

Затверджено науково-  
методичною  
радою ЖДТУ  
протокол від  
«\_\_» \_\_\_\_\_ 20\_\_ р.  
№\_\_

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**  
для проведення лабораторних робіт  
з навчальної дисципліни  
**«ЕКОНОМІКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»**

для студентів освітнього рівня «бакалавр»  
денної і заочної форми навчання

спеціальності 121 «Інженерія програмного забезпечення»  
освітньо-професійна програма  
«Інженерія програмного забезпечення»

факультет Інформаційно-комп'ютерних технологій

кафедра Інженерії програмного забезпечення

Розглянуто і  
рекомендовано  
на засіданні  
кафедри  
ІПЗ  
протокол №7  
від 04 березня  
2014р.

Розробник: старший викладач кафедри ІПЗ, Кравченко С.М.

Житомир  
2017 – 2018 н.р.

Методичні вказівки до лабораторного практикуму з дисципліни  
«Економіка програмного забезпечення» для студентів напрямку  
«Програмна інженерія» / Укладач С.М.Кравченко. – Житомир:  
ЖДТУ, 2014. – 48с.

**Укладач:** Кравченко Світлана Миколаївна, асистент кафедри  
програмного забезпечення систем

**Відповідальний редактор:** Панішев Анатолій Васильович,  
завідувач кафедри програмного забезпечення систем, д.т.н.,  
професор

**Рецензент:** Крижанівський В'ячеслав Борисович, доцент  
кафедри ПЗС, к.ф.-м.н, доцент

## Передмова

Метою дисципліни є вивчення теоретичних основ і принципів метрики складності програм, методів, моделей і засобів оцінки вартості програмного забезпечення

У результаті вивчення дисципліни студенти повинні:

- знати теоретичні основи різноманітних методів оцінки вартості програмного забезпечення;
- розраховувати трудомісткість розробки програмного продукту;
- вміти оцінювати вартість ПЗ на основі методики СОСОМО II;
- ознайомитись з методом функціональних точок;
- вміти користуватися засобами оцінки вартості програмного забезпечення;
- використовувати метод узгодженої оцінки проекту (PERT);
- застосовувати засоби для оцінки вартості ПЗ.

Програма курсу передбачає навчання у формі лекцій і лабораторних занять. Для практичного засвоєння основних тем дисципліни передбачені лабораторні заняття із застосуванням ПК у класах комп'ютерного центру університету. Саме ця частина і висвітлена в методичних вказівках, яку ви зараз читаєте.

# **Лабораторна робота № 1.1. Розрахунок трудомісткості розробки програмного продукту**

**Мета роботи :** Навчитися розраховувати трудомісткість ПП

## **Короткі відомості**

Для визначення витрат праці на розробку використовуються аналітичні і експертні методи оцінок. Вибір методу здійснюється в залежності від ступеню обліку факторів, які впливають на трудомісткість розробки. Для більш точного визначення трудомісткості розробки програмного продукту по окремим елементам і операціям процесу може бути використаний метод, в якому складаючі витрати праці визначаються з врахуванням особливості організації, яка веде розробку, і основних параметрів програмного продукту: ступеню новизни задачі, складності алгоритму, кількості різновидностей вхідної і вихідної інформації, складності організації контролю вхідної і вихідної інформації, мови програмування, використання стандартних модулів і типових задач.

Надати механізм розрахунку трудомісткості і вартості робіт проекту створення інформаційної системи державних органів на стадії розробки техніко-економічного обґрунтування проекту (до початку проектування інформаційної системи).

## **Задачі:**

Забезпечити єдиний підхід до оцінки трудомісткості і вартості всіх проектів створення інформаційних систем.

Визначити єдині нормативи на створення, розвиток і супровід інформаційних систем.

Методика включає оцінку трудомісткості тільки на розробку прикладного програмного забезпечення інформаційних систем і виключає компоненти, які вже були створені або є умовами їхнього функціонування :

Апаратне забезпечення (обчислювальне й телекомунікаційне устаткування);

Готові програмні продукти (ОС, СУБД, сервера додатків, галузеві додатки й ін.) від ІТ - вендорів (Microsoft, SAP, Oracle, IBM, Fujitsu ін.);

Готові платформи розробки (мова програмування, СУБД, бібліотеки компонентів);

Інженерна інфраструктура (серверні приміщення);

Послуги зв'язку (Інтернет, виділені канали та інш.).

## **Робоче завдання**

Функціональний розмір прикладного програмного забезпечення інформаційних систем визначається по формулі оцінки складності майбутнього проекту в балах функціональності, запропонований А.Альбрехтом:

$$\Phi P = (K_1 + K_2 + K_3)^{2,35} \quad (1)$$

Було використано емпіричне правило — зростання розміру ПЗ втриє збільшує трудомісткість розробки і виготовлення в сім раз.

Показник степені зростання трудомісткості з ростом об'єму коду дорівнює 2,35.

де класифікатори проекту створення інформаційної системи:

$K_1$  - масштаб об'єкту автоматизації;

$K_2$  - тип замовника;

$K_3$  - тип програмного забезпечення.

визначаються по таблиці №1

Таблиця №1

<b>К<sub>1</sub> – масштаб об'єкту автоматизації</b>	<b>К<sub>2</sub> – тип замовника</b>	<b>К<sub>3</sub> – тип програмного забезпечення</b>
Автоматизація бізнес процесу одного структурного підрозділу <b>1</b>	Місцевий виконавчий орган - <b>8</b>	Готове програмне забезпечення, яке потребує налаштування - <b>1</b>
Автоматизація бізнес-процесів одного відомства - <b>8</b>	Центральний державний орган - <b>14</b>	База даних - <b>6</b>
Автоматизація бізнес-процесів одного відомства з територіальними підрозділами - <b>9</b>	Державний орган, діяльність якого пов'язана з небезпекою для життя - <b>15</b>	Клієнт-серверне (товстий клієнт) - <b>8</b>
Автоматизація бізнес-процесів відомства і інтеграція з зовнішніми інформаційними системами - <b>10</b>		Клієнт-серверне (тонкий клієнт) - <b>11</b>
Автоматизація бізнес-процесів декількох відомств - <b>12</b>		Сервіс-орієнтоване - <b>15</b>
Автоматизація бізнес-процесів декількох відомств і інтеграція з зовнішніми інформаційними системами - <b>13</b>		

Розмір коду прикладного програмного забезпечення інформаційної системи в тисячах логічних рядків вихідного коду (далі – РК) визначається за формулою 2:

$$PK = \Phi P \cdot KII / 1000 \quad (2)$$

де КП - коефіцієнт переводу балу функціональності в кількість логічних рядків коду, значення якого визначається за таблицею 1.1 (додаток 1)

## **Лабораторна робота № 1.2. Визначення трудомісткості розробки ППЗ інформаційної системи**

Розрахунок трудомісткості розробки прикладного програмного забезпечення інформаційної системи в людино-місяцях (далі – Т) на основі раніше визначених даних - РК, Е, Z за формулою 4:

$$T = 2,94 \cdot (PK)^E \cdot \prod_{i=1}^7 Z_i, \quad (4)$$

де Z - показник витрат трудомісткості створення (розробки)

де E - показник масштабу трудомісткості створення (розробки) програмного забезпечення інформаційної системи, який обчислюється за формулою 3:

$$E = 0,91 + 0,01 \cdot \sum_{j=1}^5 R_j. \quad (3)$$

Формули (3,4) використані з СОСОМО II (Методика СОСОМО дозволяє оцінити трудомісткість і час розробки програмного продукту. В моделі використовується формула регресії з параметрами, визначеними на основі галузевих даних і характеристик конкретного проекту.)

