

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра прикладної математики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Визначення економічних параметрів виконання проєктів, що представлені  
мережею взаємопов'язаних робіт

(тема)

Виконав:

студент 2 курсу, групи ПМм-22-1

Закутній С.В.

(прізвище, ініціали)

Спеціальність 113 Прикладна математика

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва освітньої програми)

Керівник доц. Матвієнко О.І.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ

(підпис)

Сидоров М.В.

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна математика

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ \_\_\_\_\_

(підпис)

“ 06 ” листопада 2023 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Закутньому Сергію Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Визначення економічних параметрів виконання проєктів,  
що представлені мережею взаємопов'язаних робіт

затверджена наказом по університету від 2 листопада 2023 р. № 1276 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 січня 2024 р.

3. Вихідні дані до роботи дані про час виконання робіт проєктів

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_

1. Актуальність теми роботи \_\_\_\_\_

2. Постановка задачі \_\_\_\_\_

3. Аналіз предметної області \_\_\_\_\_

4. Метод чисельного аналізу \_\_\_\_\_

5. Результати обчислювального експерименту \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	6 – 12 листопада 2023 р.	виконано
2	Вибір та обґрунтування методу	13 – 26 листопада 2023 р.	виконано
3	Розробка алгоритму і програми	27 листопада – 10 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	11 грудня – 24 грудня 2023 р.	виконано
5	Робота над текстом пояснювальної записки	25 грудня 2023 р. – 9 січня 2024 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2024 р.	виконано

Дата видачі завдання 6 листопада 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Матвієнко О.І.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 77 с., 8 табл., 16 рис., 1 дод., 16 джерел.

ТЕОРІЯ ГРАФІВ, МЕТОДИ НЕЧІТКОЇ ЛОГІКИ, МЕТОД ДІАГРАМ ГАНТА, КРИТИЧНИЙ ШЛЯХ, ФУНКЦІЯ ПРИНАЛЕЖНОСТІ, НЕЧІТКА ЦІЛЬ, ВАРТІСТЬ ПРОЄКТУ, ЧАС ВИКОНАННЯ ПРОЄКТУ. ЕКОНОМІЧНА ХАРАКТЕРИСТИКА ПРОЄКТУ, МЕТОД ПЛАНУВАННЯ AGILE.

Об'єкт дослідження – мережа взаємопов'язаних між собою робіт.

Мета роботи – визначення економічних параметрів проєкту за допомогою теорії графів та теорії нечітких множин.

Методи дослідження – визначення критичного шляху за допомогою методу СРМ, застосування методу до проєкту створення сайту електронної комерції, аналіз отриманих результатів.

В кваліфікаційній роботі проведені дослідження, що сприяють знаходженню основних економічних параметрів проєкту таких як час виконання проєкту, його надійність та вартість проєкту, що можуть бути імплементовані у реальному житті за допомогою теорії графів та теорії нечіткої логіки. Було створено програмний продукт мовою програмування Python, який допомагає швидко знайти критичний шлях проєкту та вивести результати дослідження, а також візуалізовано ці результати у вигляді графу за допомогою програми Gephi.

## ABSTRACT

Introductory note: 77 pages, 8 tables, 16 figures, 1 appendix, 16 sources.

GRAPH THEORY, FUZZY LOGIC, GANTT CHART, CRITICAL PATH METHOD, CPM, FUZZY GOAL, PROJECT COST, PROJECT DURATION, ECONOMIC CHARACTERISTICS OF THE PROJECT, AGILE.

Object of research – network of interconnected works.

Purpose of work – defining the economic parameters of a project using graph theory and fuzzy set theory.

Methods of research – determining the critical path using the Critical Path Method (CPM), applying the method to the e-commerce website creation project, and analyzing the obtained results.

The work involved conducting research contributing to the identification of key economic parameters of the project, such as project completion time, reliability, and cost. This was achieved through the application of graph theory and fuzzy logic. A software product was developed using the Python programming language to quickly identify the critical path of the project and present research results. Additionally, these results were visualized in the form of a graph using the Gephi software. The developed tool can be implemented in real-life scenarios to facilitate project analysis and decision-making.

## ЗМІСТ

	С.
Вступ .....	8
1 Аналіз предметної області та постановка задач дослідження .....	10
1.1 Основні поняття теорії нечітких множин .....	10
1.2 Застосування теорії графів для планування проєкту .....	15
1.3 Змістовна та формальна постановка задачі .....	19
1.3.1 Змістовна постановка задачі .....	19
1.3.2 Формальна постановка задачі .....	20
1.4 Постановка задач дослідження .....	23
2 Вибір та обґрунтування методу розв’язання .....	24
2.1 Огляд методів знаходження оптимального плану проєкту в умовах визначеності термінів та обсягу робіт .....	24
2.2 Метод критичного шляху (Critical Path Method) .....	25
2.3 Метод діаграм Ганта .....	27
2.4 Метод PERT (Project Evaluation and Review Technique) .....	29
2.5 Метод планування Agile (Scrum Kanban) .....	31
Висновки за розділом 2 .....	34
3 Програмна реалізація .....	35
3.1 Вибір мови програмування .....	35
3.2 Бібліотека NetworkX для роботи з графами .....	37
3.3 Бібліотека GraphViz як засіб візуалізації даних .....	39
3.4 Gephi .....	40
3.5 Опис програми .....	41
Висновки за розділом 3 .....	42
4 Результати обчислювального експерименту .....	44
4.1 Обчислення економічних характеристик мережевого графіку для проєкту створення Backend-частини сайту електронної комерції .....	44

4.2 Обчислення економічних характеристик мережевого графіку для модифікованого проєкту створення Backend-частини сайту електронної комерції.....	55
4.3 Розрахунок та порівняння вартості проєктів.....	64
Висновки за розділом 4 .....	68
Висновки .....	70
Перелік джерел посилання .....	72
Додаток А Лістинг програми .....	74

## ВСТУП

**Актуальність теми.** У сучасному світі оптимізація грає одну з провідних ролей для подальшого розвитку будь-якої організації чи проєкту, тому правильне і доцільне використання ресурсів може призвести до стрімкого покращення процесів та підвищення прибутку, а також значно посилити конкурентоспроможність. Задля того, щоб мати можливість оцінки стану проєкту та приблизно визначити параметри його виконання, необхідно побудувати систему, яка б могла розрахувати, як саме потрібно розподілити ресурси для найбільш оптимального плану розробки.

