

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**МЕТОДИЧНІ ВКАЗІВКИ
до практичних занять з дисципліни
«Менеджмент проєктів програмного забезпечення»**

Одеса: Одеська політехніка, 2025

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»**

**МЕТОДИЧНІ ВКАЗІВКИ
до практичних занять з дисципліни
«Менеджмент проєктів програмного забезпечення»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю
121 – Інженерія програмного забезпечення**

Затверджено
на засіданні кафедри ІПЗ
Протокол № 1 від 30.08.2024 р.

Одеса: Одеська політехніка, 2025

Методичні рекомендації до практичних занять з дисципліни «Менеджмент проєктів програмного забезпечення» для здобувачів першого (бекалаврського) рівня вищої освіти за спеціальністю 121 – Інженерія програмного забезпечення (освітньо-професійна програма: Інженерія програмного забезпечення – 2022) / Укл.: *В. В. Любченко*. – Одеса: Одеська політехніка, 2025. – 37 с.

Укладач: **Любченко В. В.**, докт. техн. наук, проф.

Методичні вказівки містять теоретичні відомості, які необхідні і достатні для роботи на практичних заняттях з дисципліни «Менеджмент проєктів програмного забезпечення».

Призначаються для студентів денної форми навчання.

ЗМІСТ

Вступ.....	4
Практичне заняття 1. Аналіз зацікавлених сторін.....	5
Практичне заняття 2. Оцінювання розміру програмного проєкту. Визначення розміру програмного проєкту за допомогою методів FPA та COCOMO.....	8
Практичне заняття 3. Оцінювання розміру програмного проєкту. Визначення розміру програмного проєкту за допомогою методу UCP.....	13
Практичне заняття 4. Планування виконання проєкту.....	17
Практичне заняття 5. Аналіз ризиків проєкту.....	20
Практичне заняття 6. Моніторинг виконання проєкту.....	24
Практичне заняття 7. Управління якістю.....	26
Рекомендована література.....	28
Додаток. Знайомство з MS Project.....	29

Вступ

Сучасна індустрія інформаційних технологій вимагає від фахівців високої кваліфікації з менеджменту проєктів, особливо у контексті створення якісних та конкурентоспроможних продуктів. Вивчення менеджменту проєктів дозволяє оволодіти сучасними підходами до планування, організації та контролю за виконанням проєктів, що сприяє успішній реалізації іновацій в галузі ІТ.

Метою практичних занять з дисципліни «Менеджмент проєктів програмного забезпечення» є закріплення теоретичного матеріалу та набуття умінь виконання теорії проєктного управління до вирішення задач, що виникають під час розробки програмного забезпечення.

Практичне заняття 1

Аналіз зацікавлених сторін

Мета заняття: набуття навичок визначення і класифікації зацікавлених сторін проекту.

Зміст заняття: визначення зацікавлених сторін проекту, їх класифікація за інтересом і впливом, побудова карти зацікавлених сторін.

Теоретичні основи

Як частина визначення вимог проекту, ідентифікуються зацікавлені сторони. Зацікавлені сторони – це ті, хто має певну зацікавленість в проекті, тобто люди, на яких впливає проект або хто може вплинути на проект певним чином. Зацікавлені сторони походять з усіх частин організації, а іноді і з ширшої спільноти (постачальники, користувачі, спілки, мешканці тощо).

При ідентифікації зацікавлених сторін необхідно розглянути всі можливі зацікавлені сторони. Хоча проект може мати дуже велику групу зацікавлених сторін, неминуче буде існувати ієрархія, яка спростить ідентифікацію зацікавлених сторін. Наприклад, проект загальнодержавної системи може включати кожного менеджера у своїй групі зацікавлених сторін. Проте, така група була б некерованою, тому група повинна бути зменшена шляхом визначення лідерів груп інтересів, зокрема тих, хто впливає на систему, та інших критеріїв проекту. Проте презентації для більш широкої групи можуть знадобитися під час проекту, а ключові заходи зацікавлених сторін будуть спрямовані на всю ієрархію.

Пам'ятайте, що хоча зацікавлені сторони можуть бути як організаціями, так і людьми, в кінцевому рахунку, ви повинні спілкуватися з людьми. Переконайтеся, що ви ідентифікуєте правильних зацікавлених осіб в організації – зацікавленій стороні.

Різні зацікавлені сторони можуть мати спільність цілей на дуже загальному рівні. Наприклад, надання якісних послуг або підвищення ефективності організації. Але на більш деталізованих рівнях вони, можливо, побажають запровадити різні цілі та пріоритети. Рівень важливості для потреб та інтересів зацікавлених сторін також є ключем до успіху стратегії та розвитку проекту. Зацікавлена особа повинна мати право впливати і мати значення для проекту.

Сітка вплив / інтереси призначена для класифікації зацікавлених сторін відповідно до їх впливу на проект та їх зацікавленості у проекті (рис. 1.1).

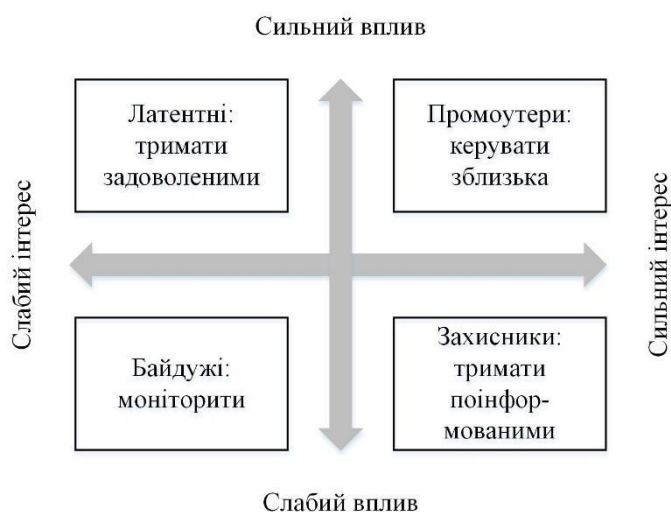


Рисунок 1.1 – Сітка вплив / інтереси

Позиція у сітці показує дії, які слід виконувати по відношенню до відповідних

зацікавлених сторін:

–Промоутери (ключові гравці) – це люди, якими потрібно постійно займатися і докладати максимум зусиль для їх задоволення. Рекомендовані канали: Issue, Change Logs, Status Meetings.

–Латентні (задовольняйте їх потреби) – з цими людьми слід працювати достатньо для того, щоб вони були задоволені, але не настільки, щоб їм набридлі ваші повідомлення. Рекомендовані канали: Steering Committee, Board Meeting Updates.

–Захисники (інформуйте їх): тримайте цих людей належним чином поінформованими і розмовляйте з ними, щоб упевнитися, що ніяких серйозних проблем не виникає. Ці люди часто можуть бути дуже корисні в деталях вашого проєкту. Найкращі канали: In-Person, Video, Email Updates.

–Байдужі (найменш важливі): спостерігайте за цими людьми, але не перевантажуйте їх надмірною комунікацією. Кращі канали: Send Email, Status Reports.

Для ефективного управління зацікавленими сторонами необхідна хороша стратегія управління зацікавленими сторонами. Вона має визначити та задокументувати підхід, який слід використовувати для збільшення підтримки та зменшення негативного впливу зацікавлених сторін протягом усього життя проєкту. Три основні сфери аналізу – внесок, відданість та підтримка, які використовуються на карті зацікавлених сторін та широко використовуються для керівництва управлінням зацікавлених сторін.

Існує ряд факторів, які визначають ступінь, в якому зацікавлена сторона має додати свій внесок у проєкт, щоб забезпечити його успіх. До таких факторів належать позиція та авторитет зацікавлених сторін в організації, ступінь, в якому проєкт покладається на конкретного зацікавленого учасника у наданні товару чи послуги, рівень «соціального» впливу особи та ступінь, до якої людині знайомі конкретні аспекти бізнесу. Кожен з цих факторів може бути проаналізований більш детально під час аналізу зацікавлених сторін. Результат підсумовується як індекс внеску, що має такі значення.

Таблиця 1.1 – Опис індексів внеску зацікавлених сторін

Значення	Опис
Критичний	Зацікавлена сторона має повноваження зробити або запобігти успіху проєкту
Бажаний	Проект може бути завершений навіть без активного внеску зацікавлених сторін, але це може серйозно вплинути на якість, тривалість та витрати на виконання Зацікавлена сторона може виступати в ролі прихильника проєкту для певних категорій осіб
Несуттєвий	Хоча зацікавлена сторона може додати свій внесок у проєкт, цей внесок або не є суттєвим, або його можна легше отримати від інших зацікавлених сторін Фізична особа з індексом внеску нижче "Несуттєвого" не є зацікавленою стороною

Кожна з зацікавлених сторін демонструє різний рівень підтримки проєкту, навіть на протязі усього життя проєкту. Діапазон підтримки до проєкту представлений такими значеннями.

Таблиця 1.2 – Опис рівнів підтримки зацікавлених сторін

Значення	Опис
Зобов'язання	Зацікавлена сторона зобов'язалася додати свій внесок у проєкт,

(зробити це)	бажано у письмовій формі, і доступна для цього Зобов'язання може бути задокументоване у формі узгодженого плану чи інших форм письмового повідомлення (електронна пошта, доповідна записка, лист, контракт, заява про наміри) із зазначенням того, що буде надано та коли
Підтримка (допомогти це зробити)	Зацікавлена сторона добре поінформована, бачить значення в тому, що робиться, розуміє свій внесок і готова забезпечити його, хоча офіційних зобов'язань не було укладено
Нейтральність (нехай це станеться)	Зацікавлена сторона може бути або може не бути проінформована про проєкт, вона не бере активної участі в будь-якій якості, або може бути байдужою щодо цілей та результатів проєкту
Не погодження (зупиніть, щоб цього не сталося)	Зацікавлена сторона може бути або може не бути проінформована, але не бачить значення в проєкті та виконаній роботі Він чи вона, скоріше за все, не будуть задіяні, а насправді вважають за краще, щоб робота взагалі не виконувалася

Перегляньте кожну із зацікавлених сторін та визначте, який їх поточний рівень вкладу та зобов'язань, та введіть їх назви(імена) на карті зацікавлених сторін. Наступним кроком є визначення місця, де кожна із зацікавлених сторін має бути для забезпечення належного рівня вкладу та прихильності до проєкту. Це відображатиме ідеальний або цільовий сценарій для проєкту. Введіть ім'я кожного учасника на карту в цільовій позиції. Якщо цільова позиція зацікавлених сторін справа або вище, намалюйте стрілку від поточної до цільової. Якщо на даний момент зацікавлена особа перебуває в ідеальній позиції, просто обведіть її ім'я. Приклад карти наведено на рис. 1.2.

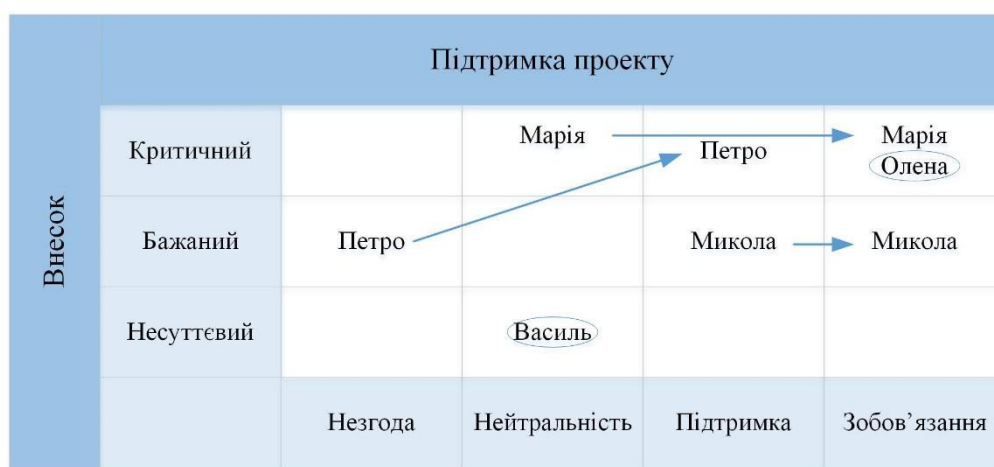


Рисунок 1.2 – Карта сценаріїв роботи з зацікавленими сторонами

Контрольні питання

1. Що таке зацікавлені сторони проєкту? Наведіть приклади.
2. Чому ідентифікація зацікавлених сторін є важливим етапом у менеджменті проєктів?
3. Які критерії використовуються для визначення ієрархії зацікавлених сторін?
4. Опишіть сітку вплив/інтересів та її ключові компоненти.
5. Хто такі промоутери в контексті аналізу зацікавлених сторін, і як з ними слід працювати?
6. Які дії рекомендовані для роботи із захисниками у проєкті?
7. Що означає індекс внеску зацікавлених сторін? Наведіть приклади значень та їх опис.

