$PM = EAF \times a_i \times (SIZE)^{b_i}$$

де PM – трудомісткість (люд. × міс.);

Size – обсяг програмного продукту в тисячах рядків вихідного тексту

(KSLOC).

ЕАF – (Effort Adjustment Factor) – добуток обраних атрибутів вартості з таблиці (див. табл. 2):

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

Коефіцієнти моделі a_i, b_i вибираються з таблиці 20.

Таблиця 21 – Значення коефіцієнтів проміжного рівня залежно від типу проекту

Тип проекту, i	a_i	b_i
1. Розповсюджений	3,2	1,05
2. Напівнезалежний	3,0	1,12
3. Вбудований	2,8	1,20

Час розробки розраховується за тією ж формулою, що і для базової моделі.

від типу (моделі) проекту				
Тип проекту	a_i	b_i	c_i	d_i
Розповсюджений	2,4	1,05	2,5	0,38
Напівнезалежний	3,0	1,12	2,5	0,35
Вбудований	3,6	1,20	2,5	0,32

$$EAF = 1,15 * 0,85 * 0,87 * 0,87 * 1 * 1 * 0,86 * 1 * 1 * 0,91 * 0,91 * 1,04 = 0,54$$

$$PM = 0,54 * 3 * (35,9 \wedge 1,12) = 89,37$$

$$TM = 2,5 * (89,37 \wedge 0,35) = 12,04$$

Завдання 2

Формула оцінки трудомісткості проєкту в люд. × міс. має вигляд:

$$PM = EAF \times A \times (SIZE)^E$$

де
$$E = B + 0,01 \times \sum_{j=1}^5 SF_j ;$$

$B = 0,91$; $A = 2,94$ для попередньої оцінки;

$A = 2,45$ для детальної оцінки;

SF_j – фактори (чинники) масштабу (Scale Factors);

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

EM_j – множники трудомісткості (Effort Multipliers). $n=7$ – для попередньої оцінки,

$n=17$ – для детальної оцінки;

EAF (Effort Adjustment Factor) – добуток обраних множників трудомісткості:

SFj	Опис	Рівень значущості факторів					
		Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
1. PREC. Precedentedness	Прецедентність, наявність досвіду аналогічних розробок	досвід у продукті і платформі відсутній	продукт і платформа не дуже знайомі	деякий досвід в і продукті і платформі присутній	продукт і платформа в основному відомі	продукт і платформа великою мірою знайомі	продукт і платформа повністю знайомі
2. 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
3. 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%
4. 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
5. 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

- 1 - Високий. Markdown Here базується на вже існуючому стандарті Markdown, тому має високий рівень прецедентності.
- 2 - Середній. Проект дозволяє гнучку розробку, але може мати обмеження через стандарт Markdown.
- 3 - Середній. Ризики можуть виникати при розробці, але вони досить керовані завдяки стандарту Markdown.
- 4 - Високий. Спільнота розробників активно співпрацює, вносить внески та розгортає проект.
- 5 - Середній. Проект розвивається та вдосконалюється, але може потребувати деякого вдосконалення в процесах розробки.
- Коефіцієнти відповідно до обраних значень

Зазначені фактори застосовуються на обох стадіях оцінки проекту.

Числові значення фактора масштабу в залежності від оцінки його рівня, наведені в таблиці 23.

Таблиця 23 – Значення чинника масштабу залежно від оцінки його рівня

Чинник масштабу, SFj	Оцінка рівня чинника (фактора)					
	Very Low	Low	Nominal	High	Very High	Extra High
1. PREC	6,20	4,96	3,72	2,48	1,24	0,00
2. FLEX	5,07	4,05	3,04	2,03	1,01	0,00
3. RESL	7,07	5,65	4,24	2,83	1,41	0,00
4. TEAM	5,48	4,38	3,29	2,19	1,10	0,00
5. PMAT	7,80	6,24	4,68	3,12	1,56	0,00

Таблиця 24 – Значення множників трудомісткості залежно від оцінки їх рівня (Early Design)