Значення кожного показника розробки –  $R_j$  ( $j=1..5$ ), множника затрат -  $Z_i$  ( $i=1..7$ ) визначаються по таблицям 3, 4.

Таблиця 3 містить нормативи кожного показника розробки  $R_j$  залежно від його рівня. Характеристики всіх рівнів по кожному показнику розробки  $R_j$  наведені в таблиці 3. Визначення всіх показників розробки  $R_j$  необхідні для розрахунку показника масштабу трудомісткості створення (розробки) прикладного програмного забезпечення за формулою 3.

Таблиця 3

Показники розробки, $R_j$	Низький рівень	Середній рівень	Високий рівень
передбачуваність проекту для розроблювача, $R_1$	4,96	3,72	2,48
гнучкість процесу розробки, $R_2$	4,05	3,04	2,03
ступінь знищення ризиків, $R_3$	5,65	4,24	2,83
згуртованість команди проекту, $R_4$	4,38	3,29	2,19
зрілість процесів в організації розроблювача, $R_5$	6,24	4,68	3,12

Таблиця 4

Множники витрат, $Z_i$	Низький рівень	Середній рівень	Високий рівень
згуртованість персоналу, $Z_1$	1,20	1,00	0,83
надійність і складність прикладного програмного	0,83	1,00	1,33
складність платформи, $Z_3$	0,87	1,00	1,29
необхідний рівень повторного використання, $Z_4$	0,95	1,00	1,07
досвідченість персоналу, $Z_5$	1,22	1,00	0,87
використання інструментів,	1,10	1,00	0,87
щільність графіка проекту, $Z_7$	1,14	1,00	1,00

## Лабораторна робота № 2. Розрахунок чисельності виконавців проекту, строки виконання роботи.

**Мета роботи:** Навчитися розраховувати загальний об'єм і трудомісткості ПЗ, визначити чисельність виконавців проекту.

### Робоче завдання

Визначення об'єму і трудомісткості ПЗ



Базою для розрахунку планового кошторису витрат на розробку ПЗ є об'єм ПЗ.

Загальний об'єм ( $V_3$ ) програмного продукту визначається за формулою 5, виходячи з кількості і об'єму функцій, які реалізуються програмою (додаток 2, табл. 2.1.)

$$V_3 = \sum_{i=1}^n V_i, \quad (5)$$

де  $V_3$  – об'єм окремої функції ПЗ;

$n$  – загальне число функцій.

Незважаючи на досить значний перелік видів одиниць виміру обсягу ПЗ, найбільш широке поширення одержали лише перші три.

Причому функціональні крапки й крапки властивостей дотепер використовуються тільки в сполученні з кількістю рядків вихідного коду (LOC). Всі інші види одиниць виміри застосовуються в основному при розробці спеціалізованих проєктів.

У даній лабораторній роботі як одиниці виміру обсягу ПЗ використовується рядок вихідного коду (LOC). Переваги використання рядків коду як одиниць виміри укладаються в тім, що ці одиниці:

- відображають сутність праці програмістів;
- широко поширені й можуть легко адаптуватися;
- дозволяють виконувати зіставлення розмірів ПЗ й продуктивності в різних групах розроблювачів;
- безпосередньо пов'язані з кінцевим продуктом;
- можуть використатися для оцінки робіт до завершення проєкту;
- дозволяють автоматизувати збір даних про кількість LOC від початку до кінця проєкту;
- дають можливість враховувати думку розроблювача про обсяг ПЗ на основі кількості написаних рядків коду.

Розрахунок обсягу програмного продукту (кількості рядків вихідного коду) припускає визначення типу програмного забезпечення (додаток 2, табл.2.3.), всебічне технічне обґрунтування функцій ПЗ й визначення обсягу кожної функції.

**Трудомісткість розробки ПЗ.** По уточненому обсязі ПЗ і нормативах витрат праці розраховуючи на одиниці об'єму, визначаються нормативна і загальна трудомісткість розробки ПЗ.

**Нормативна трудомісткість розробки ПЗ.** На підставі прийнятого до розрахунку обсягу ( $V_3$ ) і категорії складності (додаток 2, табл.2.2.) визначається нормативна трудомісткість ПЗ ( $T_H$ ), що уточнюється з урахуванням складності й новизни проекту й ступеня використання стандартних модулів при розробці.

**Загальна трудомісткість розробки ПЗ.** Нормативна трудомісткість ( $T_H$ ) є основою для визначення загальної трудомісткості (ТД розрахунок якої здійснюється різними способами залежно від розміру проекту. Загальна трудомісткість невеликих проектів розраховується за формулою 6.

$$T_3 = T_H \cdot K_C \cdot K_T \cdot K_H \quad (6)$$

Показник  $\hat{E}_N$  розраховується за формулою  $K_C = 1 + \sum K_i$ .

Значення показників  $K_C$ ,  $K_T$  і  $K_H$  визначаються з додатку 2 (табл.2.4, 2.5., 2.6.)

Чисельність виконавців ( $U_B$ ) визначається за формулою 7.

$$U_B = \frac{T_3}{T_P \cdot \Phi_{ef}} \quad (7)$$

Значення показника  $T_P$  визначається залежно від строку виконання проекту в місяцях

Ефективний фонд часу роботи одного робітника ( $\Phi_{EF}$ ) розраховується за формулою 8.

$$\Phi_{EF} = D_P - D_C - D_B - D_{BC} \quad (8)$$

де:

$D_P$ - кількість днів в році;

$D_C$ - кількість святкових днів в році;

$D_B$ - кількість вихідних днів в році;

$D_{BC}$  – кількість днів відпустки.

## **Лабораторна робота № 3. Оцінка вартості розробки програмної системи за моделлю СОСОМО**

**Мета роботи:** Розрахунок вартості ПС за моделлю СОСОМО в залежності від рівній моделі, складності системи.

### **Короткі відомості**

В моделі СОСОМО використовуються три режими, за допомогою яких класифікується складність системи, а також середовище розробки.

*Органічний режим.* Органічний режим звичайно класифікується як платіжна відомість, опис або наукове обчислення. Інші характеристики режиму: невелика команда по розробці проекту, необхідні невеликі нововведення, є несуворі обмеження і кінцевий термін, а середовище розробки є стабільним.