8. Які фактори впливають на рівень підтримки зацікавлених сторін у проєкті?
9. Навіщо потрібна стратегія управління зацікавленими сторонами, і які її основні елементи?

Практичне заняття 2

Оцінювання розміру програмного проєкту. Визначення розміру програмного проєкту за допомогою методів FPA та COCOMO

Мета заняття: набуття навичок застосування на практиці методів FPA та COCOMO.

Зміст заняття: визначення розміру, трудовитрат та часу, які потрібні для розробки програмного забезпечення.

Теоретичні основи

В методі показників функціональності (FPA) використовуються 5 інформаційних характеристик.

Зовнішній ввід – елементарний процес, який переміщає дані із зовнішнього середовища до застосування. Дані можуть містити як керуючу, так і ділову інформацію. Керуючі дані не можуть модифікувати внутрішній логічний файл.

Зовнішній вивід – елементарний процес, який переміщає дані, обчислені в застосуванні, до зовнішнього середовища. Крім того, в цьому процесі можуть оновлятися внутрішні логічні файли. Додатково цей процес може використовувати введені дані, що не підтримані внутрішніми логічними файлами, які утворюють критерії пошуку і параметри (дані, що носять тимчасовий характер і не зберігаються у внутрішньому логічному файлі).

Зовнішній запит – елементарний процес, що працює як з введеними, так і з виведеними даними. Його результат – дані, що повертаються із внутрішніх логічних файлів і зовнішніх інтерфейсних файлів.

Внутрішній логічний файл – група логічно пов'язаних даних, які розпізнаються користувачем, що розміщена всередині застосування і обслуговується через зовнішні вводи.

Зовнішній інтерфейсний файл – група логічно пов'язаних даних, які розпізнаються користувачем, що розміщена всередині іншого застосування і підтримується ним.

Кожний з виявлених характеристик ставиться у відповідність складності. Для цього характеристиці призначається низький, середній або високий ранг, а потім формується числова оцінка рангу. Для транзакцій ранжирування засноване на кількості посилань на файли і кількості типів елементів даних. Для файлів ранжирування засноване на кількості типів елементів-записів і типів елементів-даних, що входять у файл (табл. 2.1–2.2).

Таблиця 2.1 – Ранг і оцінка складності процесів

Посилання на файли	Елементи даних		
	1-4	5-15	>15
Зовнішні вводи			
0-1	Низький	Низький	Середній
2	Низький	Середній	Високий
>2	Середній	Високий	Високий
Зовнішні виводи			
0-1	Низький	Низький	Середній
2-3	Низький	Середній	Високий
>3	Середній	Високий	Високий
Зовнішні запити			
0-1	Низький	Низький	Середній
2-3	Низький	Середній	Високий
>3	Середній	Високий	Високий

Таблиця 2.2 – Ранг і оцінка складності внутрішніх логічних файлів

Типи елементів-записів	Елементи даних		
	1-19	20-50	>50
Внутрішні логічні файли			
1	Низький	Низький	Середній
2-5	Низький	Середній	Високий
>5	Середній	Високий	Високий
Зовнішні інтерфейсні файли			
1	Низький	Низький	Середній
2-5	Низький	Середній	Високий
>5	Середній	Високий	Високий

Процес застосування методу аналізу показників функціональності складається з наступних кроків:

Крок 1. Підрахунок кількості функцій в кожній категорії (вводи, виводи, запити, структури даних, інтерфейси).

Крок 2. Застосування вагових множників складності (табл. 2.3).

Таблиця 2.3 – Дані для розрахунку *FP*-метрик

Ім'я характеристики	Ранг, складність, кількість			
	Низький	Середній	Високий	Разом
Зовнішні вводи	× 3 =	× 4 =	× 6 =	=
Зовнішні виводи	× 4 =	× 5 =	× 7 =	=
Зовнішні запити	× 3 =	× 4 =	× 6 =	=
Внутрішні логічні файли	× 7 =	× 10 =	× 15 =	=
Зовнішні інтерфейсні файли	__ × 5 = __	__ × 7 = __	__ × 10 = __	= __
Загальна кількість				= __

Крок 3. Застосування факторів середовища (табл. 2.4).

Таблиця 2.4 – Визначення системних параметрів застосування

№	Системний параметр	Опис
1	Передача даних	Скільки засобів зв'язку потрібно для передачі або обміну інформацією з застосуванням або системою?
2	Розподілена обробка даних	Як обробляються розподілені дані і які функції обробки?
3	Продуктивність	Чи потребує користувач у фіксації часу відповіді або продуктивності?
4	Поширеність використаної конфігурації	Наскільки поширена апаратна платформа, на якій буде виконуватися застосування?
5	Швидкість транзакцій	Як часто виконуються транзакції? (щодня, щотижня, щомісяця)
6	Оперативне введення даних	Який відсоток інформації треба вводити в режимі онлайн?
7	Ефективність роботи кінцевого користувача	Застосування проєктувалося для забезпечення ефективної роботи кінцевого користувача?
8	Оперативне відновлення	Як багато внутрішніх файлів оновляється в онлайнівій транзакції?
9	Складність обробки	Чи виконує застосування інтенсивну логічну або математичну обробку?

10	Повторне використання	Застосування розроблялося для задоволення вимог одного або багатьох користувачів?
11	Легкість інсталяції	Наскільки трудним є перетворення і інсталяція застосування?
12	Легкість експлуатації	Наскільки ефективні і/або автоматизовані процедури запуску, резервування і відновлення?
13	Різноманітні умови розміщення	Чи була спроектована, розроблена і підтримана можливість інсталяції застосування в різних місцях для різних організацій?
14	Простота змін	Чи була спроектована, розроблена і підтримана в застосуванні простота змін?

Кожний коефіцієнт може приймати наступні значення: 0 – немає впливу, 1 – випадковий вплив, 2 – невеликий, 3 – середній, 4 – важливий, 5 – основний.

Крок 4. Обчислення коригувального множника складності:

$$CAF = 0.65 + 0.01 \sum_{i=1}^{14} F_i$$

Крок 5. Обчислення скоригованих показників функціональності

$$AFP = FP \times CAF.$$

Крок 6. Перетворення на LOC (додатково)

Перетворення показників функціональності на одиниці LOC виконується за допомогою табл. 2.5.

Таблиця 2.5 – Перерахунок FP-оцінок на LOC-оцінки

Мова програмування	Кількість LOC на одну FP	Мова програмування	Кількість LOC на одну FP
Асемблер	320	Visual C++	34
C	128	Delphi Pascal	29
C++	64	Eiffel	21
Java	53	Perl	21
Мови баз даних	40	HTML3	15
Visual Basic	32	SQL	13

Вихідна модель COSOMO має 3 варіанти, які відбивають розходження між процесами в різних областях програмного забезпечення. Ці варіанти були описані як звичайне, частково незалежне і вбудоване програмне забезпечення. Проекти по створенню звичайного програмного забезпечення характеризувалися тим, що функціональні можливості, якість, вартість і строки могли вільно мінятися, при цьому накладні витрати виявлялися мінімальними. Системи, що належали до розряду вбудованих, характеризувалися тим, що функціональні можливості, якість, вартість і строки жорстко контролювалися, а зміни вимагали узгодження з багатьма зацікавленими сторонами. Проміжний варіант частково незалежних проєктів перебуває посередині між двома цими варіантами.

Оціночні рівняння COSOMO мають таку форму:

$$\begin{aligned} \text{Трудомісткість} &= C_1 \times EAF \times (\text{Розмір})^{p_1} \\ \text{Час} &= C_2 \times (\text{Трудомісткість})^{p_2} \end{aligned}$$

де *Трудомісткість* – кількість людино-місяців;

C_1 – коефіцієнт масштабу;

EAF – уточнюючий фактор, який характеризує предметну область, персонал, середовище і інструментарій, використовуваний для створення робочих продуктів процесу;

Розмір – розмір кінцевого продукту (створеного людиною коду), який вимірюється у

вихідних інструкціях (DSI, delivered source instructions) – тисячах рядків коду (KLOC), потрібних для реалізації необхідної функціональної можливості;

p_1 – показник ступеня, що характеризує економію при великих масштабах, притаманну процесу створення кінцевого продукту, що використовується; зокрема, здатність процесу уникати непродуктивних видів діяльності;

$Час$ – загальна кількість місяців;

C_2 – коефіцієнт масштабу для строків виконання;

p_2 – показник ступеня, що характеризує інерцію і розпаралелювання, притаманні управлінню розробкою програмного забезпечення.

Таблиця 2.6 – Параметри моделі COCOMO

Тип проєкту	Значення параметрів			
	C_1	p_1	C_2	p_2
Звичайний	3,2	1,05	2,5	0,38
Частково незалежний	3,0	1,12	2,5	0,35
Вбудований	2,8	1,20	2,5	0,32

Множник EAF (effort adjustment factor), який є уточнюючим чинником робіт, визначається комбінацією багатьох параметрів. Значення параметрів наведені в табл. 2.7. Кожний параметр оцінюється як дуже низький, низький, номінальний, високий або дуже високий. Вплив значення кожного параметра визначається як множник, що звичайно змінюється в діапазоні від 0,5 до 1,5. Добуток цих 15-ти параметрів і використовується як коефіцієнт у рівнянні для трудовитрат.

Таблиця 2.7 – Значення факторів витрат у моделі COCOMO

Іденти- фікатор	Уточнюючий фактор робіт	Значення параметрів для факторів				
		Дуже низький	Низький	Номіналь- ний	Високий	Дуже високий
Атрибути програмного продукту						
RELY	Необхідна надійність	0,75	0,86	1,0	1,15	1,40
DATA	Розмір бази даних		0,94	1,0	1,08	1,16
CPLX	Складність продукту	0,7	0,85	1,0	1,15	1,30
Атрибути комп'ютера						
TIME	Обмеження часу виконання			1,0	1,11	1,50,
STOR	Обмеження об'єму основної пам'яті			1,0	1,06	1,21
VIRT	Мінливість віртуальної машини		0,87	1,0	1,15	1,30
TURN	Час реакції комп'ютера		0,87	1,0	1,07	1,15
Атрибути персоналу						
ACAP	Здатності аналітика	1,46	1,19	1,0	0,86	0,71
AEXP	Знання застосунків	1,29	1,15	1,0	0,91	0,82
PCAP	Здатності програміста	1,42	1,17	1,0	0,86	0,70
VEXP	Знання віртуальної машини	1,21	1,10	1,0	0,90	
LEXP	Знання мови програмування	1,14	1,07	1,0	0,95	

Атрибути проекту						
MODP	Використання сучасних методів	1,24	1,10	1,0	0,91	0,82
TOOL	Використання програмних інструментів	1,24	1,10	1,0	0,91	0,82
SCED	Необхідні строки розробки	1,23	1,08	1,0	1,04	1,10

Контрольні питання

1. Що таке метод показників функціональності (FPA) і для чого він використовується?
2. Які є категорії інформаційних характеристик у FPA, і що вони означають?
3. Що таке зовнішній ввід, і яка його роль у застосуванні?
4. Як визначається зовнішній вивід, і чим він відрізняється від зовнішнього вводу?
5. Що таке зовнішній запит, і як він взаємодіє з даними застосування?
6. Чим відрізняються внутрішній логічний файл та зовнішній інтерфейсний файл?
7. Які кроки включає процес застосування FPA, і в чому полягає їх сутність?
8. Як обчислюється скоригований показник функціональності (AFP)?
9. Що таке вихідна модель COSOMO, і які три варіанти вона передбачає?
10. Як обчислюються трудомісткість і строки виконання проекту за допомогою моделі COSOMO?