Нам вже неможливо уявити життя людини без інформаційних технологій, а тим паче для організації чи підприємства. Для розвитку та процвітання будь-якої компанії вкрай необхідні новітні технологічні інструменти продажу та реклами її продукції або розповсюдження будь-яких послуг. Саме тому створення сайту електронної комерції стає одним з найважливіших для подальшого розвитку компанії. Ця проблема є дуже цікавою для розгляду та потребує детального та комплексного аналізу.

**Мета і завдання кваліфікаційної роботи.** Метою кваліфікаційної роботи є навчитися прогнозувати терміни виконання проєкту та його основні економічні параметри за допомогою теорії графів та методів нечіткої логіки. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі «Визначення економічних параметрів виконання проєктів, представлених мережею взаємопов'язаних робіт»;
- обрати оптимальний метод для пошуку найкращого розв'язку поставленої задачі;
- створити програмний модуль для підрахунку економічних параметрів та графічного відображення розв'язку.

*Об'єктом дослідження* є мережа взаємопов'язаних між собою робіт одного проєкту, у даному випадку це створення сайту електронної комерції.



*Предметом дослідження є можливість розрахунку основних властивостей або параметрів проєкту, прогнозування термінів його виконання та можливість врахування форс-мажорних ситуацій.*

**Методи дослідження.** У кваліфікаційній роботі використовуються методи нечіткої логіки та методи теорії графів, серед методів прогнозування термінів проєкту також розглядаються методи СРМ (метод критичного шляху), метод діаграм Ганта, Project Evaluation and Review Technique (PERT), Agile-методи (Scrum та Kanban).

**Публікації.** Результати, отримані у кваліфікаційній роботі, були представлені на 27-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (м. Харків, 10-12 травня 2023 р.) [1].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Основні поняття теорії нечітких множин

Якщо брати до розгляду класичну множину  $A$ , то вона завжди є строго обмеженою. Для кожного об'єкту  $a$  завжди справедливе одне з тверджень:

- а)  $a$  належить множині  $A(a \in A)$ ;
- б)  $a$  не належить множині  $A(a \notin A)$ .

У цьому випадку ми притримуємося законів двозначної логіки, коли допускаються тільки два можливих вирази: істинний чи хибний. Тобто ми можемо стверджувати, що у традиційній математичній логіці вираз має лише два можливих результати, що суперечить стану реального життя, тому що насправді для того, щоб дати судження тій чи іншій дії ми можемо використовувати багато різних оцінок. Саме тому Лотфі Заде у 1965 р. створив теорію нечітких множин, за допомогою якої ми і можемо оперувати багатьма оцінками щодо конкретного явища. За думкою Заде, використання нечітких множин може сприяти створенню необхідного формального інструменту для моделювання людських міркувань та способу, яким люди приймають рішення.

Поняття нечіткої множини – це спроба математичної формалізації нечіткої інформації для створення математичних моделей. В основі цього поняття лежить ідея, що елементи, які складають дану множину і мають спільну властивість, можуть мати цю властивість в різній мірі і, отже, належати до даної множини з різним ступенем приналежності. За такого підходу, коли ми говоримо, що певний елемент належить даній множині, необхідно вказати, з яким ступенем цей елемент задовольняє властивостям даної множини.

Людина часто може оцінювати певний показник, наприклад, площу, температуру тіла (у квадратних метрах), в термінах «низької» (відповідає цифрі 1), «звичайної або нормальної» (цифра 2), «високої» (цифра 3) та «дуже

високої» (цифра 4). У теорії нечітких множин пропонується віднесення конкретної пожежі до одного з цих класів з використанням характеристичної функції приналежності елементу до певного класу (рис. 1.1).

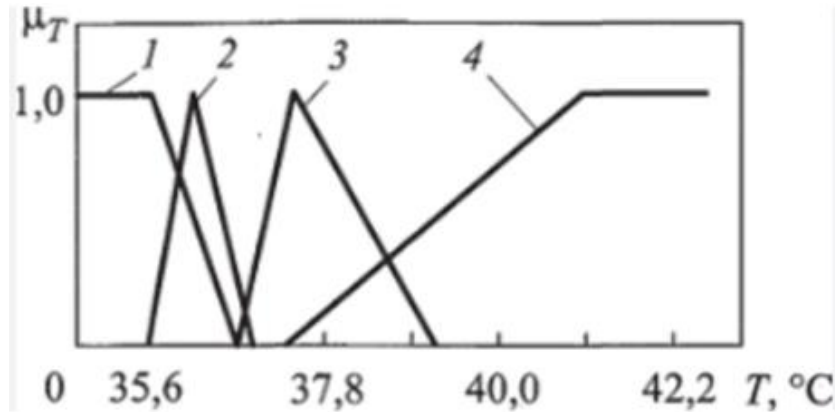


Рисунок 1.1 – Класи множин на прикладі графіку характеристичної функції

Тобто Заде розширив традиційну логіку приналежності множині для того, щоб врахувати декілька ступенів приналежності цій множині на відрізку від  $[0,1]$ , де значення на кінцях цього відрізка відповідають або ж повній відсутності цієї характеристики у розглядуваного елемента, або ж, навпаки, повній приналежності елемента до якоїсь множини. Ступінь оцінки приналежності цій множині буде характеризувати функція приналежності (або характеристична функція  $\mu_A$ ).

Такі класи також можна називати інформаційними гранулами [2]. Звичайно, кількість розглядуваних класів можна збільшувати відповідно до вимог, але згідно з дослідженнями, одночасно людина може оперувати відразу п'ятьма-дев'ятьма факторами, тому число таких класів (гранул) необхідно задавати у цих границях.

