

ПРАКТИЧНА РОБОТА №6

КОНСТРУКТИВНА МОДЕЛЬ ВАРТОСТІ СОСОМО

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

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Модель СОСОМО (Constructive Cost Model) спрямовано на три моделі створення програмного забезпечення:

- 1) розповсюджений або органічний тип (organic projects);
- 2) напівнезалежний або напіврозподілений тип (semidetached projects);
- 3) вбудований тип (embedded projects).

Модель СОСОМО поділяється на моделі: базовий (basic), проміжний (intermediate), деталізований (advanced).

Модель Basic СОСОМО – двопараметрична. 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

Модель цього рівня підходить для ранньої швидкої приблизної оцінки витрат, але точність її дуже низька.

Модель проміжного рівня уточнена за рахунок введення додаткових 15 факторів витрат (Cost Drivers) (CD_k), які згруповані за чотирма категоріями:

1) Характеристики продукту (Product Attributes):

RELY – надійність (Required Software Reliability);

DATA – Розмір БД (Size of Application Database);

CPLX – Складність продукту (Complexity of the Product);

2) Характеристики апаратного забезпечення (Hardware Attributes):

TIME – Обмеження швидкодії при виконанні програми (Run-Time Performance Constraints);

STOR – Обмеження пам'яті (Memory Constraints);

VIRT (PVOL) – Нестійкість оточення віртуальної машини (Volatility of the Virtual Machine Environment);

TURN (STIME) – Необхідний час відновлення (Required Turnabout Time);

3) Характеристики персоналу (Personnel Attributes):

ACAP (ASAP) – Аналітичні здібності (Analyst Capability);

AEXP – Досвід розробки (Applications Experience);

PCAP (PERS) – Здібності до розробки ПЗ (Software Engineer Capability);

VEXP (PEXP) – Досвід використання віртуальних машин (Virtual Machine Experience);

LEXP (LTEX) – Досвід розробки на мовах програмування (Programming Language Experience);

4) Характеристики проєкту (Project Attributes):

MODP (FCIL) – Застосування методів розробки ПЗ (Application of Software Engineering Methods);

TOOL – Використання інструментарію розробки ПО (Use of Software Tools);

SCED – Вимоги дотримання графіка розробки (Required Development Schedule).

Значення кожного атрибута вибирається з таблиці (табл. 20) відповідно до його ступеня значущості (рейтингу) в конкретному проєкті.

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

Атрибути вартості, CD _k	Рейтинг					
	Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
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
Характеристики проєкту						
13. Застосування методів розробки ПЗ	1,24	1,10	1,00	0,91	0,82	n/a
14. Використання інструментарію розробки ПЗ	1,24	1,10	1,00	0,91	0,83	n/a
15. Вимоги дотримання графіку розробки	1,23	1,08	1,00	1,04	1,10	n/a

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).

EAF – (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.

У 2000 р. методика COSOMO була вдосконалена і отримала назву COSOMO II.

Розрізняють дві стадії оцінки проекту:

- 1) попередня оцінка на початковій фазі (Early Design);
- 2) детальна оцінка після опрацювання архітектури (Post Architecture).

Для розрахунку трудомісткості необхідно спочатку оцінити фактори (чинники) масштабу (Scale Drivers) та множники трудомісткості (Cost Drivers або Effort Multipliers). Фактори масштабу застосовуються на двох стадіях оцінки проекту. Множники трудомісткості відрізняються для різних стадій оцінки проекту. На різних стадіях відрізняється їх кількість і значення. Для стадії Early Design необхідно оцінити сім множників трудомісткості, а для стадії Post Architecture – сімнадцять.

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;

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:

В методиці використовуються п'ять факторів масштабу, які визначаються наступними характеристиками проєкту:

PREC – прецедентність, наявність досвіду аналогічних розробок,

FLEX – гнучкість процесу розробки,

RESL – архітектура і дозвіл ризиків,

TEAM – спрацьованість команди,

PMAT – зрілість процесів.