Практичне заняття 3

Оцінювання розміру програмного проєкту. Визначення розміру програмного проєкту за допомогою методу UCP

Мета заняття: набуття навичок застосування на практиці методу UCP.

Зміст заняття: визначення розміру, трудовитрат та часу, які потрібні для розробки програмного забезпечення.

Теоретичні основи

Оцінювання тривалості розробки UCP (Use Case Points) виконується на основі варіантів використання програмного забезпечення, що розробляється. Методика UCP дозволяє врахувати нефункціональні вимоги, організаційні ризики, компетенцію при оцінюванні та інші критерії.

Метод складається з 5 етапів.

Етап 1. Оцінка акторів дає оцінку складності інтерфейсів системи. В методі UCP розрізняють три типи акторів, відповідно до складності проєктування та реалізації інтерфейсу взаємодії системи з ними (табл. 3.1).

Таблиця 3.1 – Визначення типів акторів

Тип актора	Вага	Примітка
Простий	1	Інша система з визначеним API (REST, SOAP, dll)
Середній	2	Інша система, взаємодія з якою виконується за визначеним протоколом (наприклад, TCP/IP)
Складний	3	В більшості випадків взаємодія з користувачем за допомогою GUI або веб-сторінок

Для обчислення нескоректованої оцінки акторів (UAW) слід підрахувати кількості акторів кожного типу, помножити ці кількості на відповідні вагові коефіцієнти та знайти суму отриманих добутків.

Етап 2. Оцінка варіантів використання дає оцінку масштабів системи. Тип варіанту використання простіше за все визначити відповідно до кількості транзакцій (неподільних операцій) в ньому (табл. 3.2).

Таблиця 3.2 – Визначення типів варіантів використання за кількістю транзакцій

Тип варіанту використання	Вага	Кількість транзакцій
Простий	5	До 4
Середній	10	Від 4 до 7
Складний	15	Від 7

Також можна визначити тип варіанту використання за кількістю класів, реалізованих в його рамках (табл. 3.3). Кількість класів можна оцінити, наприклад, за допомогою методу аналізу іменників.

Таблиця 3.3 – Визначення типів варіантів використання за кількістю класів

Тип варіанту використання	Вага	Кількість класів
Простий	5	До 5
Середній	10	Від 5 до 10
Складний	15	Від 10

Ще один підхід до визначення типу варіанту використання базується на кількості об'єктів в базі даних (БД), що змінюються в його рамках (табл. 3.4).

Таблиця 3.4 – Визначення типів варіантів використання за кількістю змінюваних об'єктів в БД

Тип варіанту використання	Вага	Кількість об'єктів в БД
Простий	5	1
Середній	10	2
Складний	15	3 і більше

Для обчислення нескоректованої оцінки варіантів використання (UUCW) слід підрахувати кількості варіантів використання кожного типу, помножити ці кількості на відповідні вагові коефіцієнти та знайти суму отриманих добутків.

Показник UCP обчислюють за формулою:

$$UCP = UAW + UUCW$$

Якщо це не перша ітерація розробки, або якщо необхідно врахувати можливість розділення результатів роботи в команді оцінюється відсоток повторного використання програмних компонентів %REUSE і розрахунок виконується за формулою:

$$UCP = (UAW + UUCW) \left(\frac{100 - \%REUSE}{100} \right).$$

Етап 3. Оцінка технічних факторів дає коефіцієнт для оцінки складності архітектури застосування (табл. 3.5). Оцінка виконується за шкалою від 0 до 5, де 0 означає відсутність впливу, 3 – середній вплив, 5 – сильний вплив на розробку.

Таблиця 3.5 – Технічні фактори

Фактор	Опис	Вага	Пояснення	Оцінка
T1	Розподіленість системи	2	Інформує про потребу системи в розподілених обчисленнях	
T2	Час відгуку	1	Визначає ефективність системи з точки зору часу відгуку, потоку робіт тощо	
T3	Ефективність кінцевого користувача	1	Визначає ефективність користувача з точки зору його (її) сприйняття	
T4	Складність обробки	1	Визначає, чи будуть застосовуватись складні алгоритми для обробки даних	
T5	Фокус на повторному використанні коду	1	Визначає, чи будуть елементи коду системи використовуватись знову	
T6	Простота інсталяції	0,5	Визначає метод інсталяції та простоту інсталяції для кінцевого користувача, чи буде потреба в спеціалісті для інсталяції системи	
T7	Простота використання	0,5	Визначає узгодженість інтерфейсу користувача з його потребами	
T8	Портативність	2	Визначає, чи має застосування працювати в різних середовищах	
T9	Простота змін	1	Визначає, чи будуватиметься система в такий спосіб, щоб спростити її	

			модифікації в майбутньому	
T10	Паралельні обчислення	1	Інформує, чи будуть мати в системі місце паралельні обчислення	
T11	Засоби захисту	1	Визначає чи потребуватиме система спеціальні засоби захисту даних чи системи	
T12	Доступ до третьої сторони	1	Визначає ступень використання системи зовнішніми системами або акторами	
T13	Потреби в спеціальному навчанні	1	Визначає, чи потрібно організувати тренінги для користувачів	

Сума добутків вагових коефіцієнтів та оцінок для кожного з технічних факторів визначає показник TFactor. Оцінка технічного фактору обчислюється за формулою

$$TCF = 0,6 + (0,01 * TFactor)$$

Етап 4. Оцінка зовнішніх факторів дає коефіцієнт для організаційних ризиків при розробці (табл. 3.6). Оцінка виконується за шкалою від 0 до 5, де 0 означає відсутність впливу, 3 – середній вплив, 5 – сильний вплив на розробку.

Таблиця 3.6 – Зовнішні фактори

Фактор	Опис	Вага	Пояснення	Оцінка
F1	Знайомство з процесом розробки	1,5	Визначає, чи знайома команда з предметною областю та технічними аспектами вирішення проблеми клієнта. Особливу увагу слід приділити знанню методології, в якій виконується проект, а також знанню мов моделювання системи	
F2	Досвід подібних проєктів	0,5	Загальна уява про досвід команди в розробці програмного забезпечення	
F3	Досвід об'єктно-орієнтованої розробки	1	Досвід в проєктуванні об'єктно-орієнтованих застосувань, а також в підтримці засобів для розробки інформаційних систем	
F4	Досвідченість провідного аналітика	0,5	Здатності аналітика отримати вимоги від клієнта та знання щодо задач, які буде вирішувати система	
F5	Мотивація	1	Здатність команди займатись призначеною задачею	
F6	Стабільність вимог	2	Визначає, чи не будуть вимоги часто змінюватись	
F7	Часткова зайнятість працівників	-1	Визначає, наскільки великою є частка працівників часткової зайнятості	
F8	Складність мови програмування	-1	Визначає, наскільки складно вивчити мову програмування: 0 – легко, за тиждень можна опанувати; 1 – щонайменше 2 тижні потрібно для опанування мови; 2 – щонайменше місяць потрібно для опанування мови;	

			3 – потрібен спеціальний тренінг з мови; 4 – потрібен спеціальний тренінг та допомога під час виконання проєкту; 5 – складно, потрібні лише досвідчені люди.	
--	--	--	--	--

Сума добутків вагових коефіцієнтів та оцінок для кожного з зовнішніх факторів визначає показник EFactor. Оцінка зовнішнього фактору обчислюється за формулою

$$EF = 1,4 + (-0,03 * EFactor)$$

Етап 5. Результуючі оцінки

Скоректовані UCP обчислюються за формулою

$$AUCP = UCP * TCF * EF.$$

Для визначення тривалості розробки потрібно знати, якій кількості робочих годин відповідає один UCP. Для цього підраховується кількість факторів з множини F1 – F8, оцінки яких за абсолютним значенням перевищують 3. Якщо результат 2 або менше, то для розрахунку тривалості розробки приймається, що одному UCP відповідає 20 робочих годин. Якщо результат 3 або 4, то одному UCP відповідає 28 робочих годин. Якщо результат перевищує 4, то рекомендовано переглянути умови виконання проєкту. У разі неможливості це зробити приймається, що одному UCP відповідає 36 робочих годин.

Контрольні питання

1. Що таке методика UCP (Use Case Points) і які її основні цілі?
2. Як визначаються три типи акторів у методиці UCP і що означає їх складність?
3. Що таке нескоректована оцінка акторів (UAW) і як вона обчислюється?
4. Як визначають типи варіантів використання і на основі яких критеріїв це виконується?
5. Що таке нескоректована оцінка варіантів використання (UUCW) і як вона обчислюється?
6. Як враховується можливість повторного використання програмних компонентів у формулі UCP?
7. Яким чином технічні фактори враховуються в методиці UCP і як обчислюється коефіцієнт TCF?
8. Що таке зовнішні фактори в методиці UCP і як вони впливають на оцінювання розробки?
9. Як обчислюються скоректовані UCP (AUCP) і які етапи для цього необхідні?
10. Як визначається тривалість розробки на основі скоректованих UCP, і які фактори впливають на кількість робочих годин для одного UCP?

Практичне заняття 4

Планування виконання проєкту

Мета заняття: набуття навичок застосування на практиці мережного планування для складання розкладів.

Зміст заняття: розробка плану проєкту з розробки програмного забезпечення.

Теоретичні основи

Мережний граф – це орієнтований граф, в якому вершинами позначені роботи проєкту, а дугами – часові взаємозв'язки робіт.

Мережний граф має відповідати таким властивостям:

1. Кожній роботі відповідає одна і тільки одна вершина. Жодна робота не може бути представлена на мережному графі двічі. Однак будь-яку роботу можна розбити на кілька окремих робіт, кожній з яких буде відповідати окрема вершина графа.

2. Жодна робота не може бути розпочата до того, як завершаться всі роботи, що безпосередньо їй передують. Тобто якщо до деякої вершини входять дуги, то робота може початися тільки після закінчення всіх робіт, з яких виходять ці дуги.

3. Жодна робота, яка безпосередньо слідує за деякою роботою, не може початися до моменту її закінчення. Іншими словами, якщо з роботи виходить кілька дуг, то жодна з робіт, в які входять ці дуги, не може початися до закінчення цієї роботи.

4. Початок і кінець проєкту позначені роботами з нульовою тривалістю. Такі роботи називаються віхами і позначають початок або кінець найбільш важливих етапів проєкту.

Приклад. Як приклад розглянемо проєкт «Розробка програмного комплексу». Припустимо, що проєкт складається з робіт, характеристики яких наведені в табл. 4.1.

Таблиця 4.1 – Характеристики робіт проєкту

Номер роботи	Назва роботи	Тривалість	Попередники
1	Постановка задачі	10	-
2	Розробка інтерфейсу	5	1
3	Розробка модулів обробки даних	7	2, 4
4	Розробка структури бази даних	6	1
5	Заповнення бази даних	8	4
6	Налагодження програмного комплексу	5	3, 5
7	Тестування і виправлення помилок	10	6
8	Складання програмної документації	5	6

Мережний графік для цього проєкту зображений на рис. 4.1.

