

# Лекція 6

Тема: Оцінка проектів на основі  
варіантів використання  
(Use Case Points)

# Що таке UCP?

- ❑ UCP (Use Case Points) – це методика оцінки проектів **на основі варіантів використання** (use cases) системи, яка оцінюється;
- ❑ В основі UCP лежить методика Function points (оцінка на основі функціональних точок системи), але вона значно спрощена для використання не експертами Function points;
- ❑ На відміну від Function points, UCP враховує нефункціональні вимоги, організаційні ризики, компетенцію при оцінці та інші критерії.

# Етапи оцінки

**Оцінка акторів**

**Незкоригована оцінка варіантів  
використання**

**Оцінка технічних факторів**

**Оцінка зовнішніх факторів**

**Остаточний підрахунок**

# Оцінка акторів

Надає нам оцінку складності інтерфейсів системи.

Всі діючі особи системи діляться на три типи: **прості, середні і складні.**

**Простий актор** – використовує систему по перерозподіленому API.

**Середній** – використовує більш складний або гнучкий API.

**Складний** у більшості випадків означає взаємодію з кінцевим користувачем.

Тип	Ваговий коефіцієнт
Простий	1
Середній	2
Складний	3
Тип	Представляє
Простий	Зовнішня система з чітко визначеним програмним інтерфейсом.
Середній	Зовнішня система, яка взаємодіє з даною системою за допомогою протоколу, або використовує текстовий інтерфейс (наприклад, алфавітно-цифровим терміналом).
Складний	Графічний інтерфейс користувача

# Оцінка акторів

Надає нам оцінку масштабу системи.

Кожний варіант використання рангується в залежності від кількості транзакцій.

Альтернатива підрахунку за допомогою:

1. Класів (табл.2)
2. Об'єктів в базі даних (табл.3).

Узагальний ваговий показник UUCP розраховується:

$$UUCP = A + UC$$

Тип	Опис	Ваговий коефіцієнт
Простий	До 4 транзакцій	5
Середній	Від 4 до 7 транзакцій	10
Тип	Опис	Ваговий коефіцієнт
Простий	До 5 класів	5
Середній	Від 5 до 10 класів	10
Складний	Більше 10 класів	15
Тип	Опис	
Простий	1 об'єкт	
Середній	2 об'єкт	
Складний	3 і більше об'єктів	

# Оцінка акторів

Підрахована кількість діючих осіб кожного типу  $n_i$  множиться на відповідний ваговий коефіцієнт  $k_{ai}$ , потім обчислюється загальний ваговий показник  $A$ .

$$A = \sum n_i \cdot k_{ai}$$

Підрахована кількість варіантів використання кожного типу  $m_i$  множиться на відповідний ваговий коефіцієнт  $k_{bi}$ , потім обчислюється загальний ваговий показник  $UC$

$$UC = \sum m_i \cdot k_{bi}$$

# Оцінка технічних факторів

Надає нам коефіцієнт для оцінки складності архітектури додатку.

Кожному показнику присвоюється значення  $T_i$  в діапазоні від 0 до 5 (0 означає відсутність значимості показника для даного проекту, 5 високу значимість).

Значення TCF обчислюється за формулою:

$$TCF = 0,6 + (0,01 \cdot (\sum T_i \cdot Вага_i))$$

Деякі технічні фактори, які використовуються для оцінки (0-5):

❑ Розподіленість системи. Чи повинен додаток підтримувати кластера,  $n$  - рівневу архітектуру. Чим складніша архітектура, тим вище оцінка.

❑ Час відгука. Чим вище очікування до навантаження системи, тим вище оцінка.

# Оцінка технічних факторів

## Показники технічної складності

Показник	Опис	Вага
T1	Розподілена система	2
T2	Висока продуктивність (пропускна здібність )	1
T3	Робота кінцевих користувачів в режимі онлайн	1
T4	Складна обробка даних	1
T5	Повторне використання коду	1
T6	Простота установки	0,5
T7	Простота використання	0,5
T8	Переносимість	2
T9	Простота внесення змін	1
T10	Паралелізм	1
T11	Спеціальні вимоги до безпеки	1



# Оцінка технічних факторів

## Показники технічної складності

Показник	Опис	Вага
T12	Безпосередній доступ до системи зі сторони зовнішніх користувачів	1
T13	Спеціальні вимоги до навчання користувачів	1

# Оцінка зовнішніх факторів

Дає нам коефіцієнт для організаційних ризиків при розробці.