Всі фактори масштабу мають певну оцінку: Very Low – дуже низька оцінка фактора, Low – низька оцінка, Nominal – середня оцінка, High – висока оцінка, Very High – дуже висока оцінка, Extra High – критично висока оцінка. Детальний опис факторів масштабу наведено в таблиці 22.

Таблиця 22 – Опис рівнів значущості факторів масштабу

SFj	Опис	Рівень значущості факторів					
		Дуже низький	Низький	Середній	Високий	Дуже високий	Критичний
1. PREC. Precedentness	Прецедентність, наявність досвіду аналогічних розробок	досвід у продукті і платформі відсутній	продукт і платформа не дуже знайомі	деякий досвід в продукті і платформах 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

Зазначені фактори застосовуються на обох стадіях оцінки проєкту.

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, SF _j	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

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, плінність більше 45%; Extra High – аналітики і програмісти мають вищу кваліфікацію, плінність 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).

Значення множників трудомісткості в залежності від їх рівня наведені в табл. 23.

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

№	Множник трудомісткості, ЕМі	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

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

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

– параметри персоналу:

- 1) Analyst Capability (ACAP) – можливості аналітика;
- 2) Applications Experience (AEXP) – досвід розробки застосунків;
- 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 проєкту:

- 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.

Значення множників трудомісткості в залежності від їх рівня наведені в табл. 25.

Таблиця 25 – Значення множників трудомісткості

№	Множник зусиль	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Екстра високий	Множник зусиль, ЕМЛ
Кадрові фактори								
1	ACAP	Можливість аналітика	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

№	Множник зусиль	Дуже низький	Низький	Номінальний	Високий	Дуже високий	Екстра високий	Множник зусиль, ЕМЈ
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

Примітка: n/a (not available) - дані відсутні, тобто відповідний рівень не оцінюється.

Тривалість проєкту або час розробки проєкту TM в COCOMO II для обох рівнів розраховується за формулою:

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

де $C = 3,67$; $D = 0.28$;

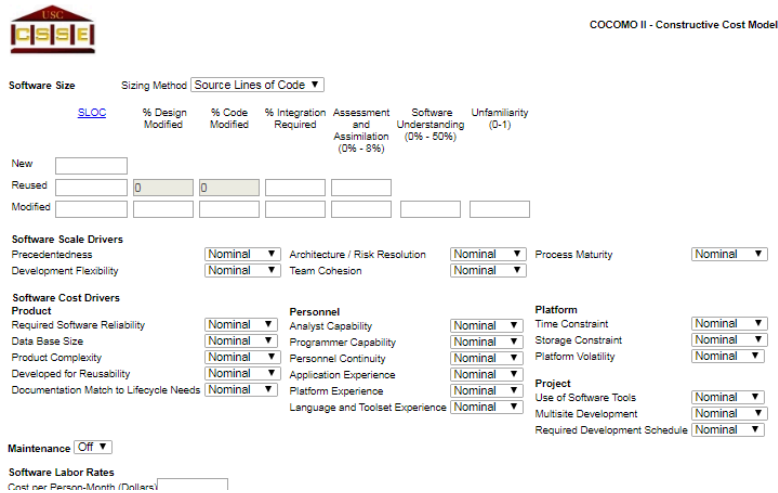
PM_{NS} – розрахована трудомісткість проєкту без урахування множника, SCED що визначає ущільнення розкладу.

Програмні застосунки для використання моделі COCOMO

Для виконання розрахунків з використання моделі COCOMO можна використовувати інструменти, такі як COCOMO Suite of Constructive Cost Models. Даний інструмент доступний за посиланням:

<https://csse.usc.edu/tools/COCOMOII.php>

Розглянемо приклад роботи онлайн-калькулятора (рис. 23).



The screenshot shows the USC CSSE COCOMO II - Constructive Cost Model online calculator interface. At the top left is the USC CSSE logo. The title bar reads "COCOMO II - Constructive Cost Model". Below the logo, there's a "Software Size" section with a "Sizing Method" dropdown set to "Source Lines of Code". A link "SLOC" is provided. Below this are input fields for "New", "Reused", and "Modified" software, each with sub-fields for "% Design Modified", "% Code Modified", "% Integration Required", "Assessment and Assimilation (0% - 8%)", "Software Understanding (0% - 50%)", and "Unfamiliarity (0-1)".