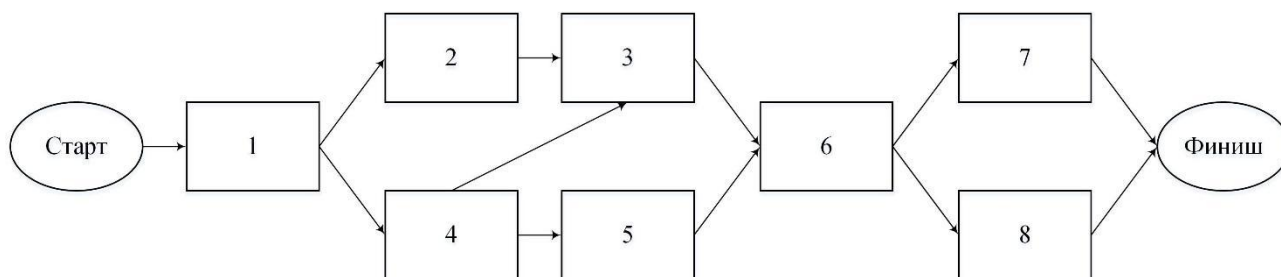


Рисунок 4.1 – Мережний граф проєкту

Мережний графік дозволяє за заданими значеннями тривалості робіт знайти критичні роботи проекту та його критичний шлях.

Критичною називається така робота, для якої затримка її початку призведе до затримки терміну закінчення проекту в цілому. Такі роботи не мають запасу часу. Некритичні роботи мають певний запас часу, і в межах цього запасу їх початок може бути затримано.

Критичний шлях – це шлях від початкової до кінцевої вершини мережевого графіка, що проходить тільки через критичні роботи. Сумарна тривалість робіт критичного шляху визначає мінімальний час реалізації проекту.

Знаходження критичного шляху зводиться до знаходження критичних робіт і виконується в два етапи.

1. Обчислення раннього часу початку кожної роботи проекту. Ця величина показує час, раніше якого робота не може бути розпочато.

2. Обчислення пізнього часу початку кожної роботи проекту. Ця величина показує час, пізніше якого робота не може бути розпочато без збільшення тривалості всього проекту.

Критичні роботи мають однакове значення раннього і пізнього часу початку.

Позначимо Dur_i – час виконання роботи i , ES_i – ранній час початку роботи i , EF_i – ранній час завершення роботи i , LS_i – пізній час початку роботи i , LF_i – пізній час завершення роботи i . Тоді

$$EF_i = ES_i + Dur_i,$$

$$ES_i = EF_j,$$

де G – множина робіт, які безпосередньо передують роботі i . Ранній час завершення початкової роботи проекту приймається рівним нулю.

Оскільки остання робота проекту має нульову тривалість, ранній час її початку збігається з тривалістю всього проекту. Тепер можемо обчислити пізні часи початку і завершення робіт:

$$LS_i = LF_i - Dur_i,$$

$$LF_i = LS_j,$$

де H – множина робіт, безпосередньо наступних за роботою i .

Мережний граф з зазначеннями часових характеристик показано на рис. 4.2.

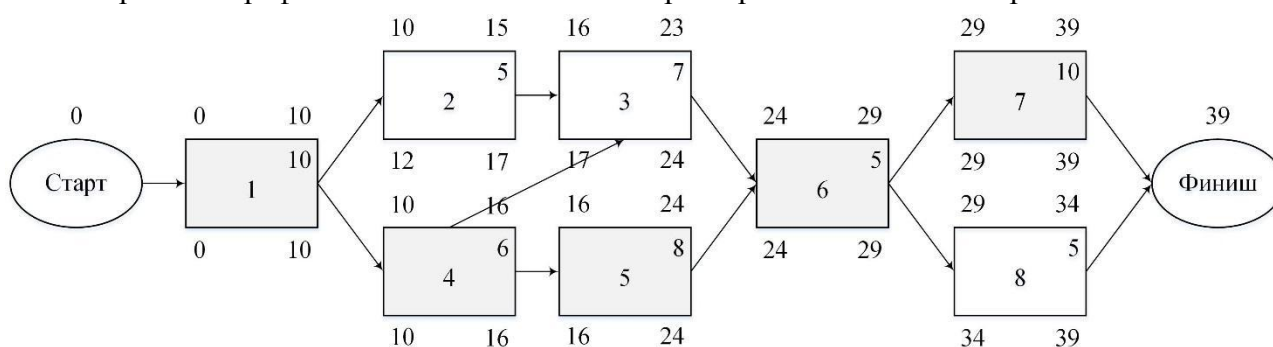


Рисунок 4.2 – Мережний граф з часовими характеристиками

Після визначення часових характеристик можна розрахувати резерви часу (total float) для кожної роботи як

$$TF_i = LS_i - EF_i.$$

Ця величина показує, наскільки можна затримати початок роботи без збільшення тривалості всього проекту. Для критичних робіт резерв часу дорівнює нулю. Тому зусилля менеджера проекту повинні бути спрямовані в першу чергу на забезпечення своєчасного виконання цих робіт.

Для некритичних робіт резерв часу більше нуля, що дає менеджеру можливість маневрувати часом їх початку і використовуваними ними ресурсами. Можливі такі варіанти.

1. Затримка початку роботи на величину, що не перевищує резерв часу, а необхідні для роботи ресурси спрямовуються для виконання робіт критичного шляху. Це може дати зменшення тривалості критичної роботи і проекту в цілому.

2. Недовантаження некритической роботи ресурсами. Як результат тривалість її збільшується в межах резерву часу, а звільнений ресурс задіюється для виконання критичної роботи, що також призведе до зменшення тривалості її і всього проекту.

Критичний шлях виходить з'єднанням критичних робіт на мережному графі (на рис. 4.2 він показаний сірими вершинами).

Контрольні питання

1. Що таке мережний граф, і які властивості він має відповідати?
2. Як позначаються початок і кінець проекту на мережному графі? Що таке віхи?
3. Що таке критична робота, і чому вона важлива для управління проектом?
4. Як визначається критичний шлях проекту, і що він показує?
5. Які етапи включає процес знаходження критичного шляху?
6. Що таке ранній і пізній час початку роботи, і як вони обчислюються?
7. Як обчислюється резерв часу для кожної роботи, і чому він дорівнює для критичних робіт?
8. Як затримка початку некритичної роботи впливає на загальну тривалість проекту?
9. Які варіанти використання резервів часу для некритичних робіт може застосувати менеджер проекту?
10. Як обчислюються часові характеристики робіт за формулами раннього завершення (EF_i) та пізнього завершення (LF_i)?

Практичне заняття 5

Аналіз ризиків проєкту

Мета заняття: Набуття навичок застосування методів якісного аналізу ризиків, розробка заданих стратегій реагування на ризик та PERT-моделювання.

Зміст заняття: визначення ризиків проєкту з розробки програмного забезпечення та розробка стратегій їх подолання.

Теоретичні основи

Ідентифікація ризиків – це виявлення ризиків, здатних вплинути на проєкт, і документальне оформлення їх характеристик.

Діаграма Ішікави (cause-effect diagram) – графічний інструмент, який дозволяє наочно і систематизовано аналізувати взаємозв'язки наслідків (effects) і причин (causes), які породжують ці наслідки або впливають на них. Ще ці діаграми називають «діаграмами рибного кістяка» (fishbone diagram) за їх зовнішню схожість зі скелетом риб. Але яке б ім'я не використовувалося, необхідно пам'ятати, що цінність цього методу полягає в сприянні категоризації і структуризації множини потенційних причин, а також ідентифікації найбільш імовірної кореневої причини для наслідку, який вивчається.

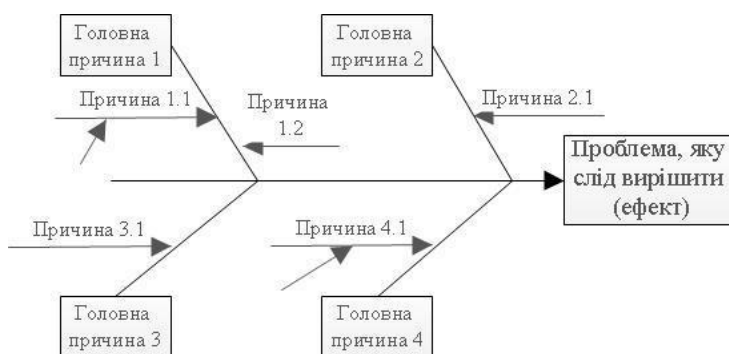


Рисунок 5.1 – Приклад діаграми Ішікави

Для визначення рангу ризику використовується матриця ймовірностей та наслідків (рис. 5.2). Ранг ризику визначається множенням ваги ймовірності і значенням наслідків.

		Імовірність			Ранг ризику	
		1	2	3		
Вплив	3	3	6	9	9	- високий
	2	2	4	6	6	- середній
	1	1	2	3	3	- низький

Рисунок 5.2 – Ранг ризику і матриця ймовірностей та наслідків

Для оцінки ризиків необхідна точна і адекватна інформація. Використання неточної інформації веде до помилок в оцінці. Невірна оцінка ризику також є ризиком.

Результатом якісного аналізу ризиків є їх докладний опис (рис. 5.3).

Номер: R-101	Категорія: Технологічний
--------------	--------------------------

Причина: Нестача кваліфікованих кадрів	Умови: Розробники будуть використовувати нову платформу – J2EE
Наслідки: Низька продуктивність розробки	Вплив: Збільшення термінів і трудомісткості розробки
Імовірність: Дуже ймовірний	Ступінь впливу: Критична
Близькість: Дуже скоро	Ранг: 6
Вихідні дані: «Зміст проєкту», «План забезпечення ресурсами»	

Рисунок 5.3 – Картка з описом ризику

Заплановані операції з реагування на ризики повинні відповідати серйозності ризику, бути економічно ефективними у вирішенні проблеми, своєчасними, реалістичними в контексті проєкту і узгодженими з усіма учасниками.

Можливі чотири методи реагування на ризики:

- ухилення від ризику (risk avoidance);
- передача ризику (risk transference);
- зниження ризику (risk mitigation);
- прийняття ризику (risk acceptance).

Ухилення від ризику передбачає зміну плану управління проєктом таким чином, щоб виключити загрозу, викликану негативним ризиком, захистити цілі проєкту від наслідків ризику або послабити цілі, що знаходяться під загрозою. Деякі ризики, що виникають на ранніх стадіях проєкту, можна уникнути за допомогою уточнення вимог, отримання додаткової інформації або проведення експертизи.

Передача ризику – це перекалення негативних наслідків загрози з відповідальністю за реагування на ризик на третю сторону. Передача ризику пов'язана просто з переносом відповідальності за його управління іншій стороні, але ризик при цьому нікуди не дівається. Передача ризику практично завжди передбачає виплату премії за ризик стороні, що приймає на себе ризик.

Зниження ризиків – це зниження ймовірності та/або наслідків негативної ризикованої події до прийнятних меж. Прийняття запобіжних заходів щодо зниження ймовірності настання ризику або його наслідків часто виявляються більш ефективними, ніж зусилля по усуненню негативних наслідків, що вживаються після настання події ризику.

І, нарешті, прийняття ризику означає, що команда проєкту усвідомлено прийняла рішення не змінювати план управління проєктом у зв'язку з ризиком або не знайшла відповідної стратегії реагування. Команда буде змушена приймати всі «невідомі ризики».

Для більш реалістичного планування з урахуванням впливу ризиків застосовують метод PERT, який майже повністю повторює метод критичного шляху, за винятком того, що PERT вважає, що тривалість кожної операції має межі, які виходять із статистичного розподілу.

Алгоритм методу PERT

1. Для кожної роботи проєкту визначаємо мінімальну, найбільш правдоподібну та максимальну тривалість.

2. Обчислюємо очікуване значення (T_e) та дисперсію (σ^2) тривалості кожної роботи.

Очікуване значення обчислюється як середнє зважене значення за формулою

$$T_e = \frac{T_o + 4T_m + T_p}{6}.$$

Оцінка середньоквадратичного відхилення обчислюється як

$$\sigma = \frac{T_p - T_o}{6}.$$

3. Визначити критичний шлях проекту, як найдовший шлях.