Математичний апарат теорії нечітких множин дасть можливість трактувати будь-які судження людини, а отже цей підхід може бути широко використаний на практиці для розв'язання різних задач, як технічних, так і гуманітарних. Теорія нечітких множин є одним з інструментів для керування складними системами в умовах неповної або нечіткої інформації.

Щоб краще зрозуміти головну ідею можемо перейти до системи оцінювання знань студентів. Наприклад у п'ятибальній системі ми вводимо п'ять можливих класів оцінок – це «відмінно», «добре», «задовільно», «незадовільно» та «погано». Але у 12-бальній (або болонській 90-бальній) системі одна й та ж сама оцінка (наприклад, «відмінно») може бути задана у ширшому діапазоні, що дозволяє більш точно оцінити досягнення студента.

З цього прикладу ми можемо зробити висновок, що зі збільшенням класів зростає точність дослідження, але і визначення правильного класу стає більш складною задачею.

У неформальному вигляді нечітка множина – це клас об'єктів, для якого не існує чіткої межі між об'єктами, які приналежать до цього класу, і тими, які до нього не приналежать [3].

Нечіткою множиною  $A$  на універсальній множині  $X$  називається сукупність пар  $(\mu_A(x), x)$ , де  $\mu_A(x)$  – це ступінь приналежності елемента  $x$  нечіткій множині  $A$ . Тобто, використовуючи наш приклад про оцінки студентів, ми можемо сказати, що множина  $X$  представляє сукупність усіх оцінок, а  $\mu_A(u)$  відображає наскільки ми відносимо успіхи студента до тієї чи іншої оцінки.

Функцію приналежності нечіткої множини  $A$  позначимо  $\mu_A(x)$ , де  $x \in X$ . Нечітко визначена ціль характеризується деякою нечіткою множиною  $\hat{X}_C$  з функцією приналежності  $\mu_C(x)$ . Значення функції приналежності далі можемо визначити як «ступінь надійності» або «надійність».

Виділяють два основних методи побудови функції приналежності – прямий та непрямий. При прямому методі експерти чи група експертів задають значення функції для кожного елемента універсальної множини. При цьому не вимагається дуже точного задання такої функції та достатньо просто фіксації її вигляду та характерних значень. У непрямому методі якщо у формалізованого об'єкта є якісь властивості, то використовують непрямі методи побудови. До них відносять, наприклад, метод попарних порівнянь на кінцевих дискретних множинах.

Побудова функції приналежності є більше психологічною, ніж математичною проблемою, що пов'язана з індивідуальним сприйняттям властивостей таких об'єктів [4].

Доповненням нечіткої множини  $A$  (рис. 1.2) називають нечітку множину  $\bar{A}$ , задану на множині  $X$  з функцією приналежності

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \text{ для всіх } x \in X.$$

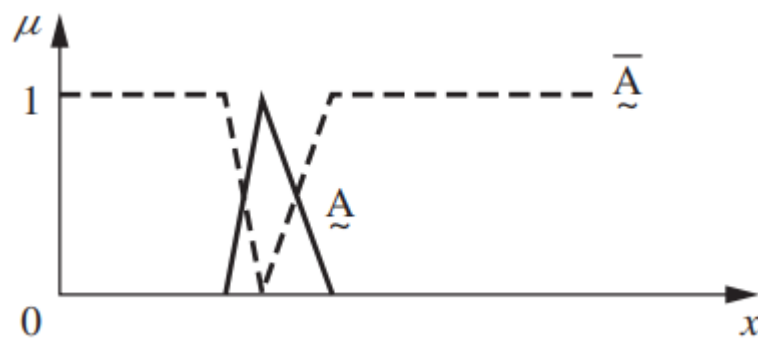


Рисунок 1.2 – Доповнення нечіткої множини

Перетином нечітких множин  $A$  і  $B$  (рис. 1.3), заданих на множині  $X$ , називається нечітка множина  $C = A \cap B$  з функцією приналежності

$$\mu_C(x) = \min\{\mu_A(x), \mu_B(x)\} \text{ для всіх } x \in X.$$

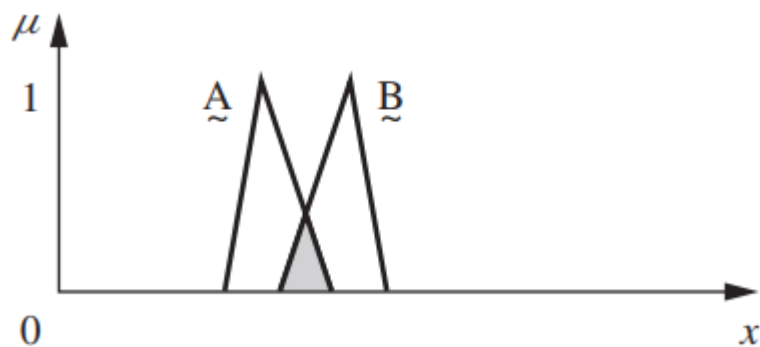


Рисунок 1.3 – Перетин нечітких множин

Об'єднанням нечітких множин  $A$  і  $B$  (рис. 1.4), заданих на множині  $X$ , називається нечітка множина  $D = A \cup B$  з функцією приналежності

$$\mu_D(x) = \max\{\mu_A(x), \mu_B(x)\} \text{ для всіх } x \in X.$$

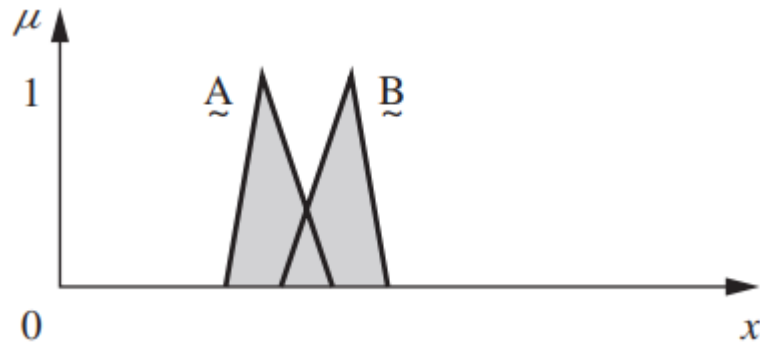


Рисунок 1.4 – Об'єднання нечітких множин

Трикутним нечітким числом  $A$  називається трійка  $\langle a, b, c \rangle$  дійсних чисел, де  $a \leq b \leq c$ ,  $\mu_A(a) = 0$ ,  $\mu_A(b) = 1$ ,  $\mu_A(c) = 0$ . Якщо  $a < b < c$ , то функція приналежності буде визначатися наступним чином:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & \text{якщо } x \in [a, b], \\ \frac{x-c}{b-c}, & \text{якщо } x \in [b, c], \\ 0, & \text{у іншому випадку.} \end{cases}$$