The main section is divided into several categories of drivers, each with a "Nominal" dropdown menu:

- Software Scale Drivers:** Precedentedness, Development Flexibility, Architecture / Risk Resolution, Team Cohesion, Process Maturity.
- Software Cost Drivers:** Product (Required Software Reliability, Data Base Size, Product Complexity, Developed for Reusability, Documentation Match to Lifecycle Needs), Personnel (Analyst Capability, Programmer Capability, Personnel Continuity, Application Experience, Platform Experience, Language and Toolset Experience), Platform (Time Constraint, Storage Constraint, Platform Volatility), and Project (Use of Software Tools, Multisite Development, Required Development Schedule).

At the bottom, there's a "Maintenance" dropdown set to "Off" and a "Software Labor Rates" section with a "Cost per Person-Month (Dollars)" input field.

Рисунок 23 – Інтерфейс онлайн калькулятора

На сайті, що доступний за наведеним нижче посиланням знаходяться посилання на інші доступні інструменти по розрахунку за моделями COCOMO, що розроблені в USC CSSE.

<https://csse.usc.edu/csse/tools/>

Також для розрахунків можна застосовувати інструмент Costar (рис. 24), який розроблено компанією SoftStar на моделі COCOMO II для автоматизації оцінки вартості розробки програмних продуктів. Завантажити програму можна за посиланням: <http://www.softstarsystems.com/>

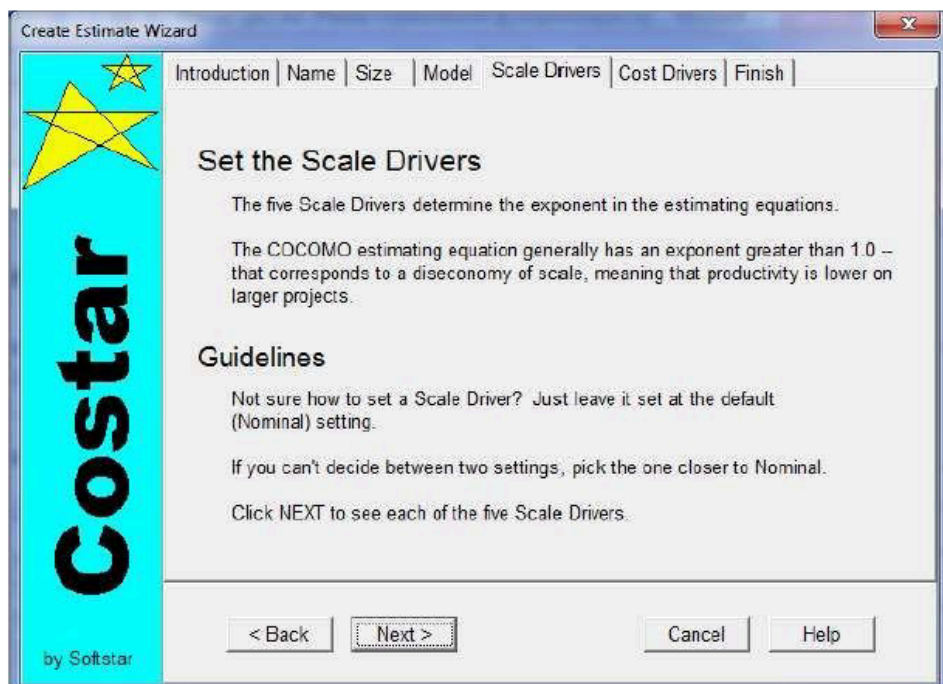


Рисунок 24 – Налаштування 5 факторів масштабу у інструменті Coststar.

Приклад виконання

Програмний застосунок, що виконує роль приклада, являє собою об'єктно-об'єктний маппер у .NET, що базується на конвенціях (рис. 25).