Показник	Опис	Вага
F1	Знайомство з технологією	1,5
F2	Досвід роботи з додатком	0,5
F3	Досвід використання об'єктно-орієнтованого підходу	1
F4	Наявність ведучого аналітика	0,5
F5	Мотивація	1
F6	Стабільність вимог	2
F7	Часткова зайнятість	-1
F8	Складні мови програмування	2

# Оцінка зовнішніх факторів

Кожному показнику  $F_i$  присвоюється значення в діапазоні від 0 до 5. Для показників  $F_1$ – $F_4$  0 означає відсутність, 3 – середній рівень, 5 – високий рівень. Для показника  $F_5$  0 означає відсутність мотивації, 3 – середній рівень, 5 – високий рівень мотивації. Для  $F_6$  0 означає високу нестабільність вимог, 3 – середню, 5 – стабільні вимоги. Для  $F_7$  0 означає відсутність спеціалістів з частковою зайнятістю, 3 – середній рівень, 5 – всі спеціалісти з частковою зайнятістю. Для показника  $F_8$  0 означає просту мову програмування, 3 – середню складність, 5 – високу складність.

Значення  $EF$  обчислюється за формулою:

$$EF = 1,4 + (-0,03 \cdot (\sum F_i \cdot Вага_i))$$

# Оцінка трудоемності проекту

З врахуванням показників технічної складності (ТСФ ) і рівня кваліфікації розробників (EF) остаточне значення показника варіантів використання UCP (use case points) визначається за формулою:

$$UCP = UUCP \cdot TCF \cdot EF$$

Перерахунок величини UCP в людино-години потребує визначення кількості людино-годин, які припадають на одну UCP (коефіцієнт  $k_{UCP}$ ). Цей показник залежить від прийнятої на підприємстві методики визначення варіантів використання (яким критеріям повинен задовольняти окремо виділений варіант використання), кваліфікації розробників, які використовують інструментальні засоби, накопиченого на підприємстві досвіду виробництва ПЗ. Його значення є стандартом даного підприємства.

# Оцінка трудоемності проекту

Потрібно розглянути показники F1 – F8 і визначити, скільки показників F1 – F6 мають значення менше 3 і скільки показників F7, F8 мають значення більше 3.

Якщо загальна кількість менша або дорівнює 2, слідє використувувати 20 люд.-г. на одну UCP, якщо 3 або 4 – 28. Якщо загальна кількість дорівнює 5 або більше, слід внести зміни в сам проект, в протилежному випадку ризик провалу дуже високий.

$$T = UCP * \text{кількість люд.-год}$$

$$t_{\text{роз}} = 2,5 \times T^{N3}$$

# Модель композиції додатку

Модель композиції є однією з конструктивних моделей вартості СОСОМО II.

Параметри даної моделі визначались на основі статистичного аналізу реальних результатів великої кількості проектів.

Модель композиції використовується на ранній стадії розробки ПЗ, коли:

- Розглядається макетування користувацьких інтерфейсів;
- Обговорюється взаємодія ПЗ і комп'ютерної системи;
- Оцінюється продуктивність;
- Визначається ступінь зрілості технології.

Модель композиції додатку орієнтована на застосуванні об'єктних вказівників.

Об'єктний вказівник – засіб непрямого виміру ПЗ, для його розрахунку визначається кількість екранів (як елементів користувацького інтерфейсу), звітів і компонентів, які необхідні для побудови додатку.

# Модель композиції додатку

Как показано в таблиці, кожний об'єктний екземпляр (екран, звіт) відносять до одного з трьох рівней складності. Ці місця підстановки вимірних і обчислених значень відмічені прямокутниками (прямокутник грає роль мітки-заповнювача). В свою чергу, сложность є функцією від параметрів клієнтських і серверних таблиць даних, які необхідні для генерації екрана і звіта, а також від кількості представлень і секцій, які входять в екран або звіт.