Якщо  $a = b < c$ , то

$$\mu_A(x) = \begin{cases} \frac{x-c}{b-c}, & \text{якщо } x \in (b, c], \\ 0, & \text{у іншому випадку.} \end{cases}$$

Якщо  $a < b = c$ , то

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & \text{якщо } x \in [a, b), \\ 0, & \text{у іншому випадку.} \end{cases}$$

Якщо  $a = b = c$ , то

$$\mu_A(x) = \begin{cases} 1, & \text{якщо } x = b, \\ 0, & \text{у іншому випадку.} \end{cases}$$

Друге число трійки зазвичай називають модою або чітким значенням нечіткого трикутного числа, а числа  $a$  і  $c$  характеризують ступінь нечіткості другого числа [5].

## 1.2 Застосування теорії графів для планування проєкту

Для того, щоб дослідити характеристики проєкту нам потрібно визначити як саме він буде розвиватися, яким буде його план та задачі, а також треба розуміти послідовність цих задач та можливість їх одночасного виконання задля збереження часу та більш швидкої реалізації.

Щоб змодельовати таку послідовність ми можемо використати математичний апарат теорії графів, який допоможе по-перше, розподілити між собою вершини (роботи проєкту), по-друге, дозволить графічно зобразити усі зв'язки між такими роботами, і по-третє, за його допомогою можна знайти критичний шлях у графі, що буде відповідати кінцевому терміну виконання робіт.

Графом називається пара  $G = \langle V, E \rangle$  – це досліджуваний граф, де  $V$  – множина вузлів (vertices);  $E$  – множина ребер (edges).

Під ступенем вершини ми розуміємо число ребер, що виходять із даної вершини, а порядком графа називається число усіх його вершин. Граф є простим, коли будь-які дві його вершини з'єднані лише одним ребром і кожне ребро з'єднує різні вершини. Ребра графа, що мають напрям, називаються дугами, а сам граф, що складається тільки з таких ребер називається орієнтованим. Кратними ребрами називають ті ребра, що з'єднують одну й ту ж пару вершин [6].

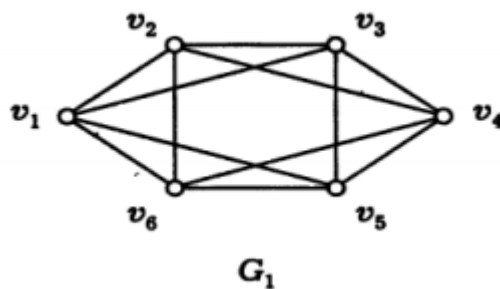


Рисунок 1.5 – Приклад неорієнтованого графа з 6 вершин та 12 ребер

Теорія графів є дуже корисною для розгляду задач планування та керування складними системами і дає змогу відобразити більшу кількість робіт, знайти необхідні зв'язки між ними. Методи теорії графів також дають можливість точно визначити терміни виконання поставленого плану та критичних робіт, тобто тих, від яких залежить виконання усього проєкту в цілому. Зазвичай, для розв'язання задач планування використовується мережевий графік.

Мережевий графік – це граф без кратних ребер, усі ребра якого є зваженими дугами.

Основою системи мережевого планування та керування проєктом є мережева модель – це графічне відображення плану, яке у літературі називається мережевим графіком. Мережевий графік – це схема, на якій чітко показані всі етапи створення спочатку проміжних продуктів з певною готовністю, а потім – завершення проєкту.



Метою використання мережевих графіків є розробка оптимального або досить близького до нього варіанту плану проєкту, який забезпечує раціональне узгодження в часі та просторі виконуваних робіт і найкраще використання ресурсів, а також ефективне управління процесом реалізації цього плану. В результаті використання мережевих графіків скорочується тривалість проєкту, знижується його трудомісткість та собівартість, зростає продуктивність праці.

Кожна окрема робота є ребром (дугою) цього мережевого графіка, також біля неї позначається її вага, тобто ресурс, який необхідний для виконання такої роботи. Це може бути як час, що необхідний на її виконання, так і кількість якихось значущих матеріалів або коштів.

Подія, з якої починається робота, називається початковою, і відповідно кінцевою є вершина, для якої ця робота закінчується.

Довжиною шляху називається сума тривалості усіх робіт, з яких даний шлях складається.

Найдовший шлях називається критичним і він є найважливішим при розгляді задач планування та керування, тому що завдяки ньому ми будемо визначати тривалість усього проєкту. Усі події та роботи, що лежать на цьому шляху називаються критичними і будь-яка затримка, що призводить до збільшення часу виконання або більшої кількості затрачених ресурсів для критичної роботи призводить до збільшення усього комплексу робіт, а отже і усього проєкту загалом.

Шлях, довжина якого трохи менша за критичний, називають підкритичним.

Загальна кількість критичних та підкритичних шляхів утворює критичну зону. Виявлення критичної зони на мережевому графіку дозволяє виділити роботи, на які потрібно звертати увагу при необхідності скорочення строків проєкту під час створення мережевого графіка або при контролі за ходом проєкту.