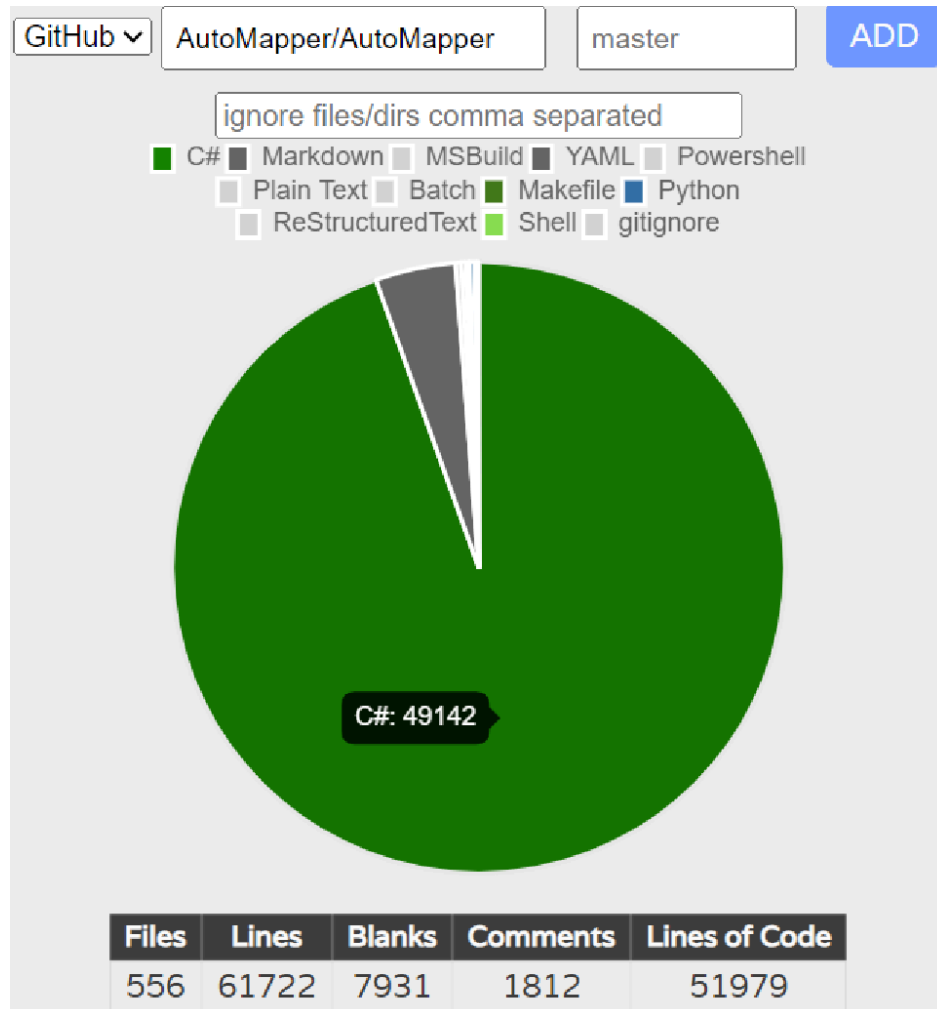


Рисунок 25 – Приклад роботи програмного застосунка

Це бібліотека, створена для вирішення оманливо складної проблеми - позбавлення від коду, який зіставляє один об'єкт з іншим. Розмір проєкту: 51979 рядків програмного коду.

Базова модель COCOMO: AutoMapper - це об'єктно-об'єктний маппер на основі конвенцій. Він розроблений для сценаріїв проєкції моделей, щоб згладити складні моделі об'єктів до DTO та інших простих об'єктів. З огляду на його складність і досвід, необхідний для розробки такого проєкту, він, швидше за все, підпадає під категорію "Semi-Detached" в моделі COCOMO.

Розрахунки:

$$PM = a * (SIZE)^b = 3 * 51,979^{1,12} = 250,52$$

$$TM = c * (PM)^d = 2,5 * (250,52)^{0,35} = 17,28$$

$$SS = \frac{PM}{TM} = \frac{250,52}{17,28} = 14,5$$

$$P = \frac{SIZE}{PM} = \frac{51,979}{250,52} = 0,2075$$

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);

SS – середня чисельність персоналу;

P – продуктивність.