4. Обчислити середнє значення тривалості проекту (T_E) шляхом підсумовування очікуваної тривалості робіт критичного шляху.

5. Обчислити середнє квадратичне відхилення для очікуваної тривалості проекту (σ_E) як

$$\sigma_E = \sqrt{\sum \sigma_i^2}.$$

6. Припустимо, що загальна тривалість проекту відповідає нормальному розподілу з T_E та σ_E^2 .

7. Обчислити імовірність виконання проекту за час T , використовуючи нормальний розподіл. Для цього розраховується величина Z , що приводиться в статистичних таблицях (Z – кількість стандартних відхилень від середньої величини):

$$Z = \frac{T - T_E}{\sigma_E}.$$

Або можна застосувати функцію Excel *normdist*(T , T_E , σ_E , 1).

Розглянемо приклад. Припустимо, що проект складається з робіт, характеристики яких наведені в табл. 5.1.

Таблиця 5.1 – Характеристики робіт проекту

Номер роботи	Назва роботи	Тривалість			Поперед-ники
		мінімаль-на	правдопо-дібна	максималь-на	
1	Постановка задачі	4	10	13	-
2	Розробка інтерфейсу	4	5	7	1
3	Розробка модулів обробки даних	5	7	11	2, 4
4	Розробка структури бази даних	3	6	9	1
5	Заповнення бази даних	8	8	11	4
6	Налагодження програмного комплексу	3	5	8	3, 5
7	Тестування і виправлення помилок	9	10	14	6
8	Складання програмної документації	3	5	7	6

Обчислимо очікувані значення та дисперсії тривалості робіт проекту (табл. 4.2).

Таблиця 4.2 – Результати обчислень

Номер роботи	Назва роботи	Очікувана тривалість	Дисперсія	Попередники
1	Постановка задачі	9,5	1,5	—
2	Розробка інтерфейсу	5,2	0,5	1
3	Розробка модулів обробки даних	7,3	1,0	2, 4
4	Розробка структури бази даних	6,0	1,0	1

5	Заповнення бази даних	8,5	0,5	4
6	Налагодження програмного комплексу	5,2	0,8	3, 5
7	Тестування і виправлення помилок	10,5	0,8	6
8	Складання програмної документації	5,0	0,7	6

Побудуємо мережний граф проєкту (рис. 5.4).

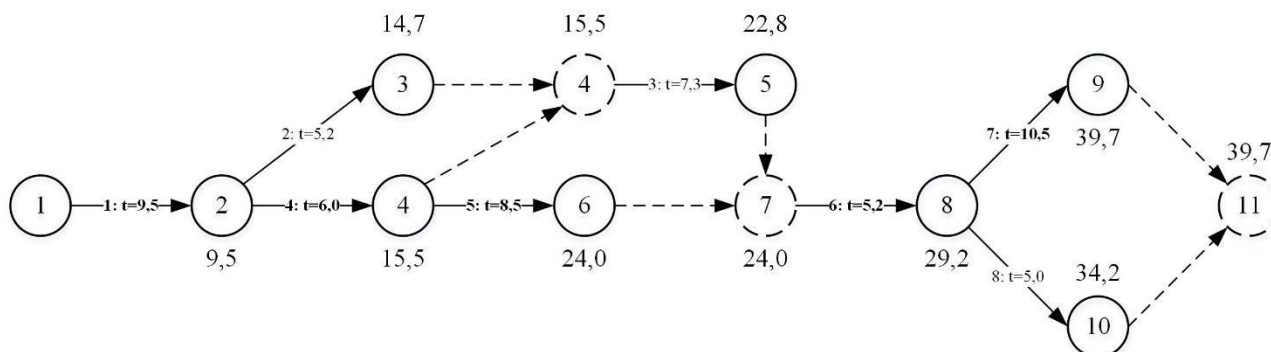


Рисунок 5.4 – Мережний граф з часовими характеристиками

Як бачимо, на критичному шляху знаходяться роботи 1, 4, 5, 6 та 7. Очікувана тривалість виконання проєкту складає 39,7, середньквадратичне відхилення – 2,2.

Ми аналізували цей приклад в попередньому занятті за допомогою методу передування і отримали тривалість проєкту 39. Визначимо імовірність завершити проєкт до 39 тижня за допомогою засобів Excel: $=\text{НОРМ.РАСП}(39; 39,7; 2,2; 1) = 0,37$.

Контрольні питання

1. Що таке ідентифікація ризиків, і яке її основне завдання?
2. Що таке діаграма Ішикави, і в яких випадках вона застосовується?
3. Як визначається ранг ризику за допомогою матриці ймовірностей та наслідків?
4. Які наслідки може мати використання неточної інформації при оцінці ризиків?
5. Які існують методи реагування на ризики, і в чому полягає кожен із них?
6. Що означає ухилення від ризику, і які підходи можуть використовуватись для цього?
7. Як здійснюється передача ризику, і які є її особливості?
8. У чому переваги зниження ризику порівняно з усуненням наслідків ризикової події?
9. Як реалізується метод PERT, і які основні етапи цього методу?
10. Як за допомогою методу PERT оцінюється імовірність виконання проєкту у заданий термін?

Практичне заняття 6

Моніторинг виконання проєкту

Мета заняття: набуття навичок застосування методу освоєного об'єму для спостереження за станом проєкту.

Зміст заняття: використання моніторингових засобів для визначення стану проєкту.

Теоретичні основи

Метод освоєного об'єму – інтегрований аналіз виконання календарного плану проєкту та бюджету за вартісними оцінками, найбільш поширений метод вимірювання виконання проєкту і його управління. Освоєний об'єм завдання – це затверджений бюджет, виділений на її рішення. Цей метод дозволяє в одному звіті подати відомості про виконання видатків та розкладу, причому і розклад і витрати вимірюються в валюті, в якій ведеться бюджет проєкту. Вимірювання і витрат, і розкладу проєкту в грошових одиницях є найбільш інформативним описом стану проєкту.

Набір вимірювачів, які застосовують в методі освоєного об'єму, наведені в табл. 6.1.

Таблиця 6.1 – Фреймворк вимірювачів

Метрика	Рівняння	Коментар
Освоєний об'єм (EV)	$EV = \% \text{ work completed} \times \text{baseline cost}$	Бюджетна вартість виконаних робіт
Планована вартість (PV)	$PV = \% \text{ work scheduled} \times \text{baseline cost}$	Запланована бюджетна вартість робіт
Фактична вартість (AC)		Фактична вартість виконаних робіт
Відхилення за строками (SV)	$SV = EV - PV$	
Відхилення за вартістю (CV)	$CV = EV - AC$	
Індекс виконання календарного плану (SPI)	$SPI = EV / PV$	$SPI < 1$: Проєкт не відповідає графіку $SPI > 1$: Проєкт випереджає графік
Індекс виконання бюджету (CPI)	$CPI = EV / AC$	$CPI < 1$: Проєкт перевищує бюджет $CPI > 1$: Проєкт в межах бюджету
Індекс бюджет-графік (CSI)	$CSI = CPI \times SPI$	$CSI < 1$: Проєкт не є здоровим $CSI > 1$: Проєкт здоровий

Індекс завершення виконання проєкту (*To Complete Performance Index*) показує рівень зусиль потрібних для того, щоб завершити проєкт вчасно:

$$TCPI = \frac{BAC - EV}{EAC - AC},$$

де *EAC* – поточна оцінка коштів для завершення проєкту.

Якщо $TCPI < 1,0$ (незвичайний випадок), то решта робіт проєкту може бути виконана з нижчим рівнем продуктивності, ніж вже завершені роботи проєкту. Слід пам'ятати, що хоча це здається очевидним, висновок також залежить від характеру робіт, що залишилися. Якщо $TCPI > 1,0$ (більш звичайний випадок), то інші роботи проєкту мають бути виконані на більш високому рівні продуктивності, ніж завершені роботи проєкту.

Контрольні питання

1. Що таке метод освоєного об'єму, і яку інформацію він дозволяє отримати про стан проєкту?

2. Як визначається метрика освоєного об'єму (EV), і що вона описує?
3. Чим відрізняється планована вартість (PV) від фактичної вартості (AC)?
4. Що означає від'ємне значення відхилення за строками (SV), і як його обчислити?
5. Як інтерпретується індекс виконання календарного плану (SPI), якщо його значення менше за 1?
6. Що означає індекс виконання бюджету (CPI), якщо його значення більше за 1?
7. Як обчислюється і що показує індекс бюджет-графік (CSI)?
8. Яке значення індексу завершення виконання проєкту (TCPI) вважається незвичайним, і що це означає для продуктивності проєкту?
9. Як використовується поточна оцінка коштів для завершення проєкту (EAC) у формулі TCPI?
10. Чому вимірювання витрат і строків у валюті бюджету проєкту є важливим аспектом методу освоєного об'єму?

Практичне заняття 7

Управління якістю

Мета завдання: набуття навичок застосування моделі розповсюдження дефектів для вибору заходів управління якістю.

Зміст завдання: визначення заходів управління якістю, які доречно використовувати в проєкті.

Теоретичні основи

Часто перед керівником проєкту постає дилема: забезпечити високу якість всього процесу розробки інформаційної системи, чи заощадити кошти на вдосконаленні ранніх стадій процесу. В IBM було розроблено модель розповсюдження дефектів (*Defect Flow Model*), яка показує, до яких фінансових результатів можуть привести заощадження на коштах контролю якості для різних фаз проєкту. Зразок опису етапу процесу розповсюдження та виявлення помилок, що використовується в моделі, показана на рис. 7.1.



Рисунок 7.1 – Шаблон опису етапу в моделі розповсюдження дефектів

З використанням обчислених кількостей помилок, які знайдені на різних етапах, та застосовуючи історичні данні про вартість помилок, можна обчислити, скільки коштуватиме виправлення помилок за різних форм роботи групи контролю якості. Далі можна застосувати *cost-benefit analysis*, щоб визначити доцільність застосування різних заходів забезпечення якості.

Розглянемо приклад. В роботі команди, яка намагається застосовувати фреймворк Scrum, можна виділити три етапи – формування беклогу продукту, розробка чергової версії продукту та приймальне тестування. За статистикою до беклогу чергової версії продукту в середньому попадають 20 помилкових історій користувачів, половина з яких підсилює свій вплив вдвічі, якщо реалізована в коді. В процесі розробки також припускаються помилки, за статистикою очікувана кількість помилок однієї версії – 30. За рахунок відлагодження та тестування вдається нейтралізувати до 60% помилок етапу. Під час приймального тестування знаходять до 80% помилок, решта уходить з продуктом користувачам. Модель розповсюдження та виявлення помилок для описаного процесу показана на рис. 7.2.

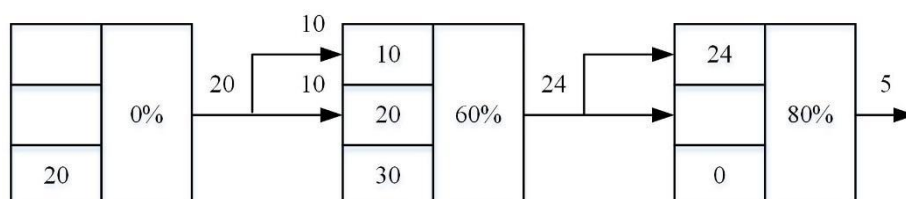


Рисунок 7.2 – Модель розповсюдження дефектів для аналізованого процесу

Як бачимо, з релізом до експлуатації потрапляє в середньому 5 помилок. Одним з засобів покращення ситуації є введення continuous integration на другому етапі, що дозволяє нейтралізувати до 90% помилок етапу. На рис. 7.3 показана модель для такого процесу.