Мережевий графік будується на базі таблиці з позначенням тривалості кожної роботи та зв'язку цієї роботи з відповідними початковою та кінцевою подіями. Також він не повинен мати циклів та кратних ребер. Мережевий

графік має тільки одну початкову та кінцеву вершину усього проєкту, тобто розгалужень усередині графу може бути безліч, але завжди все повинно як виходити з початкової вершини, так і приходити у кінцеву. Події на мережевому графіку позначаються числами, але варто зазначити, що номер будь-якої події повинен бути менше ніж номер кінцевої події.

Також дуже важливим буде описати параметри можливих подій, на базі яких буде складатися таблиця тривалості робіт та яка використовується для знаходження критичного шляху проєкту.

Для побудови такої таблиці нам буде необхідно знати час виконання робіт (або матеріальні витрати на виконання окремої роботи, в залежності від того, який саме параметр ми хочемо дослідити), а також визначитися з послідовністю цих подій.

Тож нехай  $i, j$  – це номери подій,  $i, j$  – робота для виконання, де  $i$  – її початкова подія,  $j$  – кінцева подія,  $t_{i,j}$  – тривалість роботи  $i, j$ .

Також необхідно ввести деякі інші параметри, пов'язані зі створенням такої таблиці:

–  $t_p i$  – ранній термін події  $i$ , тобто термін, раніше якого ця робота точно не може бути закінчена;

–  $t_n i$  – пізній термін події  $i$ , тобто термін, коли робота точно буде закінчена (цей термін дорівнює різниці між довжиною критичного шляху та довжиною найбільшого шляху, що йде після події  $i$ );

–  $R i$  – резерв події, що дорівнює різниці між пізнім та раннім термінами.

Слід врахувати, що подальша робота не може бути записана в таблицю, якщо не записані усі попередні роботи.

Мережеві графіки є важливим інструментом у багатьох сферах реального життя. Наприклад, в будівництві вони допомагають оптимізувати графік робіт та керувати ним, зменшуючи тривалість та витрати на будівництво. У сфері проєктування та розробки програмних продуктів мережеві графіки

допомагають визначити послідовність завдань та ресурси, необхідні для виконання проєкту. У логістиці вони допомагають визначити найефективніший маршрут доставки товарів. У виробництві вони дозволяють планувати виробничий процес та розподіляти ресурси. Також мережеві графіки використовуються у наукових дослідженнях, управлінні проєктами та багатьох інших галузях, де важливо керувати часом, ресурсами та послідовністю дій для досягнення оптимальних результатів.

### 1.3 Формальна та змістовна постановка задачі

#### 1.3.1 Змістовна постановка задачі

Великі масштаби сучасних розробок вимагають створення систем, які забезпечують можливість оцінки поточного стану та передбачення подальшого ходу розробки проєктів. Такі системи, що базуються на мережевих графіках, отримали назву СРМ (Critical Path Method) та PERT (Project Evaluation and Review Technique). Перші спроби їх використання були здійснені в управлінні будівельними роботами, а також при керуванні розробкою ракети "Поляріс". Зараз ці методи широко використовуються при створенні багатьох складних інженерних об'єктів та споруд [7].

В основі методів управління проєктами лежать методики мережевого планування, які є надзвичайно важливим інструментом для успішного керування проєктами різної складності та масштабу. Ці методики надають змогу ефективно вирішувати різноманітні завдання з метою досягнення успішного завершення проєкту.

Загалом, як зазначалося вище, методи нечіткої логіки та застосування теорії графів є дуже корисними для планування проєктів через те, що дозволяють використовувати свої математичні апарати для дослідження реальних проєктів.

В рамках даної кваліфікаційної роботи буде розглядатися задача пошуку оптимального плану для створення сайту електронної комерції та його подальшого розгортання в онлайн-режимі для кінцевого користувача на Production-середовищі.

Через те, що зараз майже кожна бізнес-організація має свій власний сайт та засоби для продажу продуктів компанії ця задача є однією з найбільш розповсюджених, а отже і вкрай актуальних на сьогоднішній день. Правильне планування для створення такого продукту дозволить не тільки зберегти час та швидше розпочати продаж товарів, але й зберегти матеріальні ресурси компанії.

Тож метою даної роботи буде правильне планування проєкту створення сайту, врахування усього комплексу робіт задля його імплементації та правильний розподіл цих робіт на критичному шляху проєкту, розрахунок приблизних термінів закінчення такого проєкту та бюджету, що необхідний для покриття усіх затрат на його реалізацію.

### 1.3.2 Формальна постановка задачі

Задачу планування та розрахунку економічних параметрів будемо розглядати за допомогою теорії графів та теорії нечіткої логіки, що дозволить прийняти правильне рішення в нечітких умовах та досягнути цілі з певною мірою впевненості.

Розглянемо мережевий графік, у якому тривалість роботи  $i, j$  буде нечітким числом  $d_{ij} = \langle a_{ij}, b_{ij}, c_{ij} \rangle$ , де  $a_{ij}$  – мінімальний час виконання роботи,  $b_{ij}$  – час, коли робота точно буде виконана,  $b_{ij} = c_{ij}$ . Функцію приналежності нечіткого числа  $d_{ij}$  будемо позначати як  $\mu_{ij}(u)$ , де  $u$  – елемент універсальної множини  $U = [0; \infty)$ . Ця величина буде показувати впевненість у тому, що

робота буде дійсно виконана у період між початковою та кінцевою подіями цієї роботи. Необхідно буде оцінити з певною мірою впевненості ступінь досягнення вчасного виконання плану, що і буде основною метою.

Задача зводиться до задачі математичного програмування з нечіткою множиною допустимих планів (через те, що ми самі можемо обирати як саме побудувати весь комплекс робіт від початку до кінця) та нечітко визначеною ціллю.

Множину усіх допустимих планів позначимо через  $\hat{X}$ , а універсальну множину на якій будуть задані усі ці плани –  $X$ . Тоді функцію приналежності нечіткої множини  $\hat{X}$  позначимо  $\mu_{\hat{X}} x$ , де  $x \in X$ .