Проміжна модель COCOMO:

Вибір значень атрибутів вартості з поясненням – нижче у таблиці 26:

Таблиця 26 – Вибір значень атрибутів вартості

Атрибути вартості, CDk	Рейтинг	Причина
Характеристики продукту		
1. Необхідна надійність ПЗ	Високий (1,15)	AutoMapper є широко використовуваною бібліотекою, тому надійність має вирішальне значення.
2. Розмір БД додатка	Дуже низький (n/a)	AutoMapper не має бази даних.
3. Складність продукту	Високий (1,15)	AutoMapper використовує алгоритм зіставлення на основі

Атрибути вартості, CDk	Рейтинг	Причина
		конвенцій для зіставлення значень джерела і призначення, що додає йому складності.
Характеристики апаратного забезпечення		
4. Обмеження швидкодії при виконанні програми	Середній (1)	AutoMapper розроблено ефективним, але його продуктивність може залежати від того, як він використовується у програмах
5. Обмеження пам'яті	Середній (1)	AutoMapper сам по собі не має великого розміру, але використання пам'яті може залежати від розміру та складності об'єктів, що зіставляються
6. Нестійкість оточення віртуальної машини	Дуже низький (n/a)	AutoMapper не потребує віртуальної машини
7. Необхідний час відновлення	Середній (1)	Хоча AutoMapper не працює у режимі реального часу, продуктивність все одно важлива
Характеристики персоналу		
8. Аналітичні здібності	Високий (0,86)	Розробка AutoMapper або участь у ньому вимагає хороших аналітичних навичок для розуміння об'єкт-об'єктного зіставлення та конвенцій
9. Досвід розробки	Високий (0,91)	Досвід роботи зі сценаріями мапування має вирішальне значення
10. Здібності до розробки ПЗ	Високий (0,86)	AutoMapper є добре розробленим проєктом
11. Досвід використання віртуальних машин	Дуже низький (1,21)	AutoMapper не вимагає використання віртуальних машин

Атрибути вартості, CDk	Рейтинг	Причина
12. Досвід розробки на мовах програмування	Високий (0,95)	AutoMapper написаний на C#, тому досвід роботи з цією мовою є важливим.
Характеристики проекту		
13. Застосування методів розробки ПЗ	Високий (0,91)	Враховуючи складність проекту, швидше за все, будуть використані формальні методи
14. Використання інструментарію розробки ПЗ	Середній (1)	Буде використано стандартні засоби розробки програмного забезпечення, такі як IDE та контроль версій
15. Вимоги дотримання графіку розробки	Середній (1)	Оскільки це проєкт з відкритим вихідним кодом, розробка може не бути строго запланованою

Розрахунки було здійснено за наступною формулою:

$$EAF = \prod_{k=1}^{15} CD_k = 1,172$$

$$PM = EAF * a * (SIZE)^b = 1,172 * 3 * 51,979^{1,12} = 293,61$$

$$TM = c * (PM)^d = 2,5 * (293,61)^{0,35} = 18,27$$

Де:

EAF (Effort Adjustment Factor) – добуток обраних атрибутів вартості

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);

COCOMO II попередня та детальна оцінка:

Фактори масштабу у табличному вигляді (табл. 27):

Таблиця 27 – Фактори масштабу

Чинник масштабу	Значення	Причина
PREC (Precedentedness)	Very High (1,24)	AutoMapper - це добре відпрацьована, широко використовувана бібліотека. Можна з упевненістю припустити, що команда має значний досвід роботи з продуктом і платформою.
FLEX (Development Flexibility)	High (2,03)	Враховуючи, що AutoMapper має відкритий вихідний код і внесок від різних розробників, процес розробки, ймовірно, є гнучким. Не маючи конкретних знань про внутрішні процеси, можна зробити консервативну оцінку.
RESL (Architecture/Risk Resolution)	High (2,83)	AutoMapper є зрілим проектом, що свідчить про те, що більшість архітектурних ризиків було враховано. Однак, не маючи конкретних даних аналізу ризиків, ми робимо консервативну оцінку.
TEAM (Team Cohesion)	Nominal (3,29)	Оскільки це проект з відкритим вихідним кодом, згуртованість команди може змінюватися. Учасники можуть не взаємодіяти так тісно, як команда, що знаходиться в одному приміщенні.
PMAT (Process Maturity)	Nominal (4,68)	Без конкретних знань про рівень їхньої моделі зрілості можливостей (СММ) важко надати точну оцінку. З огляду на очевидний успіх і широке використання проекту, ми робимо консервативну оцінку.