*Напіврозділений режим.* Напіврозділений режим типізується прикладними системами, наприклад, компіляторами, системами баз даних або редакторами. Інші характеристики: невелика команда по розробці проекту середнього розміру, необхідні деякі інновації, помірні обмеження і кінцевий термін, а середовище розробки дещо нестабільне.

*Впроваджений режим.* Впроваджений режим характеризується режимами реального часу, наприклад, системами контролю повітряного руху, мережами АТМ або воєнними системами. Інші характеристики: велика команда розробників проекту, великий об'єм необхідних інновацій, жорсткі обмеження і терміни здачі. Середовище розробки в цьому випадку складається з багатьох складних інтерфейсів, включаючи ті з них, які поставляються замовникам разом з апаратним забезпеченням.

Модель СОСОМО поділяється на рівні: базовий, проміжний, деталізований. Значення драйверів витрат (додаток 3, табл.3.1)

### **Робоче завдання**

**1.** Розрахувати за базовим рівнем моделі СОСОМО трудовитрати (E) і визначити час розробки (TDEV). Визначити середню чисельність персоналу (SS) і рівень продуктивності (P), якщо:

Варіант 1. розмір проекту, який розроблюється, оцінюється в 10 KLOC.

Варіант 2. розмір проекту, який розроблюється, оцінюється в 300 KLOC.

Варіант 3. розмір проекту, який розроблюється, оцінюється в 50 KLOC.

Варіант 4. розмір проекту, який розроблюється, оцінюється в 55 KLOC.

Варіант 5. розмір проекту, який розроблюється, оцінюється в 320 KLOC.

Варіант 6. розмір проекту, який розроблюється, оцінюється в 25 KLOC.

Варіант 7. розмір проекту, який розроблюється, оцінюється в 72 KLOC.

Варіант 8. розмір проекту, який розроблюється, оцінюється в 85 KLOC.

Варіант 9. розмір проекту, який розроблюється, оцінюється в 400 KLOC.

Варіант 10. розмір проекту, який розроблюється, оцінюється в 7,5 KLOC.

## **2. Визначити режим складності системи за проміжним рівнем моделі СОСОМО, якщо:**

Варіант 1. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат ACAP, PCAP, TIME, DATA, PLEX змінюються до високих, всі інші значення номінальні.

Варіант 2. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат RELY, DATA, PVOL, PCAP, змінюються до низьких, всі інші значення номінальні.

Варіант 3. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат ACAP, CPLX змінюються до високих, TIME, DATA, PLEX змінюються до низьких, всі інші значення номінальні.

Варіант 4. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат TIME, PLEX, CPLX, змінюються до дуже високі, всі інші значення номінальні.

Варіант 5. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат TOOL, SCED змінюються до низьких, PLEX, STOR змінюються до дуже високі, всі інші значення номінальні.

Варіант 6. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат CPLX, STOR, DOCU, PCAP змінюються до дуже високі, всі інші значення номінальні.

Варіант 7. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат ACAP, APEx, PCAP, PLEX змінюються до низьких, всі інші значення номінальні.

Варіант 8. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат CPLX, SCED змінюються до дуже низькі, ACAP змінюються до низьких, всі інші значення номінальні.

Варіант 9. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат RELY, DATA, ACAP, PCAP, STOR змінюються до низькі, всі інші значення номінальні.

Варіант 10. розмір проекту за першим завданням відповідно варіанту; значення множників (драйверів) витрат SITE, TOOL змінюються до дуже низькі, SCED змінюються до низьких, всі інші значення номінальні.

**3. Оцінити трудовитрати, тривалість і середню чисельність персоналу проекту по моделі COCOMO II (для попередньої оцінки). Значення S згідно варіанту завдання 1. Показник Rj середній рівень (таблиця 3), Zi – високий рівень (таблиця 4).**

# Лабораторна робота № 3.1 Метод функціональних точок

**Мета роботи:** Навчитися розраховувати функціональні точки, пов'язані з даними і транзакціями.

## Короткі відомості

Аналіз функціональних точок — стандартний метод вимірювання розміру програмного продукту з точки зору користувачів системи.

Метод призначений для оцінки на основі логічної моделі об'єму програмного продукту кількістю функціонала, який затребуваний замовником і поставляється розробником. Безумовним плюсом методу є те, що вимірювання не залежать від технологічної платформи, на якій буде розроблюватись продукт, і він забезпечує однаковий підхід до оцінки всіх проєктів в компанії.

При аналізі методом функціональних точок потрібно виконати наступну послідовність кроків:

- Визначення типу оцінки.
- Визначення області оцінки і границь продукту.
- Підрахунок функціональних точок, пов'язаних з даними.
- Підрахунок функціональних точок, пов'язаних з транзакціями.
- Визначення сумарної кількості не вирівняних функціональних точок (UFP)
- Визначення значення фактору вирівнювання (VAF)
- Розрахунок кількості вирівняних функціональних точок (AFP)

## Робоче завдання

1. Визначити оцінки в не вирівняних функціональних точках об'єкту даних «Студент» (рис.1)



Рисунок 1. База даних студент

Складність даних визначається на основі матриці складності (табл.6)

Таблица 6

	1-19 DET	20-50 DET	50+ DET
1 RET	Низька	Низька	Середня
2-5 RET	Низька	Середня	Висока
6+ RET	Середня	Висока	Висока

В залежності від типів файлів відбувається оцінка даних (табл.7)

Таблица 7

Складність даних	Кількість UFP (ILF)	Кількість UFP (EIF)
Низька	7	5
Середня	10	7
Висока	15	10

2. Підрахунок функціональних точок, пов'язаних з транзакціями

Визначити оцінку управляючої транзакції для діалогового вікна (рис.2)

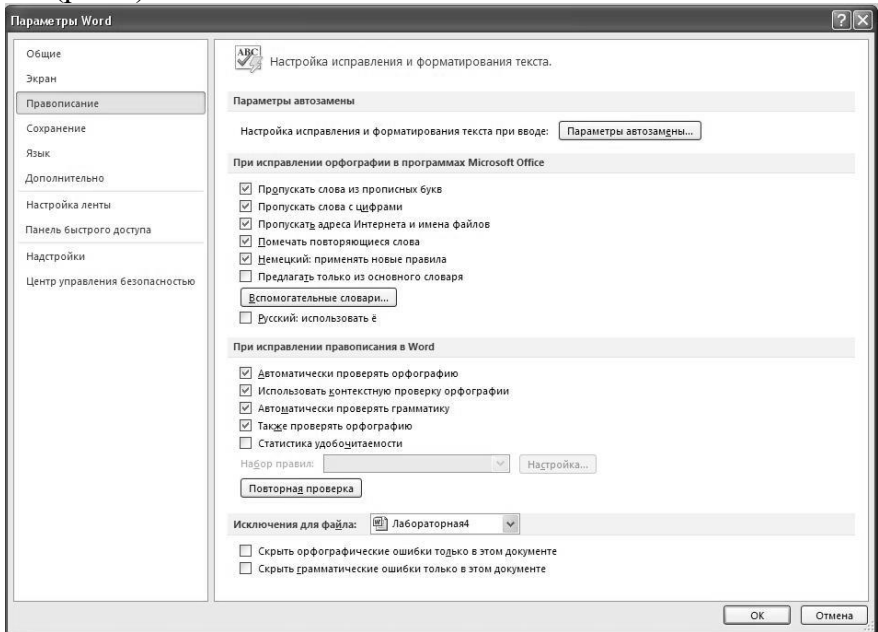


Рисунок 2. Діалогове вікно Параметри Word

Для оцінки складності транзакцій служать матриці, які представлені в таблицях 8,9.