Оцінка кількості об'єктних вказівників

Тип об'єкта	Кількість	Вага			Всього
		Простий	Середній	Складний	
Екран	<input type="text"/>	<input type="text"/> *1	<input type="text"/> *2	<input type="text"/> *3	= <input type="text"/>
Звіт	<input type="text"/>	<input type="text"/> *2	<input type="text"/> *5	<input type="text"/> *8	= <input type="text"/>
3GL-компонент	<input type="text"/>			<input type="text"/> *10	= <input type="text"/>

# Модель композиції додатку

## Оцінка складності екрана

Екрани	Кількість серверних і клієнтських таблиць даних		
Кількість представлень	Всього <4 (<2 срв, <3 клт)	Всього <8 (2-3 срв, 3-5 клт)	Всього >8 (>3 срв, >8 клт)
<3	Простий	Простий	Середній
3-7	Простий	Середній	Складний
>8	Середній	Складний	Складний

## Оцінка складності звіта

Звіти	Кількість серверних і клієнтських таблиць даних		
Кількість представлень	Всього <4 (<2 срв, <3 клт)	Всього <8 (2-3 срв, 3-5 клт)	Всього >8 (>3 срв, >8 клт)
0 або 1	Простий	Простий	Середній
2 або 3	Простий	Середній	Складний
>4	Середній	Складний	Складний



# Модель композиції додатку

Після визначення складності кількості екранів, звітів і компонентів зважується у відповідності до табл. 1. Кількість об'єктних вказівників визначається перемноженням вихідного числа об'єктних екземплярів на вагові коефіцієнти і ніаступним підсумуванням проміжних результатів.

Для обліку реальних умов розробки обчислюється процент повторного використання програмних компонентів %REUSE і визначається кількість нових об'єктних вказівників NOP:

$$NOP = (\text{Об'єктні вказівники}) \cdot \left[ \frac{100 - \%REUSE}{100} \right]$$

Для оцінки трудовитрат, оснований на величині NOP, потрібно знати швидкість розробки продукта PROD. Цю швидкість визначають по наступній таблиці, яка враховує рівень досвіду розробників і зрілість середовища розробки.

# Модель композиції додатку

## Оцінка зрілості розробки

Досвід/можливості розробника	Зрілість/можливості середовища розробки	PROD
Дуже низька	Дуже низька	4
Низька	Низька	7
Номінальна	Номінальна	13
Висока	Висока	25
Дуже висока	Дуже висока	50

Проектні трудовитрати  $T$  оцінюються за формулою:

$$T = NOP / PROD$$

Де  $PROD$  – продуктивність розробки, виражена об'єктних вказівниках.

# Метод PERT

Інженерний метод оцінки трудоемності проекту PERT оснований на характеристиках 3 оцінок:

$M_i$  — найбільш вірогідна оцінка трудовитрат.

$O_i$  — мінімально можливі трудовитрати на реалізацію пакета робіт. Ні один ризик не реалізувався. Швидше точно не зробимо. Вірогідність того, що ми вкладемося в ці витрати, рівно 0.

$P_i$  — песиместична оцінка трудовитрат. Всі ризики реалізувались.

Оцінку середньої трудоемності по кожному елементарному пакету можна визначити за формулою:

$$E_i = (P_i + 4M_i + O_i)/6.$$

Для розрахунку середньоквадратичного відхилення використовується формула:

$$SKO_i = (P_i - O_i)/6.$$

Згідно центральної граничної теореми теорії ймовірностей сумарна трудоемність проекту може бути розрахована за формулою:

$$E = \sum E_i$$

# Метод PERT

А середнєквadraticне відхилення для оцінки сумарної трудоемності буде складати:

$$CKO = \sqrt{\left(\sum CKO_i\right)^2}$$

Тоді для оцінки сумарної трудоемності проекту, яку ми не перевищимо з ймовірністю 95%, можна застосувати формулу:

$$E_{95\%} = E + 2 * CKO.$$

Це означає, що вірогідність того, що проект перевищить дану оцінку трудоемності складає всього 5%.