Множники трудомісткості у табличному вигляді (попередня оцінка)
наведено у табл. 28:

Таблиця 28 – Множники трудомісткості (попередня оцінка)

Множник трудомісткості	Значення	Причина
PERS (Personnel Capability)	High (0,83)	AutoMapper - це добре налагоджений проєкт з великою спільнотою учасників. Можна з упевненістю припустити, що учасники мають високу кваліфікацію.

Множник трудомісткості	Значення	Причина
PREX (Personnel Experience)	Very High (0,74)	Враховуючи, що AutoMapper є зрілим проектом, цілком ймовірно, що учасники добре знайомі з програмою, інструментами та платформою.
RCPX (Product Reliability and Complexity)	High (1,33)	AutoMapper є широко використовуваною бібліотекою, що свідчить про її високу надійність. Крім того, враховуючи її функціональність, вона, ймовірно, має помірний рівень складності.
RUSE (Developed for Reusability)	High (1,07)	AutoMapper призначено для багаторазового використання у різних проектах, тож цілком ймовірно, що її було розроблено з думкою про багаторазове використання.
PDIF (Platform Difficulty)	Low (0,87)	AutoMapper є бібліотекою .NET, а .NET є стабільною платформою без серйозних обмежень на пам'ять і продуктивність.
FCIL (Facilities)	High (0,87)	Враховуючи, що AutoMapper є проектом з відкритим вихідним кодом, цілком ймовірно, що учасники використовують різноманітні інструменти і мають хороші канали зв'язку.
SCED (Required Development Schedule)	Nominal (1)	Не маючи конкретних знань про графік їхньої розробки, важко надати точну оцінку. Номінальне значення використовується як консервативна оцінка.

Множники трудомісткості (детальна оцінка) наведено у таблиці 29:

Таблиця 29 – Множники трудомісткості (детальна оцінка)

Множник трудомісткості	Значення	Причина
Analyst Capability (ACAP)	High (0,85)	Зважаючи на складність проєкту AutoMapper, цілком ймовірно, що в ньому беруть участь висококваліфіковані аналітики.
Applications Experience (AEXP)	High (0,88)	AutoMapper є зрілим проєктом, що свідчить про те, що команда має значний досвід розробки додатків.
Programmer Capability (PCAP)	High (0,88)	Якість коду та широке використання бібліотеки свідчать про високий рівень здібностей програмістів.
Personnel Continuity (PCON)	Nominal (1)	Не маючи конкретної інформації про плинність кадрів в команді, найбезпечніше припустити номінальну оцінку.
Platform Experience (PEXP)	High (0,91)	Проєкт побудований на .NET, і команда, схоже, добре володіє цією платформою.
Language and Tool Experience (LTEX)	High (0,91)	Проєкт написаний на C#, і досвід команди в цій мові видно з кодової бази.
Required Software Reliability (RELY)	Nominal (1)	Бібліотека широко використовується, що свідчить про необхідність надійної роботи, але це не та сфера, де надзвичайна надійність є критично важливою.
Database Size (DATA)	Very Low (n/a)	AutoMapper не орієнтована на роботу з базами даних; вона в першу чергу призначена для зіставлення об'єктів між собою.
Software Product Complexity (CPLX)	High (1,17)	AutoMapper вирішує складну задачу - зіставлення між об'єктами, що передбачає роботу з різними сценаріями.
Required Reusability (RUSE)	High (1,07)	Так як бібліотека призначена для використання в інших проєктах,