Таблица 8

EI	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Низька	Низька	Средня
2 FTR	Низька	Средня	Висока
3+ FTR	Средня	Висока	Висока

Таблица 9

EO & EQ	1-5 DET	6-19 DET	20+ DET
---------	---------	----------	---------



0-1 FTR	Низька	Низька	Середня
2-3 FTR	Низька	Середня	Висока
4+ FTR	Середня	Висока	Висока

Оцінка транзакцій в невіривнених функціональних точках (UFP) може бути отримана з таблиці 10.

*Таблиця 10*

Складність транзакцій	Кількість UFP (EI)	Кількість UFP (EO& EQ)
Низька	3	4
Середня	4	5
Висока	6	7

## **Лабораторна робота № 4. Метод узгодженої оцінки проекту (PERT)**

**Мета роботи:** За допомогою методу PERT отримати достатньо реалістичні оцінки трудомісткості і терміну реалізації програмного проекту, швидко і без великих витрат.

### **Короткі відомості**

Реалізація методу основана на використанні власного досвіду або досвіду колег, отриманого в схожих проектах. Це найбільш прагматичний підхід, який дозволяє отримати достатньо реалістичні оцінки трудомісткості і терміну реалізації програмного проекту, швидко і без великих витрат.

Діапазон невизначеності достатньо охарактеризувати трьома оцінками:

Мі — найбільш вірогідна оцінка трудовитрат;

O<sub>i</sub> — оптимістична оцінка трудовитрат; мінімально можливі трудовитрати на реалізацію пакету робіт;

P<sub>i</sub> — песимістична оцінка трудовитрат;

Задачею проекту була розробка на основі стандартів J2EE загальносистемного ПЗ для переводу робочих місць на нову триланкову архітектуру. Був розроблений набір стандартних компонентів і сервісів, з яких як з конструктора можливо ефективно і якісно збирати прикладні підсистеми. Багаторівнева архітектура реалізувала стандартний шаблон MVC (рисунок 3), кожний з компонентів якого мав «точки розширення» для прикладної розробки, які на рисунку виділені червоним кольором.

Такими точками розширення є:

користувацький екран (UI Form), який збирався з готових візуальних компонентів;

обробники (Action), які оброблювали на сервері додатків подій від активних візуальних компонентів, які входять в склад екрану;

об'єкти (Business Obj), які моделювали прикладну область і до яких звертались обробники подій.

Новий додаток, який розроблюється, містить 20 користувацьких екранів, 60 обробників подій, 16 нових бізнес-об'єктів і 40 нових бізнес-методів, які необхідно додати, як в нові, так і в вже існуючі бізнес-об'єкти

## **Робоче завдання**

За допомогою методу PERT розрахувати:

1. Оцінку середньої трудомісткості по кожному елементарному пакету;
2. Середньоквадратичне відхилення;
3. Сумарну трудомісткість проекту;
4. Середньоквадратичне відхилення для оцінки сумарної трудомісткості;
5. Оцінки сумарної трудомісткості проекту з вірогідністю 95%.



2. Для кожної функції планувальник формує оптимістичну, песимістичну і найбільш вірогідну LOC-оцінку.
3. Для кожної функції обчислюється очікуване значення LOC-оцінки
4. Визначається значення LOC-продуктивності розробки функції
5. Обчислюється загальна оцінка витрат на проект
6. Обчислюється загальна оцінка вартості одного рядка

## Робоче завдання

Розглядаємо замовлення, яке надійшло від «УкрАВТО». Необхідно створити ПЗ для робочої станції дизайнера автомобіля. Ідентифіковані наступні основні функції ПЗ:

1. Засоби управління користувацьким інтерфейсом (КІ)
2. Аналіз двомірної графіки (2D)
3. Аналіз трьохмірної графіки (3D)
4. Управління базою даних (БД)
5. Засоби комп'ютерної дисплейної графіки (КГ)
6. Управління периферією (УП)
7. Модулі проектного аналізу (МП)

1. Заповнити Очікувану оцінку LOC в наступній таблиці:

Таблиця 11. Початкова таблиця оцінки проекту

Функція	Оптимістична (LOC)	Найбільш вірогідна (LOC)	Песимістична (LOC)	Очікувана (LOC)
КІ	1800	2400	2650	
2D	4100	5200	7400	
3D	4600	6900	8600	
БД	2950	3400	3600	
КГ	4050	4900	6200	
УП	2000	2100	2450	
МП	6600	8500	9800	
Всього:				

2. Визначити продуктивність

Таблиця 12. Дані з метричного базису фірми

Функція	LOC <sub>ан</sub>	Продуктивність <sub>ан</sub> (LOC/люд.-міс.)	Продуктивність (LOC/люд.-міс.)
КІ	585	1260	
2D	3000	440	
3D	3000	440	
БД	1117	720	
КГ	2475	400	
УП	214	1400	
МП	1400	1800	

## 3. Розрахувати вартість і витрати

Таблиця 13. Кінцева таблиця оцінки проекту

Функція	Очікувана (LOC)	П_Вартість <sub>ан</sub> (\$/LOC)	Вартість (\$)	Витрати
КІ		14		
2D		20		
3D		20		
БД		18		
КГ		22		
УП		28		
МП		18		
Всього				

Перевірити розрахунки за допомогою FP-показників, всі інформаційні характеристики мають середній рівень складності.

## 1. Знайти очікувану оцінку і кількість

Таблиця 14. Оцінка інформаційних характеристик проекту

Функція	Опт-на	Найб. вірогідна	Песим-на	Очікувана	Складність	Кількість
Введення	20	24	30		x4	
Виведення	12	15	22		x5	
Запити	16	22	28		x4	
Логічні файли	4	4	5		x10	
Інтерфейсні файли	2	2	3		x7	
Загальна кількість						

Кожний коефіцієнт може приймати наступні значення: 0 — нема впливу, 1 — випадкове, 2 — невелике, 3 — середнє, 4 — важливе, 5 — основне.

2. Визначити FP згідно значень з таблиць 14,15.
3. Продуктивність, витрати і вартість.
4. Перевірити достовірність результатів.