№	Множник трудомісткості, EMi	Оцінка рівня множника трудомісткості						
		Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
1	PERS	2,12	1,62	1,26	1,00	0,83	0,63	0,50
2	PREX	1,59	1,33	1,22	1,00	0,87	0,74	0,62
3	RCPX	0,49	0,60	0,83	1,00	1,33	1,91	2,72
4	RUSE	n/a	n/a	0,95	1,00	1,07	1,15	1,24
5	PDIF	n/a	n/a	0,87	1,00	1,29	1,81	2,61
6	FCIL	1,43	1,30	1,10	1,00	0,87	0,73	0,62
7	SCED	n/a	1,43	1,14	1,00	1,00	n/a	n/a

- 1 - Оцінка: Extra High (Висока кваліфікація аналітиків і програмістів, плинність менше 4%).
- 2 - Оцінка: Extra High (Досвід персоналу в застосуванні, інструментах і платформі добре відомий).
- 3 - Оцінка: Nominal (Продукт не є дуже складним, і надійність не вимагає жорстких обмежень).
- 4 - Оцінка: Low (Розробка не передбачає повторне використання в інших продуктах).
- 5 - Оцінка: Extra Low (Спеціальні обмеження по пам'яті і швидкодії відсутні, платформа стабільна).
- 6 - Оцінка: Extra High (Інтегровані засоби підтримки життєвого циклу, інтерактивні мультимедіа комунікації).
- 7 - Оцінка: Nominal (Необхідний виконання графіка робіт збігається з номінальною тривалістю).

Кількість і значення множників трудомісткості відрізняються для різних стадій оцінки проєкту:

1) Стадія попередньої оцінки трудомісткості програмного проєкту (Early Design).

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

– параметри персоналу:

1. PERS (Personnel Capability) – кваліфікація персоналу (Extra Low – аналітики і програмісти мають нижчу кваліфікацію, плинність більше 45%; Extra High – аналітики і програмісти мають вищу кваліфікацію, плинність менше 4%);

2. PREX (Personnel Experience) – досвід персоналу (Extra Low – нове застосування, інструменти і платформа; Extra High – застосування, інструменти і платформа добре відомі);

– параметри продукту:

3. 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);

4. RUSE (Developed for Reusability) – розробка для повторного використання (Low – не вимагається; Extra High – передбачається повторне використання в інших продуктах);

– параметри платформи:

5. 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:

6. FCIL (Facilities) – обладнання (Extra Low – інструменти найпростіші, комунікації ускладнені; Extra High – інтегровані засоби підтримки життєвого циклу, інтерактивні мультимедіа комунікації);

7. SCED (Required Development Schedule) – необхідний виконання графіка робіт (Very Low – 75% від номінальної тривалості; Very High – 160% від номінальної тривалості).

Таблиця 25 – Значення множників трудомісткості

№	Множник зусиль	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Екстра високий	Множник зусиль, ЕМІ
Кадрові фактори								
1	ACAP	Можливість аналітики	1,42	1,29	1,00	0,85	0,71	n/a
2	AEXP	Досвід застосування	1,22	1,10	1,00	0,88	0,81	n/a
3	PCAP	Можливості програміста	1,34	1,15	1,00	0,88	0,76	n/a
4	PCON	Неперервність персоналу	1,29	1,12	1,00	0,90	0,81	n/a
5	PEXP	Досвід платформи	1,19	1,09	1,00	0,91	0,85	n/a
6	LTEX	Досвід мови та інструментів	1,20	1,09	1,00	0,91	0,84	n/a
Фактори продукту								
7	RELY	Необхідна надійність програмного забезпечення	0,84	0,92	1,00	1,10	1,26	n/a
8	DATA	Розмір бази даних	n/a	0,23	1,00	1,14	1,28	n/a
9	CPLX	Складність програмного продукту	0,73	0,87	1,00	1,17	1,34	1,74
10	RUSE	Необхідна можливість повторного використання	n/a	0,95	1,00	1,07	1,15	1,24
11	DOCU	Відповідність документації	0,81	0,91	1,00	1,11	1,23	n/a
Фактори платформи								
12	TIME	Обмеження часу виконання	n/a	n/a	1,00	1,11	1,29	1,63
13	STOR	Основне обмеження зберігання	n/a	n/a	1,00	1,05	1,17	1,46