Множник трудомісткості	Значення	Причина
		потрібна висока придатність до повторного використання.
Documentation Match to Life Cycle Needs (DOCU)	High (1,11)	Проект має велику документацію.
Execution Time Constraint (TIME)	Nominal (1)	Хоча продуктивність є важливою, немає жодних доказів екстремальних обмежень часу виконання.
Main Storage Constraint (STOR)	Nominal (1)	AutoMapper не є вимогливим до обсягу пам'яті.
Platform Volatility (PVOL)	Low (0,87)	Платформа .NET є зрілою та стабільною.
Use of Software Tools (TOOL)	Nominal (1)	AutoMapper використовує стандартний набір програмних інструментів, типовий для проектів такого типу.
Multisite Development (SITE)	Nominal (1)	Не маючи конкретної інформації про місця розробки, можна припустити номінальну оцінку.
Required Development Schedule (SCED)	Nominal (1)	Немає жодних доказів екстремального тиску на графік.

Розрахунки для попередньої оцінки:

$$E = B + 0,01 * \sum_{j=1}^5 SF_j = 1,0507$$

$$EAF = \prod_{k=1}^7 EM_k = 0,6616$$

$$PM = EAF * A * (SIZE)^E = 0,6616 * 2,94 * 51,979^{1,0507} = 123,53$$

$$TM = SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (123,53)^{0,28+0,2*(1,0507-0,91)} = 16,189$$

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);

Розрахунки для детальної оцінки:

$$E = B + 0,01 * \sum_{j=1}^5 SF_j = 1,0507$$

$$EAF = \prod_{k=1}^{17} EM_k = 0,659$$

$$PM = EAF * A * (SIZE)^E = 0,659 * 2,45 * 51,979^{1,0507} = 102,535$$

$$TM = SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (102,535)^{0,28+0,2*(1,0507-0,91)} = 15,286$$

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);

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 50 000 рядків програмного коду, зробимо розрахунки для розмірів 25000, 75000 та 100 000:

Для цих розрахунків:

$$E = B + 0,01 * \sum_{j=1}^5 SF_j = 1,0507$$

Для попередньої оцінки:

$$EAF = \prod_{k=1}^7 EM_k = 0,6616$$

Для детальної оцінки:

$$EAF = \prod_{k=1}^{17} EM_k = 0,659$$

Розмір 25 000 рядків програмного коду:

Попередня оцінка:

$$\begin{aligned} PM &= EAF * A * (SIZE)^E = 0,6616 * 2,94 * 25^{1,0507} = 57,248 \\ TM &= SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (57,248)^{0,28+0,2*(1,0507-0,91)} \\ &= 16,189 \end{aligned}$$

Детальна оцінка:

$$\begin{aligned} PM &= EAF * A * (SIZE)^E = 0,659 * 2,45 * 25^{1,0507} = 47,519 \\ TM &= SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (47,519)^{0,28+0,2*(1,0507-0,91)} \\ &= 15,286 \end{aligned}$$

Розмір 75 000 рядків програмного коду:

Попередня оцінка:

$$\begin{aligned} PM &= EAF * A * (SIZE)^E = 0,6616 * 2,94 * 75^{1,0507} = 181,581 \\ TM &= SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (181,581)^{0,28+0,2*(1,0507-0,91)} \\ &= 18,23 \end{aligned}$$

Детальна оцінка:

$$\begin{aligned} PM &= EAF * A * (SIZE)^E = 0,659 * 2,45 * 75^{1,0507} = 150,726 \\ TM &= SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (150,726)^{0,28+0,2*(1,0507-0,91)} \\ &= 17,213 \end{aligned}$$

Розмір 100 тисяч:

Попередня оцінка:

$$\begin{aligned} PM &= EAF * A * (SIZE)^E = 0,6616 * 2,94 * 100^{1,0507} = 245,665 \\ TM &= SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (245,665)^{0,28+0,2*(1,0507-0,91)} \\ &= 20,01 \end{aligned}$$

Детальна оцінка:

$$PM = EAF * A * (SIZE)^E = 0,659 * 2,45 * 100^{1,0507} = 203,916$$

$$TM = SCED * C * (PM_{NS})^{D+0,2*(E-B)} = 1 * 3,67 * (203,916)^{0,28+0,2*(1,0507-0,91)} = 18,893$$

Отримати значення PM та TM по всім моделям для одного й того ж значення параметра SIZE, обравши номінальний (середній) рівень складності проекту, що має високу ступінь новизни.