Таблиця 15. Оцінка системних параметрів проекту

Коефіцієнт регулювання складності		Оцінка
F1	Передачі даних	2
F2	Розподілена обробка даних	0
F3	Продуктивність	4
F4	Поширеність використовуваної конфігурації	3
F5	Швидкість транзакцій	4
F6	Оперативний ввід даних	5
F7	Ефективність роботи кінцевого користувача	5
F8	Оперативне оновлення	3
F9	Складність обробки	5
F10	Повторне використання	4
F11	Легкість інсталяції	3
F12	Легкість експлуатації	4
F13	Різноманітні умови розміщення	5
F14	Простота вимірювань	5
Всього		52

## Лабораторна робота № 5. Засоби оцінки вартості програмного забезпечення

**Мета роботи:** За допомогою програми SoftStar Systems Costar розрахувати вартість ПЗ на основі власних параметрів.

### Короткі відомості

**Параметри вартості.** Параметр вартості (cost driver) – це суб’єктивна величина, яка оцінює різні часові, якісні і ресурсні аспекти розробки ПЗ. Кожний з параметрів може бути

відкаліброваним. Калібрування параметрів вартості – це корегування значень параметрів, яка впливає на значення трудовитрат, і відповідно на час і вартість, при оцінці програмного проекту. При калібруванні вказаних далі сімнадцяти параметрів вибирається оціночний рівень (дуже високий, високий, вище номінального, номінальний, нижче номінального, низький, дуже низький) параметру. В формулах цей рівень відображується у вигляді коефіцієнту трудовитрат і, таким чином, на кожній стадії розробки проекту впливає на вартість і тривалість той або іншої стадії. Виділяють наступні групи параметрів (табл.16): продукту (product factors), платформи (platform factors), персоналу (personnel factors) і проекту (project factors). В табл. 17 наданий короткий опис кожного параметру.

*Таблиця 16*

Параметри	Опис
Продукту	Враховують характеристики ПЗ, кий розроблюється. (RELY, DATA, CPLX, RUSE, DOCU)
Платформи	Враховують характеристики програмно-апаратного комплексу, який необхідний для функціонування ПЗ. (TIME, STOR, PVOL)
Персоналу	Враховують рівень знань і злагодженості роботи колективу програмістів. (ACAP, PCAP, PCON, APEX, PLEX, LTEX)
Проекту	Враховують вплив сучасних підходів і технологій, територіальної віддаленості членів колективу розробників і терміни виконання проекту. (TOOL, SITE, SCED)

*Таблиця 17*

Параметри	Опис
RELY (Required Software Reliability)	Враховує міру виконання програмою задуманої дії на протязі певного часу
DATA (Database Size)	Враховує вплив об'єму тестових даних на розробку продукту. Рівень цього параметру розраховується як співвідношення байт в тестованій базі даних до SLOC в програмі
CPLX (Product Complexity)	Включає п'ять типів операцій: управління, рахунків, пристрій-залежні, управління даними, управління користувацьким інтерфейсом. Рівень складності це суб'єктивне середньо-зважене значення рівнів типів

	операцій
RUSE (Developed for Reusability)	Враховує трудовитрати, необхідні додатково для написання компонентів, призначених для повторного використання в даному або наступних проектах. Використовує наступні оціночні рівні: “в проєкті”, “в програмі”, “в лінійці продуктів”, “в різних лінійках продуктів”. Значення параметру накладає обмеження на наступні параметри: RELY и DOCU
DOCU (Documentation Match To Life-Cycle Needs )	Враховує ступінь відповідності документації проєкту його життєвому циклу
TIME (Execution Time Constraint)	Враховує часові ресурси, які використовуються ПЗ, при виконанні поставленої задачі
STOR (Main Storage Constraint)	Враховує процент використання сховищ даних
PVOL (Platform Volatility)	Враховує термін життя платформи (комплекс апаратного і програмного забезпечення, який необхідний для функціонування ПЗ, який розроблюється)
ACAP (Analyst Capability)	Враховує аналіз, здатність проєктувати, ефективність і комунікативні здібності групи спеціалістів, які розроблюють вимоги і специфікації проєкту. Параметр не повинен оцінювати рівень кваліфікації окремо взятого спеціаліста
PCAP (Programmer Capability)	Враховує рівень програмістів в колективі. При виборі значення для цього параметру слід особливо звернути увагу на комунікативні і професійні здібності програмістів і на командну роботу в цілому
PCON (Personnel Continuity)	Враховує плинність кадрів в колективі
APEX (Applications Experience)	Враховує досвід колективу при роботі над додатками певного типу
PLEX (Platform Experience)	Враховує вміння використовувати особливості платформ, такі як графічний інтерфейс, бази даних, сітьовий інтерфейс, розподілені системи
LTEX (Language and Tool Experience)	Враховує досвід програмістів (мови, середовище і інструменти)
TOOL (Use Of Software Tools)	Враховує рівень використання інструментів розробки
SITE (Multisite Development)	Враховує територіальну віддаленість (від офісу до міжнародних офісів) членів команди розробників засоби комунікації, які ними використовуються (від телефону до відео конференц-зв'язку)
SCED (Required Development Schedule)	Враховує вплив часових обмежень, накладених на проєкт і на значення трудовитрат



## Робоче завдання

Запустити програму Costar 7.0 Demo

Ввести в програму свої параметри (вводити параметри згідно свого варіанту з лабораторної 2, завдання 2):

1 крок – вибрати модель СОСОМО II – ранню розробку проекту чи постархітектурну

2 крок – ввести кількість рядків вихідного коду (SLOC)

3 крок – вибрати фактори масштабу (5 характеристик)

4 крок – вибрати Параметри вартості (cost driver) – (17 характеристик)

5 крок - Отримати результат.

**Estimate1 - Detail Report**

Costar 7.0 Demo      22.03.2013    15:42:34      Page: 1

Estimate Name:	Estimate1	Estimate ID:	
Model Name:	COCOMO II 2000	Model ID:	2000
Process Model:	COCOMO II Model	Phases:	Waterfall
Component Name:	Component1	Component ID:	
Increment:	1	Level:	1
Developed Size:	1,300	EAF:	1.0000

Phase	Effort (Person-Months)	Cost (K\$)	Duration (Months)	Staffing
RQ -- Requirements	0.3	0.0	0.9	0.3
PD -- Product Design	0.7	0.0	1.4	0.5
DD -- Detailed Design	1.1	0.0	1.4	0.8
CT -- Code & Unit Test	1.5	0.0	1.8	0.8
IT -- Integration & Test	0.7	0.0	1.1	0.7
Development (PD+DD+CT+IT)	3.9	0.0	5.7	
Totals (RQ+PD+DD+CT+IT)	4.2	0.0	6.6	
MN -- Maintenance (per year)	0.0	0.0		0.0

Drivers & Size / Model / REVL / Reuse / Function Points / Increments / Breakage / Costs / Rates / Maint / Filter / Descr.