69

№	Множник зусиль	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Екстра високий	Множник зусиль, ЕМІ
14	PVOL	Волатильність платформи	n/a	0,87	1,00	1,15	1,30	n/a
Фактори проєкту								
15	TOOL	Використання програмних засобів	1,17	1,09	1,00	0,90	0,78	n/a
17	SITE	Багатосайтовий розвиток	1,22	1,09	1,00	0,93	0,86	0,80
16	SCED	Необхідний графік розробки	1,43	1,14	1,00	1,00	1,00	n/a

- 1- Very High Висока кваліфікація аналітиків важлива для правильного розуміння та визначення вимог проекту.
- 2 - Very High Проект великий, і великий досвід у розробці застосувань підтримує високу продуктивність розробників.
- 3 - Very High Високі навички програмістів важливі для швидкого та ефективного впровадження функціональності.
- 4 - Nominal Тривалість роботи персоналу не вважається критичною на цьому етапі.
- 5 - Very High Великий досвід роботи з платформою сприяє ефективній розробці та усуненню проблем.
- 6 - Very High Великий досвід використання мов програмування і інструментальних засобів полегшує розробку.
- 7 - High Проект вимагає високої надійності, що впливає на трудомісткість.
- 8 - Nominal Розмір бази даних знаходиться на номінальному рівні.
- 9 - High Проект вважається складним, що впливає на трудомісткість.
- 10 - Very High Висока необхідність можливості багаторазового використання.
- 11 – High Висока відповідність документації потребам життєвого циклу.
- 12 – Nominal Обмеження часу виконання знаходиться на номінальному рівні.
- 13 – Nominal Обмеження пам'яті знаходиться на номінальному рівні.
- 14 – Low Низька змінність платформи вказує на стабільність умов розробки.
- 15 - Very High Високе використання інструментальних програмних засобів полегшує розробку.
- 16 – Low Багатоабонентська (віддалена) розробка знаходиться на низькому рівні.
- 17 - Nominal Необхідний виконання графіка робіт знаходиться на номінальному рівні.

2) Стадія детальної оцінки після опрацювання архітектури (Post Architecture). Для цієї оцінки необхідно оцінити для проекту рівень сімнадцяти множників трудомісткості :

– параметри персоналу:

- 1) Analyst Capability (ACAP) – можливості аналітика;
- 2) Applications Experience (AEXP) – досвід розробки застосувань;
- 3) Programmer Capability (PCAP) – можливості програміста;
- 4) Personnel Continuity (PCON) – тривалість роботи персоналу;
- 5) Platform Experience (PEXP) – досвід роботи з платформою;
- 6) Language and Tool Experience (LTEX) – досвід використання мови програмування і інструментальних засобів.

– параметри продукту:

- 7) Required Software Reliability (RELY) – необхідна надійність програми;
- 8) Database Size (DATA) – розмір бази даних;
- 9) Software Product Complexity (CPLX) – складність програми;
- 10) Required Reusability (RUSE) – необхідна можливість багаторазового використання;
- 11) Documentation Match to Life-Cycle Needs (DOCU) – відповідність документації потребам життєвого циклу.

– параметри платформи:

- 12) Execution Time Constraint (TIME) – обмеження часу виконання;
- 13) Main Storage Constraint (STOR) – обмеження пам'яті;
- 14) Platform Volatility (PVOL) – змінність платформи.