# Метод PERT

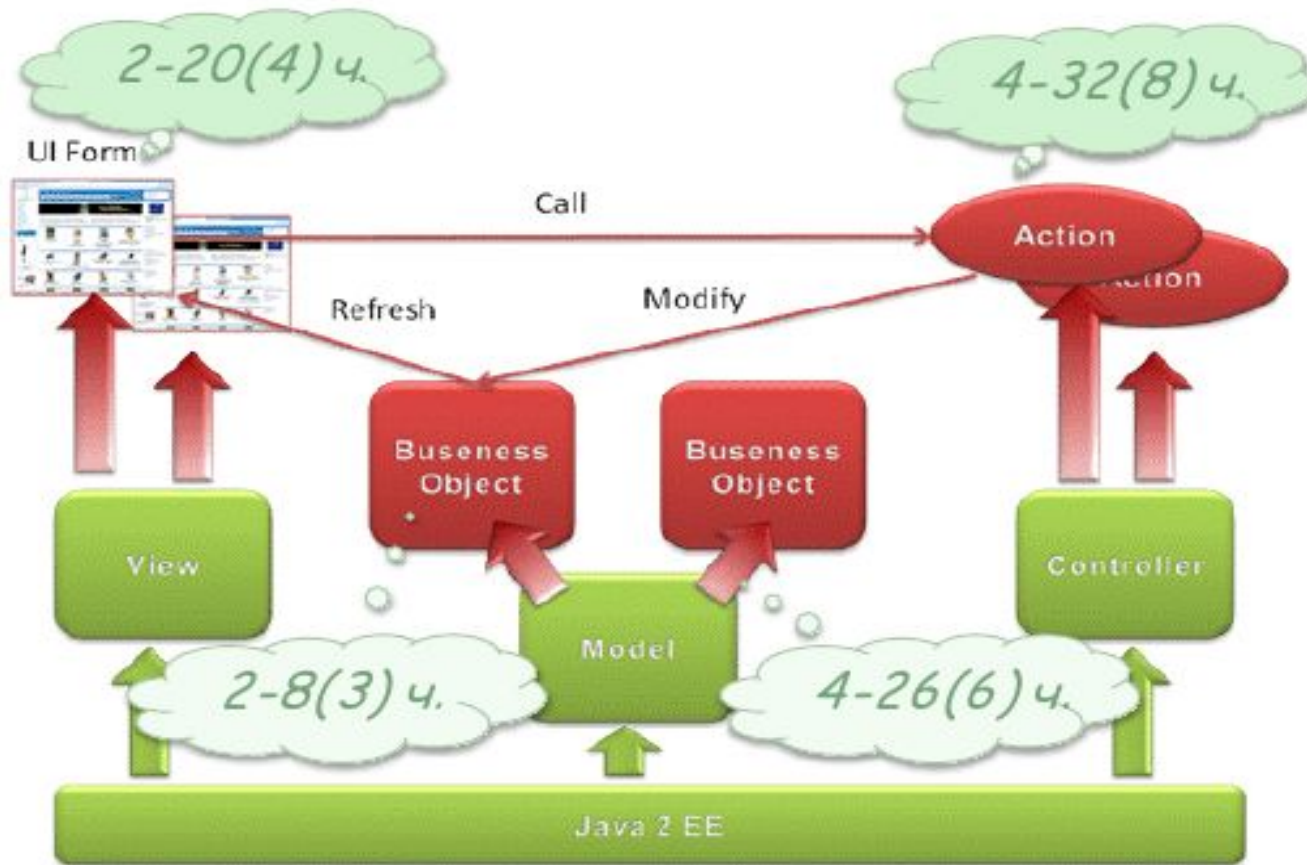


Рисунок 5.1. Високорівнева архітектура J2EE фреймворка для розробки додатку.

# Метод PERT

Високорівнева архітектура реалізовувала стандартний паттерн MVC, кожний з компонентів якого мав «точки розширення» для прикладної розробки, які на рисунку виділені червоним.

Такими точками розширення є:

Користувацький екран (UI Form), який збирався з готових візуальних компонентів.

Обробники(Action), які оброблювали на сервері додатків події від активних візуальних компонентів, які входять у склад екрану.

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

Новий додаток містить 20 користувацьких екранів, 60 обробників подій, 16 нових бізнес-об'єктів і 40 нових бізнес-методів.

# Метод PERT

Визначити:

1.  $E_i$  для кожного елементарного пакету
2.  $SKO_i$  - середньоквадратичне відхилення для кожного пакету
3. Середня трудоемність робіт по кодуванню
4. Середньоквадратичне відхилення для оцінки сумарної трудоемності.
5. Оцінку сумарної трудоемності проекту, яку ми не перевищимо з вірогідністю 95%