Estimate1: 4.2 PM, 6.6 Months      Component1: 4.2 PM      EAF: 1.0000    Level: 1

Рисунок 4. Детальний звіт

Після появи вікна з результатами (рис.4) – перенести значення, які у червоному обрамлені, в закладку Costs (рис 5)

Costar - Estimate1 (Component1)

File View Reports Components Tools Preferences Help

Estimate: Estimate1 ID: Model: COCOMO II 2000

Component: Component1 ID: Increment: 1

ACT ARC CBR CDR CMP CST DET EBR EFF EGS GCS GMI GST IDT ISM MSZ NAM SCH SZ SSM STR

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements	RQ:	0.3	0.9	0.1		Total Size: 1,300
Development	PD+DD+CT+IT:	3.9	5.7	4.3	331.4	
Total	RQ+PD+DD+CT+IT:	4.2	6.6	4.4	309.7	

Costs for Component: Component1

Cost per Person-Month			
Requirements	\$ 300	<input type="checkbox"/> Inherit RQ	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Product Design	\$ 700	<input type="checkbox"/> Inherit PD	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Detailed Design	\$ 1100	<input type="checkbox"/> Inherit DD	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Code & Unit Test	\$ 1500	<input type="checkbox"/> Inherit CT	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Integration & Test	\$ 700	<input type="checkbox"/> Inherit IT	<input type="checkbox"/> Use Rates Tab & Labor Distribution
Maintenance	\$ 0	<input checked="" type="checkbox"/> Inherit MN	<input type="checkbox"/> Use Rates Tab & Labor Distribution

Drivers & Size / Model / REVL / Reuse / Function Points / Increments / Breakage / Costs / Rates / Maint / Filter / Descr /

Estimate1: 4.2 PM, 6.6 Months Component1: 4.2 PM EAF: 1.0000 Level: 1

Рисунок 5. Введення витрат

Вивести вікно результату, де будуть виведені зусилля 1 людино-місяця, строк розробки, вартість.

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

PREC — прецедентність, наявність досвіду аналогічних розробок (Very Low — досвід в продукті і платформі відсутній; Extra High — продукт и платформа повністю знайомі)

FLEX — гнучкість процесу розробки (Very Low — процес строго детермінований; Extra High — визначені тільки загальні цілі).

RESL — архітектура і дозвіл ризиків (Very Low — ризики невідомі/не проаналізовані; Extra High — ризики дозволені на 100%)

TEAM — спрацьованість команди (Very Low — формальна взаємодія; Extra High — повна довіра, взаємозамінність і взаємодопомога).

PMAT — зрілість процесів (Very Low — CMM Level 1; Extra High — CMM Level 5)

Значення фактора масштабу, в залежності від оцінки його рівня відображено в таблиці 13.

*Таблиця 18*

	<b>Very Low</b>	<b>Low</b>	<b>Nominal</b>	<b>High</b>	<b>Very High</b>	<b>Extra High</b>
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

## Додаток 1

Таблиця 1.1

Мова програмування	КП (кількість логічних рядків коду на одну функціональну точку)
1	2
Basic Assembler	320
Autocoder	320
Netron/CAP	296
Macro Assembler	213
C	128
Пакетні файли DOS	128
Basic	107
Макроси Lotus	107
ALGOL	105
COBOL	105
FORTTRAN	105
JOVIAL	105
Змішані мови програмування	105
JCL	96
VPF	95
Pascal	91
COBOL (ANSI 85)	91
APS	86
Slogan	81
RPG	80
Modula-2	80
PL/1	80
Паралельний Pascal	80
Fortran 95	71
Mantis	71
Sabretalk	70
Mapper	69
ColdFusion	68

Datastage	67
Ideal	66
Basic (ANSI)	64
FORTH	64
LISP	64
PROLOG	64
Powerhouse	63
Uniface	61
.NET	60
JSP	59
LOGO	58
C#	58
J2EE	57
Розширених LISP	56
RPG III	56
ASP	56
Java	55
JavaScript	54
C++	53
YACC	53
Culprit	51
Natural	51
KML	50
Visual Basic	50
REXX	50
Ada 95	49
PL/SQL	47
CICS	46
SIMULA	46
Taskmate	45
Focus	45
Web Scripts	44
Pacbase	42
Мови баз даних	40
Clipper DB і dBase III	40
Informix	40

Oracle и SYBASE	40
Openroad	39
Access	38
VBScript	38
Advantage	38
PeopleSoft	37
Cool:Gen/IEF	37
DBase IV	36
Мови підтримки прийняття рішень	35
FoxPro 2.5	34
APL	32
Статичні мови (SAS)	32
Maestro	30
DELPHI	29
Стандартні об'єктно-орієнтовані мови	29
Powerbuilder	28
VB.Net	28
OBJECTIVE-C	27
Lotus Script	23
Oracle Developer /2000	23
Smalltalk	21
awk	21
EIFFEL	21
Shell-сценарії (Perl)	21
Стандартні мови 4-го покоління (4GL)	20
OR3 (4GL)	20
Application Builder	20
CORBA	20
Cristal Reports	20
Datatrieve	20
CLIPPER	19
ABAP (SAP)	18
HTML 3.0	15

Siebel Tools	13
SQL	13
Easytrieve+	13
SQL Forms	11
Excel	6
QUATTRO PRO	6
Мови створення піктограм	4

## Додаток 2

Таблиця 2.1

### Каталог функції програмного забезпечення

№ з/п		Обсяг функцій (рядків вихідного коду) використанням середовища розробки додатків		
		(Borland)	C++ Builder (Borland)	Visual C++ (Microsoft)
1	2	3	4	5
1. Введення, аналіз вхідної інформації, генерація кодів і процесор вхідної мови				
101	Організація введення інформації	100	110	150
102	Контроль, попередня проробка і введення інформації	520	550	450
103	Аналіз вхідної мови (синтаксичний і семантичний)	630	660	660
104	Перетворення операторів вхідної мови і команди іншої мови	1050	1050	980
105	Обробка вхідного замовлення і формування таблиць	750	900	1340
106	Перетворення вхідної мови в машинні команди (транслятор, препроцесор, макрогеніратор)	4300	4300	5100
107	Синтаксичний і семантичний аналіз вхідної мови і генерація кодів команд	8700	5400	5400



108	Процесор мови	3000	2300	2300
109	Організація введення/висновку інформації в інтерактивному режимі	220	220	320
110	Організація введення/висновку інформації з мережі терміналів	3680	3340	3200
111	Керування введенням/висновком	2700	2900	2400
2. Формування, введення і обслуговування баз даних				
201	Генерація структури бази даних	3450	3950	4300
202	Генерація підсхеми бази даних	1540	1610	2060
203	Формування баз даних	1700	1750	2180
204	Обробка наборів і записів бази даних	2050	2350	2670
205	Обслуговування бази даних у пакетному режимі	1030	1100	1260
206	Обслуговування бази даних в інтерактивному режимі	3800	4400	6950
207	Маніпулювання даними	8400	8670	9550
209	Реорганізація бази даних	130	190	220
210	Завантаження бази даних	3150	2950	2780
3. Формування і обробка файлів				
301	Формування послідовного файлу	340	360	290
302	Сортування файлу	340	360	290
303	Автоматичне сортування файлів	1040	1150	930
304	Обслуговування файлів	520	540	420