– параметри проекту:

15) Use of Software Tools (TOOL) – використання інструментальних програмних засобів;

16) Multisite Development (SITE) – багатоабонентська (віддалена) розробка;

17) Required Development Schedule (SCED) – необхідний виконання графіка робіт.

$$E = 0,91 + 0,01 * (2,48 + 3,04 + 4,24 + 2,19 + 4,68) = 1,0763$$

$$EAF \text{ Early Design} = 0,5 * 0,62 * 1 * 0,95 * 0,62 * 1 = 0,182$$

$$EAF \text{ Post Architecture} = 0,71 * 0,81 * 0,76 * 1 * 0,85 * 0,84 * 1,1 * 1 * 1,17 * 1,15 * 1,1 * 1 * 1 * 0,87 * 0,78 * 1,09 * 1 = 0,375$$

$$PM \text{ Early Design} = 0,182 * 2,94 * (35,9 \wedge 1,0763) = 25,24$$

$$PM \text{ Post Architecture} = 0,375 * 2,45 * (35,9 \wedge 1,0763) = 43,34$$

$$TM \text{ Early Design} = 1 * 3,67 * (25,24 \wedge (0,28 + 0,2 * (1,0763 - 0,91))) = 10,08$$

$$TM \text{ Post Architecture} = 1 * 3,67 * (43,34 \wedge (0,28 + 0,2 * (1,0763 - 0,91))) = 11,95$$

Формула оцінки трудомісткості проекту в люд. × міс. має вигляд:

$$PM = EAF \times A \times (SIZE)^E$$

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

де

$B = 0,91$; $A = 2,94$ для попередньої оцінки;

$A = 2,45$ для детальної оцінки;

Тривалість проекту або час розробки проекту TM в СОСОМО II для обох рівнів розраховується за формулою:

$$TM = SCED \times C \times (PM_{NS})^{B+0,2 \times (E-B)}$$

де $C = 3,67$; $D = 0,28$;

PM_{NS} – розрахована трудомісткість проекту без урахування множника,

SCED що визначає ущільнення розкладу.

Бачимо, що оцінки РМ моделі COSOMO 1 більші за оцінки COSOMO 2

	COCOMO.1	COCOMO.2	COCOMO.2.1	COCOMO.2.2
PM	<u>165,5</u>	<u>89,37</u>	<u>25.24</u>	<u>43.34</u>
TM	<u>14,94</u>	<u>12,04</u>	<u>10.08</u>	<u>11.95</u>

4. 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.

	<u>COCOMO.1</u>	<u>COCOMO.2</u>	<u>COCOMO.2.1</u>	<u>COCOMO.2.2</u>
<u>PM</u>	<u>165,5</u>	<u>165,5</u>	<u>144,27</u>	<u>120,23</u>
<u>TM</u>	<u>14,94</u>	<u>14,94</u>	<u>17,61</u>	<u>16,62</u>

$$E = 0,91 + 0,01 * (2,48 + 3,04 + 4,24 + 3,29 + 4,68) = 1,0873$$

$$EAF \text{ Early Design} = 1 * 1 * 1 * 1 * 1 * 1 = 1$$

$$EAF \text{ Post Architecture} = 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 = 1$$

$$PM \text{ Early Design} = 1 * 2,94 * (35,9 ^ 1,0873) = 144,27$$

$$PM \text{ Post Architecture} = 1 * 2,45 * (35,9 ^ 1,0873) = 120,23$$

$$TM \text{ Early Design} = 1 * 3,67 * (144,27 ^ (0,28 + 0,2 * (1,0873 - 0,91))) = 17,61$$

$$TM \text{ Post Architecture} = 1 * 3,67 * (120,23 ^ (0,28 + 0,2 * (1,0873 - 0,91))) = 16,62$$

IV. Висновки.

У ході лабораторної роботи було проведено В ході виконання даної роботи, я навчився визначати трудомісткість, час роботи, чисельність персоналу та продуктивність залежно від стадії розробки за методикою COSOMO.1 та COSOMO.2 При всіх номінальних значеннях модель проміжного рівня COSOMO дорівнює базовій COSOMO. Оцінки РМ COSOMO.2 більші ніж COSOMO.1.

Виконав: студент ІП-11 Олександр Головня