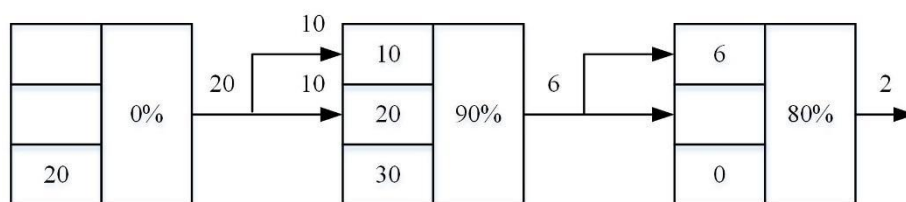


Рисунок 7.3 – Модель розповсюдження дефектів для модифікованого процесу

В цьому випадку з релізом до експлуатації потрапляє в середньому 2 помилки. Та чи варто було витратити кошти на забезпечення continuous integration? Для цього необхідно розрахувати витрати та виграші. Хай відомо, що виправлення помилок на етапі складання беклогу коштує 1 грошову одиницю (го), на етапі розробки – 6 го, під час приймального тестування – 15 го, після релізу – 80 го. А вартість організації continuous integration становить 300 го.

Розрахуємо вартість виправлення помилок:

$$Cost_1 = 36 \times 6 + 19 \times 15 + 5 \times 80 = 901 \text{ го}$$

$$Cost_2 = 54 \times 6 + 4 \times 15 + 2 \times 80 = 544 \text{ го}$$

Отже економія складе 357 го, тобто перевищить кошти впровадження continuous integration (навіть без врахування повторного використання інсталюваних засобів для нього). Таким чином можна стверджувати про доречність запровадження continuous integration в процес.

Контрольні питання

1. У чому полягає основна дилема керівника проєкту щодо забезпечення якості процесу розробки інформаційної системи?
2. Що таке модель розповсюдження дефектів (Defect Flow Model), і яка її основна мета?
3. Як у моделі розповсюдження дефектів визначаються етапи виявлення та усунення помилок?
4. Чому економія коштів на вдосконаленні ранніх стадій процесу може призводити до фінансових втрат на пізніх етапах?
5. Як історичні дані про вартість помилок використовуються для оцінки ефективності роботи групи контролю якості?
6. Що показує шаблон опису етапу процесу розповсюдження дефектів?
7. Які ключові метрики потрібні для обчислення вартості виправлення помилок на різних етапах?
8. Як застосовується аналіз витрат і вигод (cost-benefit analysis) для прийняття рішень щодо заходів забезпечення якості?
9. У яких випадках виправлення помилок може бути більш доцільним на пізніх етапах проєкту, а не на ранніх?
10. Як підхід до контролю якості впливає на загальну вартість та успішність проєкту?

Рекомендована література

1. Катренко А. В. Управління ІТ-проектами. Львів: Новий світ-2000, 2021. 550 с.
2. Brewer J. L., Dittman K. C. Methods of IT Project Management. Purdue University Press, 2022. 582 p.
3. Peters L. J. Software Project Management. Auerbach Publications, 2024. 244 p.

Додаток. Знайомство з MS Project

MS Project є застосуванням родини Microsoft Office, тому загальні принципи роботи з ним повністю повторюють принципи роботи з іншими застосуваннями Microsoft Office. Призначення MS Project – інформаційна підтримка керування проєктами.

При запуску MS Project автоматично створюється новий проєкт. Якщо ж такого з певної причини не відбулося, або ви працювали з іншим проєктом, то для створення нового проєкту потрібно виконати **File** → **New**. Для зміни параметрів проєкту слід виконати **Project** → **Project Information**, відкриється діалогове вікно (рис. А.1), в якому можна задати відповідні параметри.

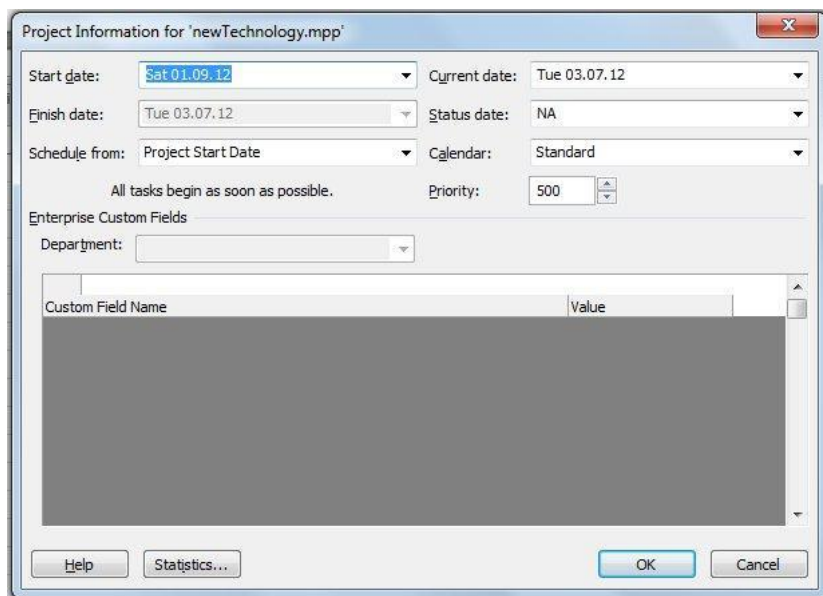


Рисунок А.1 – Діалогове вікно інформації про проєкт

Як усталено для проєкту встановлюється відлік від дати початку, тому дата закінчення проєкту неактивна. Часто проєкт планується таким чином, що він повинен бути обов'язково завершений до певної дати. В цьому випадку відома дата завершення, тоді в списку, що розкривається, **Schedule from** потрібно вибрати **Project Finish Date**. Потім під час складання плану визначається, коли проєкт повинен бути початий, щоб всі роботи були закінчені в строк.

Як усталено для проєкту встановлюється стандартний календар, у якому вже встановлені час і початок роботи, а також час перерви. До календаря не включені вихідні та укорочені дні. Часто такий варіант роботи не влаштовує при плануванні проєкту, тому можна змінити наявні настроювання або створити новий календар.

Для роботи з календарем виконайте **Project** → **Change Working Time** (рис. А.2). У верхньому списку **For Calendar** відразу встановлений стандартний календар проєкту. У центрі на календарі кольорами відповідно до легенди відразу відзначений робочий і неробочий час.

У вкладці **Work Weeks** для обраних днів можна змінити параметри. Варіант **Use Project default time for these days** застосовує стандартні настроювання, якщо раніше параметри дня були змінені. Варіант **Set days to nonworking time** позначить обраний робочий день як вихідний, а варіант **Set day(s) to these specific working times** – як робочий день, але для нього можна задати новий час роботи та перерви (наприклад, короткий день).

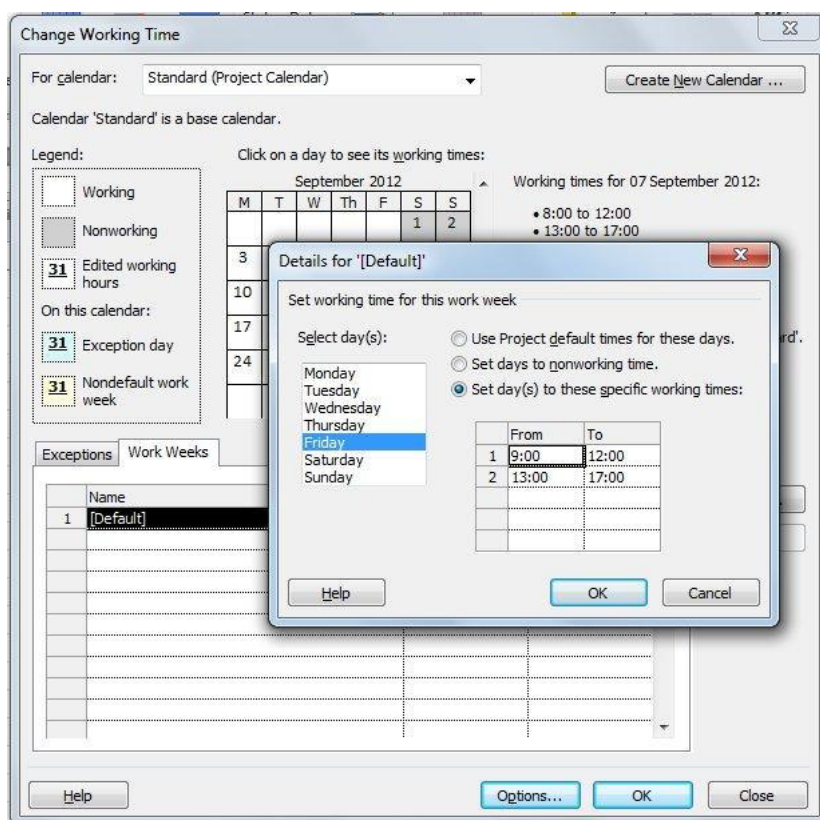


Рисунок А.2 – Діалогове вікно зміни робочого часу в календарі

Для введення відомостей про ресурси потрібно відкрити **Resource Sheet** через меню **Resource** → **Team Planner**. Дані можна починати безпосередньо вводити в таблицю, що з'явилася (рис. А.3). Після введення назви ресурсу йому автоматично привласнюється коротка назва, але її можна змінити на свій розсуд.

newTechnology.mpp - Microsoft Project													
Resource Sheet Tools													
File	Task	Resource	Project	View	Format								
Team Planner View	Assign Resources	Resource Pool Assignments	Substitute Resources	Add Resources Insert	Information	Notes	Details	Level Selection	Level Resource	Level All	Leveling Options	Clear Leveling	Next Overallocation
Timeline	Start Sat 01.09.12												
	Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt. Rate	Cost/Use	Accrue At	Base Calendar	Code	
1	Учені	Work		У			2 100,00€/hr	120,00€/hr	5 000,00€	End	Standard		
2	Соціолог	Work		С			3 100,00€/hr	120,00€/hr	0,00€	End	Standard		
3	Менеджер	Work		М			2 80,00€/hr	90,00€/hr	0,00€	Prorated	Standard		
4	Програміст	Work		П			2 200,00€/hr	500,00€/hr	20 000,00€	End	Standard		
5	Наладчик	Work		Н			1 100,00€/hr	150,00€/hr	0,00€	Prorated	Standard		
6	Технолог	Work		Т			3 120,00€/hr	150,00€/hr	0,00€	Prorated	Standard		
7	Психолог	Work		П			1 100,00€/hr	200,00€/hr	2 000,00€	End	Standard		

Рисунок А.3 – Аркуш ресурсів

Для зручності введення можна скористатися діалоговим вікном, яке викликається подвійним щикликом по будь-якому осередку рядка потрібного ресурсу (рис.А.4).

Resource Information

General Costs Notes Custom Fields

Resource name: Initials:

Email: Group:

Code:

Booking type: Type:

Material label:

Default Assignment Owner:

Resource Availability

NA	Available From	Available To	Units
NA	NA	NA	2

☐ Generic ☐ Budget ☐ Inactive

Рис. А.4. Діалогове вікно введення даних про ресурс

Важлива властивість ресурсів – їх вартість (витрати) використання в проєкті (рис. А.5). В MS Project є два типи вартості ресурсів: погодинна ставка (**Standard Rate**) і вартість за використання (**Per Use Cost**).

Resource Information

General Costs Notes Custom Fields

Resource Name:

Cost rate tables

For rates, enter a value or a percentage increase or decrease from the previous rate. For instance, if a resource's Per Use Cost is reduced by 20%, type -20%.

A (Default) B C D E

Effective Date	Standard Rate	Overtime Rate	Per Use Cost
--	100,00\$/h	200,00\$/h	2 000,00\$

Cost accrual:

Рисунок А.5 – Вкладка **Costs** діалогового вікна введення даних про ресурс

Погодинна ставка виражається у вартості використання ресурсу в одиницю часу, наприклад, 50 у.о. за годину. В цьому випадку вартість участі ресурсу в проєкті складає час, протягом якого він працює в проєкті, помножений на погодинну ставку. Також в цьому випадку передбачається оплата за понаднормові часи роботи.