Нечітко визначена ціль буде формалізуватися нечіткою множиною  $\hat{X}_C$  з функцією приналежності  $\mu_C x$ ,  $x \in X$ .

Усі плани з  $X$  є мережевими графіками, що задані на одному і тому ж графі, але усі вони відрізняються між собою тривалістю робіт. При цьому час роботи  $i, j$  будемо позначати як  $u_{ij}$ .

Для планів із множини  $X$  виконуються нерівності  $a_{ij} \leq u_{ij} \leq b_{ij}$ . Впевненість у тому, що робота  $i, j$  буде виконана за час, що не більший від  $u$ , буде дорівнювати  $\mu_{ij} u$ .

Якщо число дуг у розглядуваному графі дорівнює  $n$ , то кожному допустимому мережевому графіку  $x \in \hat{X}$  поставимо у відповідність вектор значень тривалості робіт  $u x$ :

$$u(x) = (u_{i_1, j_1}, \dots, u_{i_n, j_n}), \quad (1.1)$$

де  $u_{i_k, j_k}$  – чітке значення тривалості роботи  $i_k, j_k$ .

Тоді у відповідності з формулою (1.1)

$$\mu_{\hat{X}}(x) = \min_k \mu_{i_k j_k}(u_{i_k j_k}).$$

Введемо також інші необхідні позначення:

–  $t(x)$  – тривалість критичного шляху;

–  $T^{\min}$  – тривалість критичного шляху у випадку, коли тривалість роботи  $i_k, j_k$ ,  $k = 1, 2, \dots, n$ , дорівнює  $a_{i_k j_k}$ , тобто у тому випадку коли на критичному шляху усі роботи будуть виконуватися за мінімально прогнозований час. Тоді  $u(x^{\min}) = (a_{i_1 j_1}, \dots, a_{i_n j_n})$ ;

–  $T^{\max}$  – тривалість критичного шляху у випадку, коли тривалість роботи  $i_k, j_k$ ,  $k = 1, 2, \dots, n$ , дорівнює  $b_{i_k j_k}$ , тобто усі роботи будуть виконуватися за максимально прогнозований час. Тоді  $u(x^{\max}) = (b_{i_1 j_1}, \dots, b_{i_n j_n})$ .

У якості функції приналежності  $\mu_{\hat{C}}(x)$  допустимого плану  $x$  нечіткій цілі використаємо показник близькості розглядуваного плану до найбільш ефективного плану, тобто плану  $x^{\min}$ , у такому випадку

$$\mu_{\hat{C}}(x) = \frac{T^{\max} - t(x)}{T^{\max} - T^{\min}}. \quad (1.2)$$

Розв'язком задачі будемо вважати досягнення цілі з визначеною мірою впевненості, при цьому враховуючи ступінь виконання обмежень нашої задачі, а тобто ступінь приналежності плану множині  $\hat{X}$ . У такому випадку розв'язком буде перетин нечіткої цілі із множиною допустимих планів або ж нечітка множина  $\hat{D}$  з функцією приналежності

$$\mu_{\hat{D}}(x) = \min\{\mu_{\hat{X}}(x), \mu_{\hat{C}}(x)\}. \quad (1.3)$$

Одним з найбільш популярних та поширених варіантів вибору оптимального плану є вибір тієї альтернативи, що має максимальну ступінь приналежності розв'язку, тобто альтернативи  $x^0$ , для якої

$$\mu_{\hat{D}}(x^0) = \max_{x \in X} \mu_{\hat{D}}(x) = \max_{x \in X} \min \{ \mu_{\hat{X}}(x), \mu_{\hat{C}}(x) \}. \quad (1.4)$$

#### 1.4 Постановка задач дослідження

Після аналізу поставленої задачі та її детального розбору необхідно визначити певні цілі, які потрібно буде задовольнити протягом усього дослідження. До них можна віднести наступне:

- побудувати мережевий графік, на якому будуть вказані усі події та роботи даного проєкту;
- створити таблицю усіх необхідних для проєкту робіт із вказанням економічних параметрів, що необхідні для виконання даної роботи (час, кількість затрачених ресурсів, грошей тощо);
- на основі даних таблиці та за допомогою теорії нечіткої логіки та теорії графів знайти критичний шлях, розрахувати значення економічних параметрів та робіт, що лежать на цьому шляху;
- обрати інструменти програмної розробки та розробити програмний продукт для автоматичного підрахунку досліджуваних економічних параметрів та відобразити критичний шлях на мережевому графіку задля покращеної візуалізації результатів;
- визначити план робіт з оптимальною мірою впевненості, для того, щоб мінімізувати ризики і спрогнозувати приблизний час виконання проєкту із врахуванням множини можливих планів та обмежень;
- зробити висновки про дану роботу та проаналізувати можливі застосування проведених досліджень до реального життя.

## 2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

### 2.1 Огляд методів знаходження оптимального плану проєкту в умовах визначеності термінів та обсягу робіт

Не секрет, що в умовах швидкої зміни технологій та постійних коливань економічної сфери в умовах кризи та розквіту компаніям дуже важко заздалегідь спланувати розвиток свого бізнесу та проєкту. Такі зміни потребують постійної гнучкості та адаптивності до нових реалій і ефективного планування на ранньому етапі скорочує ризики невдачі та забезпечує успішне завершення, роблячи його однією з ключових складових сучасного управління проєктами.

Планування дає змогу чітко визначити завдання та вимоги для досягнення цілей проєкту, дозволяє визначити необхідні ресурси, що допомагає уникнути зайвих витрат і забезпечити ефективне управління фінансами, допомагає ідентифікувати потенційні ризики та розробити стратегії їх управління, зменшуючи негативний вплив на проєкт. За допомогою ефективного планування дуже легко комунікувати із усіма розробниками щодо подальших дій та впровадження нового функціоналу, що забезпечує розуміння та узгодженість всіх членів команди.