305	Обробка файлів	750	800	720
306	Обробка файлів у діалоговому режимі	2400	2600	3050
307	Спільна обробка групи файлів	4900	5300	6180
308	Керування файлами	5130	5380	5750
309	Формування файлу	1100	1080	1020
4. Генерація програм і ПЗ, а також настроювання ПЗ				
401	Генерація робочих програм	3680	4120	3360
402	Генерація програм за опису користувачів	10870	12330	9880
403	Формування службових таблиць	570	620	1070
404	Система генерації ПЗ	5120	5340	4980
405	Система настроювання ПЗ	250	300	370
5. Керування ПЗ, компонентами ПЗ і зовнішніми пристроям				
501	Монітор ПЗ (керування роботою компонентів)	350	360	740
502	Монітор системи (керування роботою комплексу ПЗ)	3750	3880	7740
503	Керування зовнішніми пристроями і об'єктами	6850	7340	5900
504	Обробка переривань	890	730	540
505	Керування зовнішньою пам'яттю	250	210	200
506	Обробка помилкових і збійних ситуацій	430	410	410
507	Забезпечення інтерфейсу між компонентами	730	750	970
6. Відлагодження прикладних програм, обмін інформацією між МД і МС, допоміжні програмні функції				

601	Налагодження прикладних програм в інтерактивному режимі	4500	4700	4300
602	Довідка і навчання	680	680	720
603	Допоміжні і сервісні програми	460	490	580
7. Розрахункові завдання, формування і висновок на зовнішні носії (АЦПУ) документів складної форми і файлів				
701	Математична статистика і прогнозування	8370	9570	9320
703	Розрахунок показників	410	500	460
704	Процесор звітів	1070	1230	3200
705	Формування і висновок на зовнішні носії	2650	2850	3500
706	Попередня обробка й печатка файлів	540	560	470
707	Графічний висновок результатів	300	310	480
708	Інтерактивний редактор тексту	3800	3910	4540
709	Вимір стану ресурсів в інтерактивній системі	650	440	480

Таблиця 2.2

Норми часу на розробку ПЗ (Тн) залежно від обсягу ПЗ ( $V_3$ ) і групи складності ПЗ (люд./дн.)

Об'єм ПЗ (рядка вихідного коду, LOC)	Категорії складності ПЗ			Номер норми
	1-ша	2-га	3-я	
1	2	3	4	5
200	-	-	21	1
300	-	-	23	2
400	-	-	25	3
500	-	-	27	4
600	-	33	28	5
800	-	38	32	7
900		40	34	8
1000	51	43	36	9
1200	54	45	38	10
1400	57	48	40	11
1600	60	50	42	12
1800	64	54	45	13
2000	68	57	48	14
2200	73	61	51	15
2400	76	64	54	16
2600	81	68	57	17
2800	86	72	60	18
3000	91	76	64	19
3200	97	81	68	20
3400	103	86	72	21
3600	ПЗ	92	77	22
3800	117	98	82	23
4000	124	104	87	24

4200	133	111	93	25
4400	141	118	99	26
4600	151	126	105	27
4800	160	134	112	28
5000	170	142	119	29
5500	182	152	127	30
6000	194	162	135	31
6500	206	172	144	32
7000	220	184	154	33
8000	252	210	175	35
8500	268	224	187	36
9000	288	240	200	37
9500	307	256	214	38
10000	327	273	228	39
11000	349	291	243	40
12000	374	312	260	41
13000	399	333	278	42
14000	427	356	297	43
15000	456	380	317	44
16000	487	406	339	45
18000	520	434	362	46
20000	556	464	387	47
22000	595	496	414	48
24000	636	530	442	49
26000	679	566	472	50
28000	727	606	505	51

30000	775	646	540	52
32000	830	692	577	53
34000	888	740	617	54
36000	950	792	660	55
38000	1016	847	706	56
40000	1087	906	755	57
42000	1161	968	807	58
44000	1242	1035	863	59
46000	1328	1107	923	60
48000	1420	1184	987	61
50000	1620	1267	1056	62

Таблиця 2.3.

Категорії складності	Характеристики ПЗ
1	<p>ПЗ, що обкладають однієї або декількома з наступних характеристик:</p> <ol style="list-style-type: none"> <li>1. Наявність складного інтелектуального мовного інтерфейсу з користувачем</li> <li>2. Режим роботи в реальному часі.</li> <li>3. Забезпечення телекомунікаційної обробки даних і керування знищуваними об'єктами.</li> <li>4. Машина графіка.</li> <li>5. Багатомашинні комплекси.</li> <li>6. Забезпечення істотного розпаралелювання обчислень.</li> </ol>
2	<p>ПЗ, що обкладають однієї або декількома з наступних характеристик:</p> <ol style="list-style-type: none"> <li>1. Оптимізаційні розрахунки.</li> <li>2. Забезпечення настроювання ПЗ на зміни структур вхідних і вихідних даних.</li> <li>3. Настроювання ПЗ на нестандартну конфігурацію технічних коштів.</li> <li>4. Забезпечення переносимості ПЗ.</li> <li>5. Реалізація особливо складних інженерних і наукових розрахунків.</li> </ol>
3	<p>ПЗ, що не володіють перерахованими вище характеристиками.</p>

Таблиця 2.4.  
Додаткові коефіцієнти складності ПЗ

Характеристика ПЗ	Значення Кс
1. Функціонування ПЗ в розширеному операційному середовищі (зв'язок з іншими ПЗ).	0,08
2. Інтерактивний доступ.	0,06
3. Забезпечення зберігання, ведення й пошуку даних у складних структурах.	0,07
4. Наявність у ПЗ одночасно декількох характеристик по табл. 2.4 :	
4.1. 2 характеристики.	0,12
4.2. 3 характеристики.	0,18
4.3. Понад 3-х характеристики.	0,26

Таблиця 2.5.  
Значення поправочного коефіцієнта, що враховує використання стандартних модулів типових програм і ПЗ (Кт)

Ступінь охоплення реалізованих функцій розроблювального ПЗ стандартними модулями, типовими програмами і ПЗ	Значення Кт
1. Від 60% і вище.	0,6
2. Від 40% до 60%.	0,7
3. Від 20% до 40%.	0,8
4. До 20%.	0,9
5. Типові програми й ПЗ не використовуються для реалізації функцій розроблювального ПЗ.	1,0



Таблиця 2.6.