Величина **Per Use Cost** позначає вартість використання обладнання або співробітника в задачі, що не залежить від того, скільки часу задіяний в задачі ресурс. Загальні витрати на використання ресурсу визначаються шляхом множення вартості використання на число задач, у яких він задіяний.

Для ресурсу може бути зазначена вартість як одного із двох типів, так і обох типів. При визначенні загальних витрат на використання ресурсу погодинні витрати та витрати на використання підсумуються.

План робіт найкраще складати в поданні **Gantt Chart** (меню **View**). Для додавання задачі (роботи) в план проєкту потрібно в таблицю ліворуч від діаграми послідовно ввести назву задачі, її тривалість тощо (рис. А.6). Для першої задачі дата початку автоматично призначається рівній даті початку проєкту, для всіх наступних задач дата початку автоматично визначається залежно від тривалості, задач-попередників, задіяних ресурсів. Після введення відомостей про задачу праворуч на діаграмі з'явиться відрізок, що відображає задачу.

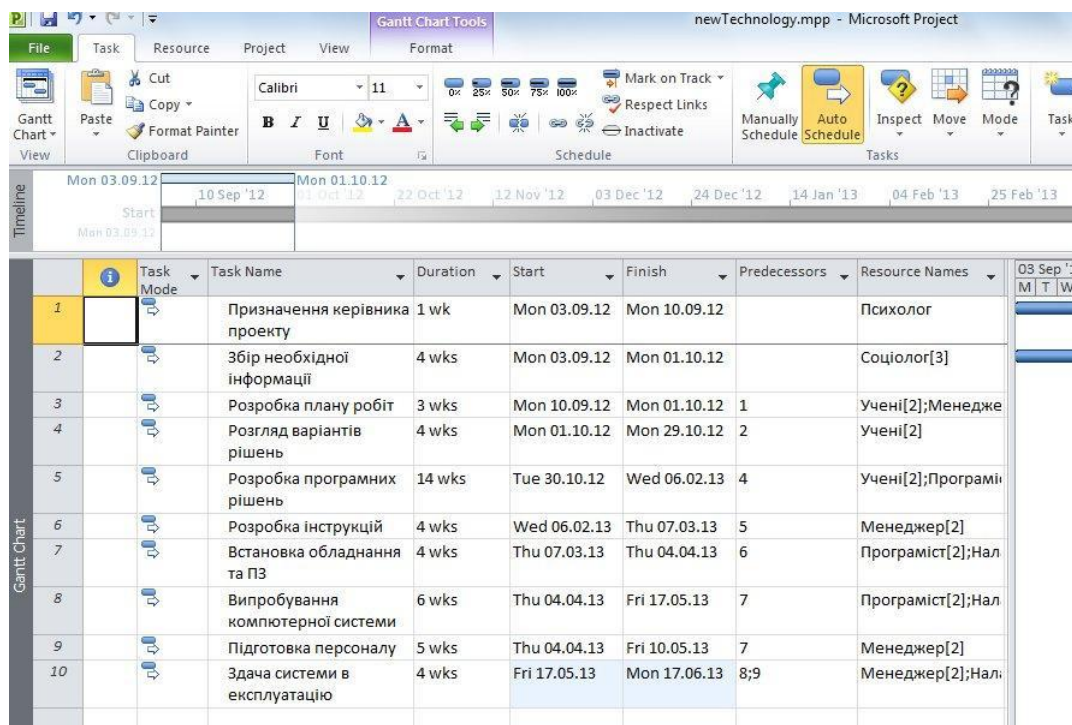


Рисунок А.6 – Діаграма Ганта для введення відомостей про роботи

Для зручності введення даних про задачу можна використати діалогове вікно, що викликається подвійним щикликом по будь-якому осередку рядка потрібної задачі.

На вкладці **General** (рис.А.7) можна ввести назву задачі та її тривалість. У полі **Priority** задається значення, що показує важливість і доступність задачі для вирівнювання завантаження ресурсів. Першими затримуються або розділяються задачі з більше низьким пріоритетом.

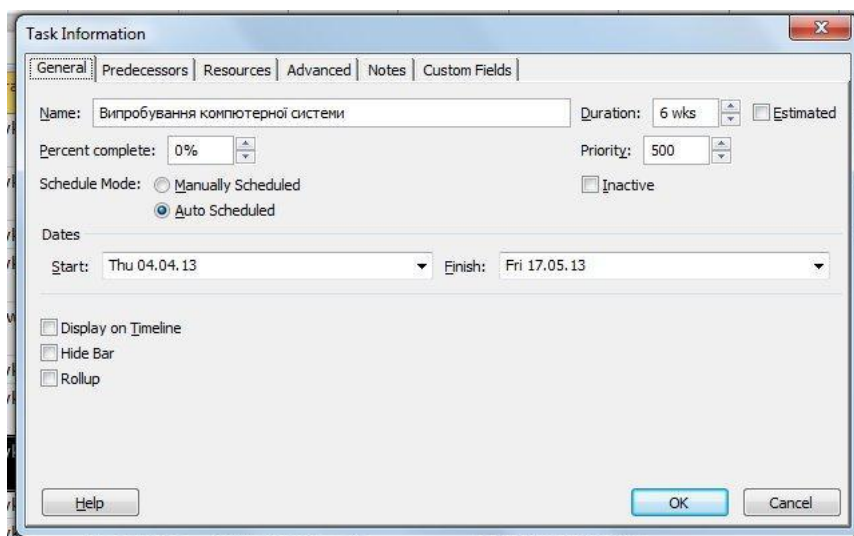


Рисунок А.7 – Вкладка **General** діалогового вікна введення відомостей про задачу

На вкладці **Predecessors** задаються задачі-попередники, для вибору задачі поставте курсор в осередок стовпця **Task Name** і виберіть відповідну назву задачі зі списку, що розкривається. При цьому автоматично визначається номер цієї задачі в списку. Якщо ж ви вирішили вводити дані про задачі без діалогового вікна, то при введенні декількох задач-попередників їх необхідно розділити крапкою з комою.

Відомості про задіяні ресурси аналогічно вводяться на вкладці **Resources** з вказівкою одиниць (кількості) задіяного ресурсу.

При проектуванні розрізняють задачі та фази. Будь-яка задача автоматично стає фазою, як тільки в неї з'являється вкладена задача. Доти поки в задачі немає вкладених задач, вона не є фазою. Додавання в план фази нічим не відрізняється від додавання задачі. Щоб помістити задачу на більше низький рівень структури, потрібно виділити цю задачу та скористатися командою меню **Schedule** → **Indent Task**. Для переміщення задачі на попередній рівень структури потрібно у меню використати команду **Schedule** → **Outdent Task**.

Зв'язок між двома задачами визначає, яким образом час початку або завершення однієї задачі впливає на час початку або завершення іншої. Один зв'язок може поєднувати тільки дві задачі. Але при цьому в одній задачі може бути кілька зв'язків з іншими задачами.

В MS Project є чотири типи зв'язків між задачами:

- 1) зв'язок типу **Finish-to-Start (FS)**, – найпоширеніший тип залежності між задачами, при якій задача Б не може початися, поки не завершена задача А;
- 2) зв'язок типу **Start-to-Start (SS)** позначає залежність, при якій задача Б не може початися доти, поки не почалася задача А. За допомогою такого зв'язку звичайно поєднуються задачі, які повинні виконуватися майже одночасно;
- 3) зв'язок типу **Finish-to-Finish (FF)** позначає залежність, при якій задача Б не може закінчитися доти, поки не закінчилася задача А. Звичайно таким зв'язком поєднуються задачі, які повинні виконуватися майже одночасно, але при цьому одна не може закінчитися, поки не завершена інша;
- 4) зв'язок типу **Start-to-Finish (SF)** позначає залежність, при якій задача Б не може закінчитися доти, поки не почалася задача А. Звичайно такий зв'язок використовується в тому випадку, коли А є задачею з фіксованою датою початку, яку не можна змінити. У такому випадку дата початку наступної задачі не змінюється при збільшенні тривалості попередньої.

При створенні зв'язку як усталено створюється зв'язок типу (FS), при цьому наступна задача на календарному плані займе місце після попередньої задачі.

Діаграма Ганта – це один з найбільш популярних способів графічного подання плану проєкту, застосовуваний у багатьох програмах керування проєктами. З діаграми можна одержати інформацію про послідовності задач, їхньої відносної тривалості й тривалості проєкту в цілому.

Діаграма Ганта є графіком, на якому по горизонталі розміщена шкала часу, а по вертикалі розташований список задач (рис. А.8). Довжина відрізків, що позначають задачі, пропорційна тривалості задач. Поруч із відрізками може відображатися додаткова інформація, наприклад, назви задіяних у них ресурсів. Склад діаграми визначається її налаштуванням.

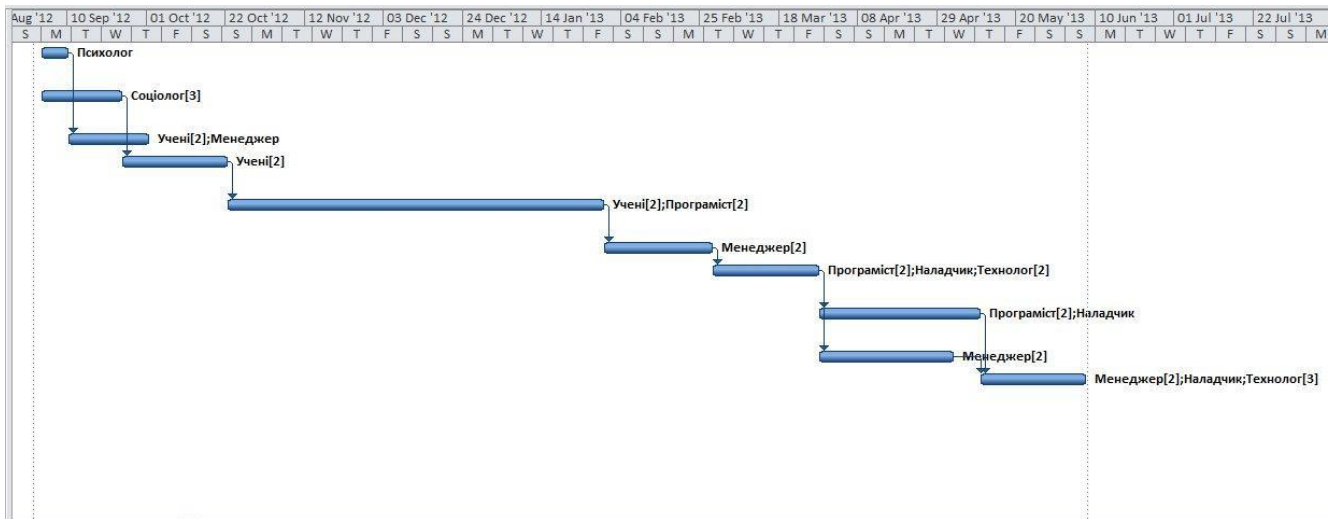


Рисунок А.8 – Діаграма Ганта

За допомогою діаграми Календар можна представити план робіт у звичному для більшості співробітників виді. Ця діаграма відображає інформацію про план проєкту у вигляді таблиці з сім'ю або п'ятьма колонками, що відповідають дням тижня, і множиною рядів, що відповідають тижням. Задачі на календарі позначені відрізками, які починаються в день початку робіт над завданням і закінчуються в день їхнього закінчення (рис. А.9). Робота із задачами (зміна строків, створення, видалення тощо) здійснюється так само, як і на діаграмі Ганта.