Будь-який менеджер чи розробник захоче оптимізувати свої зусилля задля успішної реалізації завдання чи проєкту, саме для цього було створено декілька методів, що можуть бути застосовані для правильного планування. До таких методів відносять метод критичного шляху (CPM), що дозволяє виділити критичні задачі на шляху створення проєкту та дозволяє оптимізувати його розклад. Це також і метод діаграм Ганта, що представляє собою візуальне представлення проєкту у вигляді стовпців, що показує тривалість окремих робіт. Метод програмування проєктів (Project Scheduling) включає в себе математичне моделювання та оптимізацію проєкту з врахуванням різноманітних обмежень.



Метод PERT являє собою статистичний метод для оцінки часу виконання задач проєкту та визначення ймовірностей завершення робіт у заданий термін. Метод Agile є підходом до керування проєкту, що базується на гнучкості та ітеративності у процесі розробки і добре підходить до проєктів, що постійно змінюються та розвиваються у процесі виконання робіт.

Розглянемо ці методи більш детально та спробуємо визначити, який саме підійде найкраще для розрахунку економічних параметрів проєкту та вибору оптимального плану.

## 2.2 Метод критичного шляху (Critical Path Method)

Створений у кінці 1950-х років Морганом Волкером та Джеймсом Келлі метод критичного шляху використовує послідовність і тривалість робіт для визначення загальної тривалості проєкту. Загалом метод створює два можливих типи інформації – кількість запасу часу або вільного часу для кожної активності, а також сам критичний шлях, тобто найдовший шлях на графіку.

Теоретично кажучи, для проєкту може існувати і декілька критичних шляхів, але зазвичай, найдовший та критичний завжди один. Якщо завдання має вільний час, це означає, що початок виконання цього завдання може бути відкладений без впливу на інші завдання у графіку. Для керівника проєкту знання цієї інформації є дуже корисним. Правильне використання вільного часу може вирішити питання успіху чи невдачі в багатьох проєктах.

Уявімо ситуацію, коли керівник проєкту опиняється в середині проєкту і дізнається, що одне з завдань виконується довше, ніж було заплановано. Якщо це завдання є частиною критичного шляху, і воно не буде завершено вчасно, весь проєкт буде затриманий. З огляду на характер цієї діяльності, можливо, додавання додаткового ресурсу прискорило б виконання і дозволило команді завершити її вчасно. Однак в проєкті немає вільних ресурсів, оскільки всі вони працюють над іншими завданнями.

Але що, якщо один з ресурсів, які потрібні для цієї діяльності, працює над завданням з вільним часом? Тоді його завдання може бути відкладено, і його ресурси можна використовувати для завдання, яке відстає. Здатність виявити ресурси з вільним часом може бути вирішальною в цій ситуації.

Коли завдання, яке відстає за графіком, повертається на потрібну траєкторію, керівник проєкту може перемістити спільний ресурс назад на його початкове завдання. Без знання про вільний час (slack), ніхто не міг би точно визначити, які ресурси можна перемістити, коли це можна зробити і на який термін, не впливаючи на термін закінчення проєкту.

Знання про вільний час графіку надає керівнику проєкту необхідну інформацію для ефективного управління ресурсами. Вони можуть визначити, які ресурси мають вільний час, коли цей час доступний і на який період його можна використовувати, не впливаючи на кінцевий термін проєкту. Це допомагає оптимізувати розподіл ресурсів для досягнення максимальної продуктивності та забезпечує вчасне завершення проєкту.

Метод критичного шляху використовується для оцінки мінімальної тривалості проєкту та визначення кількості гнучкості графіку на логічних мережевих шляхах у межах моделі графіку. Цей метод аналізу мережевого графіку розраховує ранні дати початку, ранні дати закінчення, пізні дати початку та пізні дати закінчення для всіх дій, ігноруючи будь-які обмеження ресурсів, виконуючи аналіз наперед і назад через мережу графіку. Критичний шлях – це сукупність дій (робіт), яка представляє найдовший шлях через проєкт і визначає найкоротшу можливу тривалість проєкту. Отримані ранні та пізні дати початку та закінчення не обов'язково є графіком проєкту; вони вказують на періоди часу, протягом яких може бути виконана дія, використовуючи параметри, введені в модель графіку для тривалості дій, логічних зв'язків, передчасних та запізнених зв'язків та інших відомих обмежень. Метод критичного шляху використовується для розрахунку критичних шляхів та кількості загальної і вільної гнучкості або гнучкості графіку на логічних мережевих шляхах у межах моделі графіку [8].

Метод критичного шляху є достатньо популярним та широко використовуваним механізмом для управління проєктами, він допомагає зрозуміти кількість ресурсів, що необхідні для задоволення потреб проєкту, дозволяє визначити можливі затримки для планування, є зручним у візуалізації отриманих результатів та допомагає з легкістю пріоритезувати завдання між собою. Саме тому він був обраний у якості досліджуваного методу.

### 2.3 Метод діаграм Ганта

Діаграми були названі у честь Генрі Ганта, тому що вважається, що йому прийшла ідея планування проєктів за допомогою горизонтальних гістограм. А першим, хто втілював це у життя, був Кароль Адамецький. Більшість його проєктів були пов'язані зі сталеварною діяльністю.

Діаграми Ганта – це інструмент управління проєктами, який ілюструє, як запланована робота виконується впродовж часу. Зазвичай вони складаються з двох частин: у лівій частині перераховано список завдань, а в правій – часова шкала з полосами, що показують виконану роботу. Графік Ганта також може включати дати початку і завершення завдань, контрольні точки, залежності між завданнями і виконавців.

Наразі вони часто використовуються як інструменти для створення планів і стратегій розробки. Щоб відповідати вимогам сучасного програмного забезпечення, інструменти для розробки планів і стратегій, такі як Jira Software, включають в себе розгортання списків завдань, ресурсні панелі управління та інші можливості. Це допомагає командам дотримуватися єдиної стратегії проєкту, незважаючи на ітеративний характер розробки програмного забезпечення. Jira містить два інструменти для розробки планів і стратегій, які дозволяють створювати діаграми Ганта для проєктів: дорожні карти, які дозволяють створювати плани на основі завдань Jira, призначених для команди, та інструмент Advanced Roadmaps, який надає ті ж функції, але на рівні команд та організацій [9].