Проміжна COCOMO, розмір – 100 000 рядків програмного коду (рис. 26):

Product Attributes	
Required Reliability	1.15 (H)
Database Size	1.00 (N)
Product Complexity	1.00 (N)
Computer Attributes	
Execution Time Constraint	1.00 (N)
Main Storage Constraint	1.00 (N)
Platform Volatility	1.00 (N)
Computer Turnaround Time	1.00 (N)
Personnel Attributes	
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)
Project Attributes	
Modern Programming Practices	1.10 (L)
Use of Software Tools	1.10 (L)
Required Development Schedule	1.00 (N)

Рис. 26. Параметри проекту

Результат виконаних розрахунків представлено на рисунку 27:

COCOMO RESULTS for Semi-detached 100								
MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person-months)	DURATION, (in months)	STAFFING, (recommended)
semi-detached	2.4021984086100003	1.12	2.5	0.35	100.000	417.454	20.661	20.205

Рисунок 27 – Результат виконаних розрахунків

COCOMO II, розмір – 100 000 рядків програмного коду.

Параметри представлено на рисунку 28:

COCOMO II - Constructive Cost Model

Software Size Sizing Method Source Lines of Code ▾

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	<input type="text" value="100000"/>						
Reused	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>		
Modified	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Software Scale Drivers

Precedentedness	Low ▾	Architecture / Risk Resolution	High ▾	Process Maturity	High ▾
Development Flexibility	High ▾	Team Cohesion	High ▾		

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	High ▾	Analyst Capability	High ▾	Time Constraint	Nominal ▾
Data Base Size	Nominal ▾	Programmer Capability	High ▾	Storage Constraint	Nominal ▾
Product Complexity	Nominal ▾	Personnel Continuity	High ▾	Platform Volatility	Nominal ▾
Developed for Reusability	High ▾	Application Experience	High ▾		
Documentation Match to Lifecycle Needs	High ▾	Platform Experience	High ▾	Project	
		Language and Toolset Experience	High ▾	Use of Software Tools	High ▾
				Multisite Development	Nominal ▾
				Required Development Schedule	Nominal ▾

Рисунок 28 – Параметри проекту, що містить більше 100 000 рядків програмного коду

Результат проведених розрахунків:

Effort = 224.9 Person-months
Schedule = 19.7 Months

Таким чином, було проведено дослідження трудомісткості розробки програмного продукту за допомогою моделей COCOMO та COCOMO II. Було встановлено, що розмір програмного коду (SIZE) має значний вплив на трудомісткість (PM) та час розробки проекту (TM) для різних моделей COCOMO II [39].

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.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Охарактеризуйте та порівняйте моделі COCOMO та COCOMO II.
2. Які існують рівні створення продукту моделі COCOMO?
3. Які характеристики мають organic projects моделі COCOMO?
4. Які характеристики мають semidetached projects моделі COCOMO?
5. Які характеристики мають embedded projects моделі COCOMO?
6. Назвіть рівняння базового рівня моделі COCOMO.
7. Назвіть характеристики продуктів проміжного рівня моделі COCOMO?
8. Які характеристики персоналу проміжного рівня моделі COCOMO?
9. Якою є роль Effort Adjustment Factor?
10. Які існують стадії оцінки проєкту у моделі COCOMO II.
11. Дайте визначення Scale Drivers, Effort Multipliers.
12. Чи відрізняються фактори масштабу для різних стадій оцінки проєкту?
13. Які множники трудомісткості необхідно оцінити для стадії попередньої оцінки проєкту?
14. Які множники трудомісткості необхідно оцінити для стадії детальної оцінки проєкту?