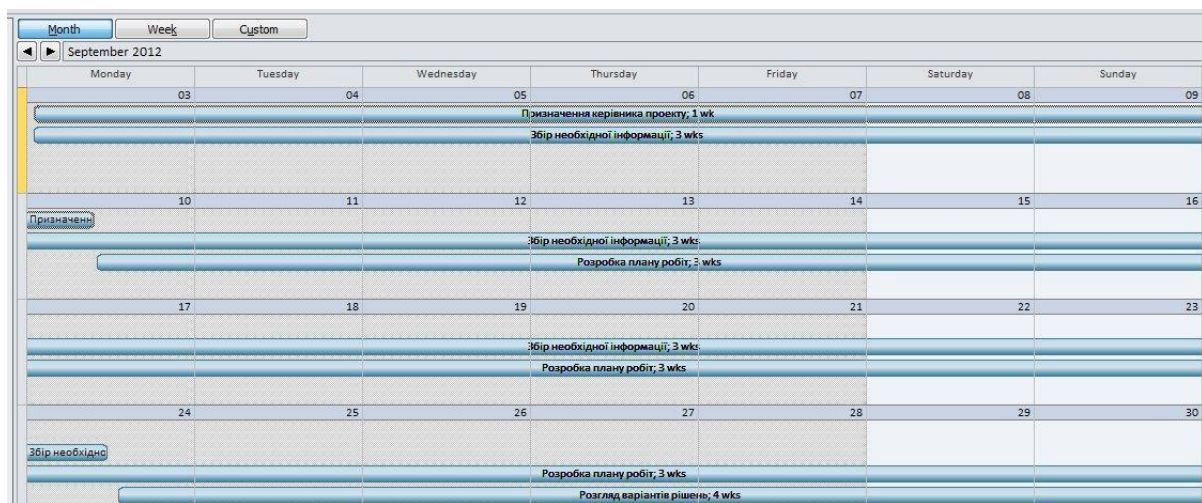


Рисунок А.9 – Фрагмент діаграми Календар

Для визначення критичного шляху в проєкті MS Project відкрийте подання Діаграма Ганта та на вкладці **Format** встановіть позначку **Critical Tasks**. Після цього критичний шлях на діаграмі (послідовність критичних задач) виділитися червоним кольором (рис. А.10).

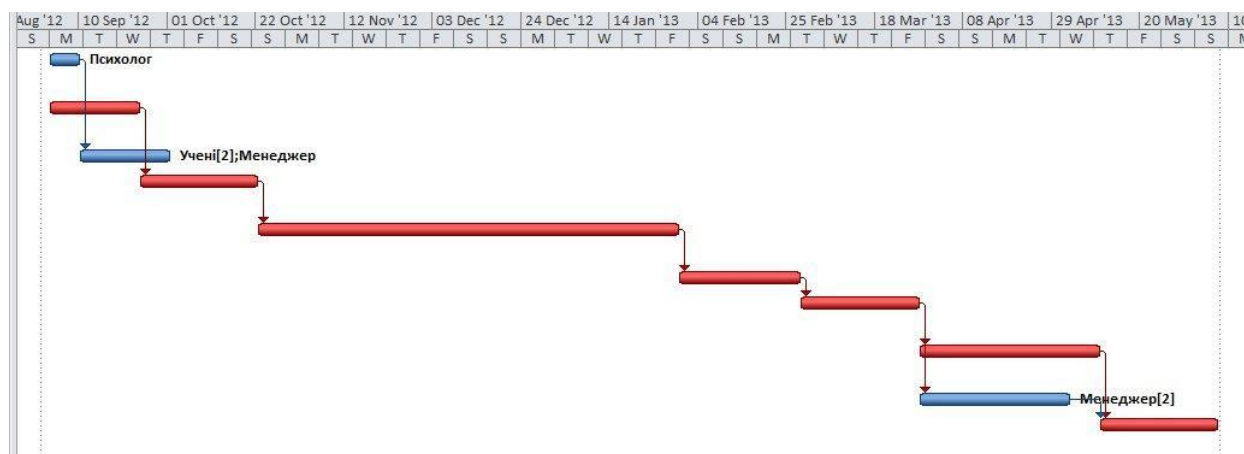


Рисунок А.10 – Діаграма Ганта з відображенням критичного шляху

Щоб визначити рівномірність завантаження ресурсів, потрібно відкрити подання **Resource Sheet**. У ньому всі ресурси, завантаження яких перевищує їх доступність, виділені червоними кольорами, а в колонці і поруч з назвами відображається спеціальний значок (рис. А.11).

		Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt. Rate	Cost/Use	Accrue At	Base Calendar
1	⚠	Учені	Work		У		2	100,00€/hr	120,00€/hr	5 000,00€	End	Standard
2		Соціолог	Work		С		3	100,00€/hr	120,00€/hr	0,00€	End	Standard
3		Менеджер	Work		М		2	80,00€/hr	90,00€/hr	0,00€	Prorated	Standard
4		Програміст	Work		Пр		2	200,00€/hr	500,00€/hr	20 000,00€	End	Standard
5		Наладчик	Work		Н		1	100,00€/hr	150,00€/hr	0,00€	Prorated	Standard
6		Технолог	Work		Т		3	120,00€/hr	150,00€/hr	0,00€	Prorated	Standard
7		Психолог	Work		Пс		1	100,00€/hr	200,00€/hr	2 000,00€	End	Standard

Рисунок А.11 – Перевищення доступності ресурсів

Перевищення доступності ресурсу полягає в тому, що для виконання призначеної роботи ресурсу потрібно більше часу, чим у нього є, тобто ресурсу призначено більше задач, чим він може виконати в доступний робочий час. Існує кілька причин, здатних привести до цього. Найпоширенішої серед них є призначення ресурсу на задачі, виконання яких повністю або частково здійснюється одночасно. Іншим варіантом може бути збільшення обсягу робіт задачі, що привело до перевищення припустимого рівня завантаження ресурсу. Також призначення ресурсу через зміни в плані може доводитися на дні, коли ресурс недоступний.

Вирівнювання завантаження ресурсів – це усунення конфліктів ресурсів або перевищення доступності ресурсів. Коли Project виконує вирівнювання завантаження ресурсів, то вибрані призначення ресурсу розподіляються і переплануються.

Вирівняти завантаження ресурсів можна кількома способами. По-перше, зменшивши обсяг роботи перевантажених ресурсів, скоротивши деякі задачі або призначивши інших співробітників на їхнє виконання. По-друге, позбувшись від перетинання задач, вставивши в розклад перерви в задачах або призначеннях, або змінивши дати їх початку і закінчення. Нарешті, урахувавши роботу, виконувану ресурсом поверх норми, як понаднормову.

Для вирівнювання завантаження ресурсів в Microsoft Project можна скористатися автоматизованими засобами, а можна перерозподілити завантаження вручну.

Вирівнювання ресурсів здійснюється після визначення критичного шляху на діаграмі Ганта. Для ручного вирівнювання можна зменшити тривалості задач, причому почати варто з тих, що лежать на критичному шляху. При цьому слід пам'ятати, що скорочення тривалості задач може не тільки забрати їх із критичного шляху, але і зробити критичною іншу задачу. Для скорочення тривалості задачі можна застосувати кілька методів: по-перше, можна скоротити обсяг роботи, необхідний для її виконання. По-друге, можна додати ресурси для прискорення виконання роботи при збереженні її обсягу. Нарешті, можна розбити задачу на підзадачі, виконувані одночасно різними співробітниками.

Для автоматичного вирівнювання ресурсів виконайте **Format → Leveling Option**. Щоб прийняти налаштування за замовчуванням (рис. А.12), натисніть кнопку **OK**.

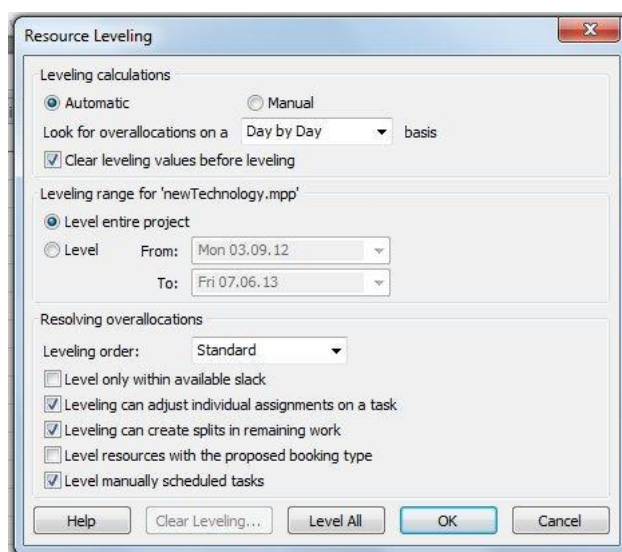


Рисунок А.12 – Діалогове вікно вирівнювання завантаження ресурсів

Порядок вирівнювання, обраний у діалоговому вікні **Leveling order**, визначає, які задачі повинні бути затримані або перервані в першу чергу. При **Standard** порядку вирівнювання Project перевіряє залежності попередників, часовий резерв, дати, обмеження та пріоритети. При вирівнюванні **ID Only** Project просто проходить за списком задач у порядку, заданому їх ідентифікаторами. При вирівнюванні **Priority**, **Standard** Project спочатку перевіряє встановлені пріоритети, першими затримуються або розділяються задачі з нижчим пріоритетом, а потім бере до уваги стандартні фактори.

Мережний графік є ще одним популярним засобом візуалізації плану проєкту. На ньому задачі представлені у вигляді блоків, з'єднаних стрілками в блок-схему відповідно до взаємозв'язків завдань у плані проєкту.

Для перегляду мережного графіка виконайте **View → Network Diagram** (рис. А.13). Блоки графіка розрізняються кольорами з формою залежно від типу задачі (звичайна задача, завершальна задача або фаза) і її стану (виконується, не виконується, завершена). У блоці може бути зазначена додаткова інформація, наприклад, назва задачі, дати початку з закінчення, тривалість і задіяні ресурси. На мережному графіку паралелограмами позначені фази, прямокутниками – задачі, а шестикутником – завершальні задачі. Початі задачі перекреслені однією лінією, а завершені – двома. Стрілки відповідають зв'язкам між задачами.

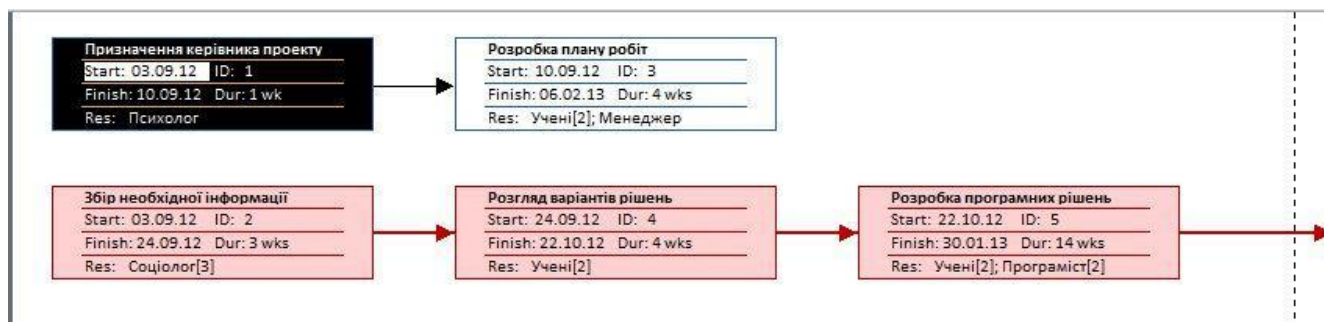


Рисунок А.13 – Фрагмент мережного графіка

Подвійний щиглик по блоку відкриває діалогове вікно редагування даних про задачу. Також можна відредагувати тип зв'язку між задачами, для цього потрібно виконати подвійного щиглика мишею по лінії зв'язку.

Навчальне видання

Методичні рекомендації до практичних занять з дисципліни «Менеджмент проєктів програмного забезпечення» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 121 – Інженерія програмного забезпечення

Укладач: Віра Вікторівна Любченко