Поправочні коефіцієнти, що враховують новизну ПЗ (Кн)

Категорія новизни ПЗ	Ступінь новизни	Використання		Значення Кн
		На основі нового типу ПК	У середовищі нової ОС	
А	Принципово нові ПЗ, що не мають доступних аналогів.	+	+	1,75
		-	+	1,6
		+	-	1,2
Б	ПЗ, що є розвитком визначеного параметричного ряду ПЗ.	-	-	1,0
		+	+	1,0
		-	-	0,9
		+	-	0,3
В	ПЗ, що є розвитком визначеного параметричного ряду ПЗ, розроблених для раніше освоєних типів конфігурації ПК й ОС.	-	-	0,7

## Додаток 3

### Таблиця 3.1

Значення драйверів витрат при розробці ПЗ в рамках моделі  
COCOMO

Параметри вартості	Показники					
	Дуже низький	Низький	Номіна- льний	Висо- кий	Дуже високий	Над- високий
Атрибути продукту						
Необхідна надійність ПЗ (RELY)	0,82	0,92	1,00	1,10	1,26	
Розмір бази даних (DATA)		0,90	1,00	1,14	1,28	
Складність програмного продукту (CPLX)	0,73	0,87	1,00	1,17	1,34	1,74
Необхідний рівень повторного використання (RUSE)		0,95	1,00	1,0,7	1,15	1,24
Документація (DOCU)	0,81	0,91	1,00	1,11	1,23	
Атрибути персоналу						
Здібності	1,42	1,19	1,00	0,85	0,71	

аналітика (АСАР)						
Досвід створення додатків (APEX)	1,22	1,10	1,00	0,88	0,81	
Здібності програміста (PCAP)	1,34	1,15	1,00	0,88	0,70	
Досвід в області віртуальних машин (PLEX)	1,19	1,09	1	0,91	0,85	
Досвід в області мов програмування (LTEX)	1,20	1,09	1	0,91	0,84	
Наступність персонала (PCON)	1,29	1,12	1	0,90	0,81	
Атрибути проекту						
Використання практики сучасного програмування (SITE)	1,22	1,09	1,00	0,93	0,86	0,80

Сучасні інструменти програмування (TOOL)	1,17	1,09	1,00	0,90	0,78	
Необхідний графік розробки (SCED)	1,43	1,14	1,00	1,00	1,00	
Атрибути платформи						
Обмеження часу виконання (TIME)			1,00	1,11	1,29	1,63
Обмеження головного сховища (STOR)			1,00	1,05	1,17	1,46
Змінність платформи (PVOL)		0,87	1,00	1,15	1,30	

## Контрольні питання:

1. Що таке LOC-оцінки?
2. Що відносять до неалгоритмічних методів оцінки вартості ПЗ?
3. Що відносять до неалгоритмічних методів оцінки вартості ПЗ?
4. На які групи прийнято розділяти метрики складності програм?
5. На чому базуються метрики розміру програм?
6. На чому базуються метрики складності потоку управління програм?
7. На чому базуються метрики складності потоку даних?
8. На яких принципах ґрунтується методика аналізу Функціональних точок?
9. При підрахунку функціональних точок, пов'язаних з даними, що таке DET?
10. При підрахунку функціональних точок, пов'язаних з даними, що таке RET?
11. На яких характеристиках ґрунтується оцінка складності транзакції при підрахунку функціональних точок?
12. Які режими використовуються в моделі COSOMO II?
13. Які рівні моделі COSOMO існують і які вони?
14. За якою формулою розраховується оцінка трудовитрат в базовому рівні COSOMO II?
15. За якою формулою розраховується оцінка трудовитрат в проміжному рівні COSOMO?
16. Що таке фактор EAF в моделі COSOMO II?
17. Що таке метод узгодженої оцінки?
18. Яким чином можна оцінити проект на основі LOC і FP-метрик?
19. За якими категоріями групуються драйвери (множники) витрат?

## Література

1. Орлов С.А. Технологии разработки программного обеспечения: - — СПб.: Питер, 2002. - 464 с.
2. Липаев В.В. Экономика производства сложных программных продуктов. – М.: Синтег, 2008. - 432 с.
3. Р.Т. Фатрелл, Д.Ф. Шафер, Л.И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. М.: Издательский дом “Вильямс”, 2004. – 1125 с.
4. Ройс У. Управление проектами по созданию программного обеспечения: Пер. с англ. – М.: ЛОРИ, 2002. -424 с.
5. Василенко Ю.Н. Алгоритмические методы оценки программного обеспечения. – Мат. конф. "Інженерія програмного забезпечення 2005". – НАУ. – 2005. – 42-51 с.
6. Баценко Д.В. Классификация параметров стоимости модели постархитектуры. – Мат. конф. "Інженерія програмного забезпечення 2005". – НАУ. – 2005. – 51-56 с.
7. Щебетин Ю.В. Методика анализа функциональных точек программного обеспечения. – Мат. конф. "Інженерія програмного забезпечення 2005". – НАУ. – 2005. – 56-62 с.
8. Сидоров Н.А., Баценко Д.В., Василенко Ю.Н., Щебетин Ю.В., Иванова Л.Н. Методы и средства оценки стоимости программного обеспечения. "Проблеми системного підходу в економіці". – НАУ. – 2004. – № 7 – 113-118 с.
9. Сидоров Н.А., Лидер Д.А., Хальдун Д. Deskриптивная модель дистанционного завершения высшего образования в области информационных технологий. //Проблеми системного підходу в економіці. – НАУ. – 2003. – № 5 – 6-10 с.
10. Колдовский В. Разработка ПО: метрики программных проектов. [Електронний ресурс].  
[http://itc.ua/articles/razrabotka\\_po\\_metriki\\_programmnyh\\_proektov\\_27774](http://itc.ua/articles/razrabotka_po_metriki_programmnyh_proektov_27774)

11. *Изосимов А.В., Рыжко А.Л.* Метрическая оценка качества программ. - М.: МАИ. - 1989. – 450 с.
12. Методика расчетов трудоемкости и стоимости работ на разработку информационных систем . АО «НИТ». 20с. (в слайдах)
13. Электронный ресурс:  
[http://citforum.ru/SE/project/arkhipenkov\\_lectures/](http://citforum.ru/SE/project/arkhipenkov_lectures/)

## **Зміст**

<b>Передмова .....</b>	<b>3</b>
<b>Лабораторна робота № 1.1. Розрахунок трудомісткості розробки програмного продукту.....</b>	<b>4</b>
<b>Лабораторна робота № 2. Розрахунок чисельності виконавців проекту, строки виконання роботи. ....</b>	<b>8</b>
<b>Лабораторна робота № 3. Оцінка вартості розробки програмної системи за моделлю COCOMO .....</b>	<b>11</b>
<b>Лабораторна робота № 3.1 Метод функціональних точок .....</b>	<b>14</b>
<b>Лабораторна робота № 4. Метод узгодженої оцінки проекту (PERT) .....</b>	<b>17</b>
<b>Лабораторна робота № 4.1. Виконання оцінки проекту на основі LOC і FP метрик.....</b>	<b>19</b>
<b>Лабораторна робота № 5. Засоби оцінки вартості програмного забезпечення .....</b>	<b>22</b>
<b>Додаток 1.....</b>	<b>28</b>
<b>Додаток 2.....</b>	<b>32</b>
<b>Додаток 3.....</b>	<b>42</b>
<b>Контрольні питання:.....</b>	<b>45</b>
<b>Література.....</b>	<b>46</b>
<b>Зміст .....</b>	<b>48</b>