Існує дуже багато готових програмних рішень, що допомагають побудувати такі діаграми, серед них VisionFlow, ProjectInsight, Celoxis, також відома організація Atlassian дозволяє створювати такі діаграми на базі задач кожного розробника проєкту. Ну і також слід зазначити, що однією з популярних програм для побудови таких діаграм є і Microsoft Excel. Приклад діаграми Ганта зображено на рисунку 2.1.

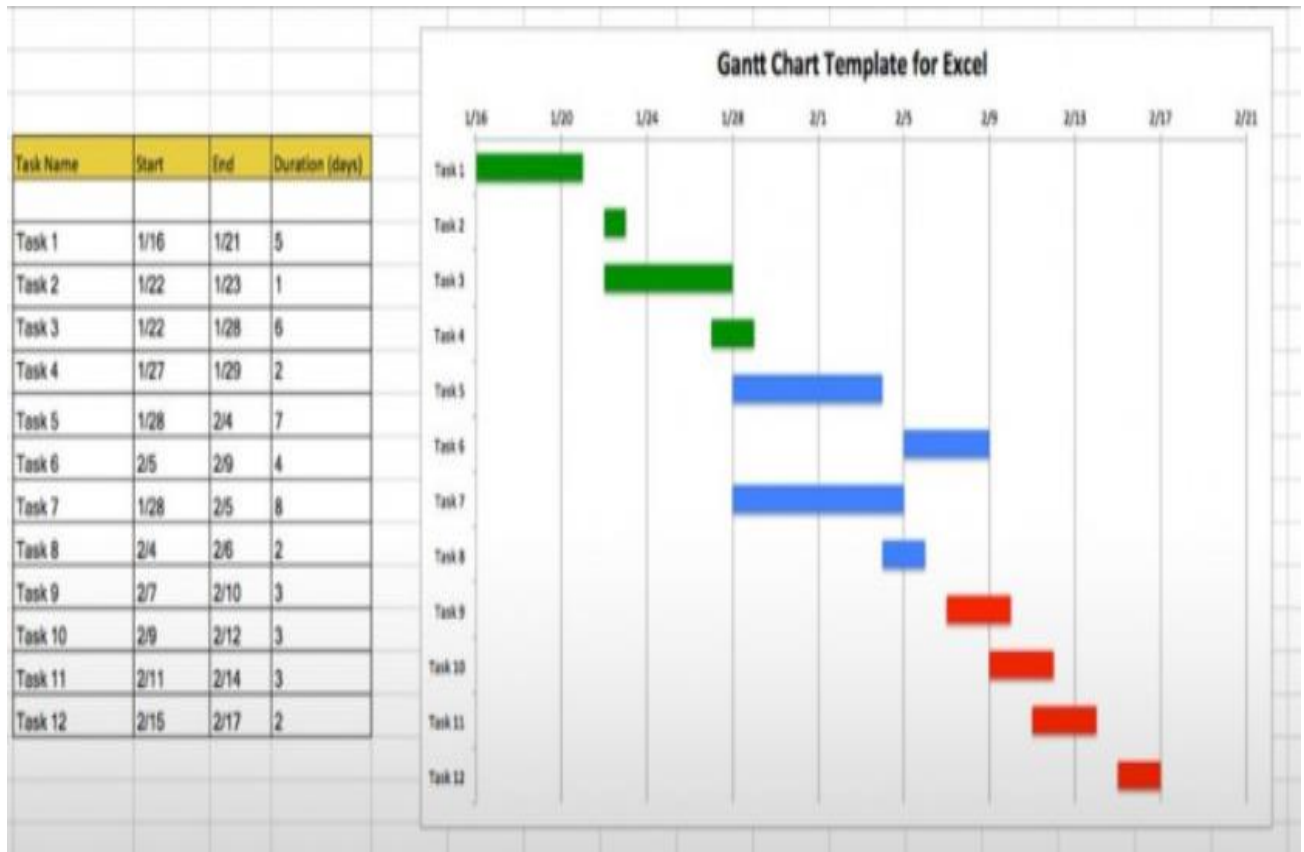


Рисунок 2.1 – Діаграма Ганта у Microsoft Excel

Переваги таких діаграм у тому, що вони дозволяють легко візуалізувати інформацію про проєкт та його роботи, дають можливість поглянути на проєкт цілком на одній діаграмі, допомагають у побудові Roadmap (плану дій проєкту), і звичайно ж, допомагають керувати ресурсами. Недоліками є необхідність постійного оновлення даних для створення діаграми, можливість неправильного відображення повної картини при неправильному використанні, а налаштування діаграми може зайняти багато часу.

## 2.4 Метод PERT (Project Evaluation and Review Technique)

Цю методику використовують різні організації вже понад 50 років. Її було розроблено в 1958 році Відділом із реалізації спеціальних проєктів ВМС США. Методика PERT являє собою систему керування, яка допомагає планувати великі та складні проєкти. У межах цієї методики здійснюється глибокий аналіз проєкту до дати початку його реалізації, починаючи із загального оцінювання. Ця методика використовувалася для розробки ракет Polaris і навіть під час проведення зимових Олімпійських ігор 1968 року у Греноблі.

Project Evaluation and Review Technique (PERT) – метод оцінки та аналізу проєктів, який використовується в управлінні проєктами. PERT призначений для дуже масштабних, одноразових та складних проєктів. Метод враховує наявність невизначеності, даючи можливість розробити робочий графік проєкту без точного знання деталей і необхідного часу для всіх його складових. Найбільш популярною частиною PERT є саме метод критичного шляху.

Для планування проєктів потрібно розуміти обсяг робіт з оцінкою того, скільки кожна задача може зайняти часу. Оскільки частіше за всього виконавці переоцінюють свої можливості або навпаки, намагаються закласти ризики та свої часові резерви. Тобто, просто кажучи, не можуть об'єктивно оцінити реальні затрати часу на виконання задач. Для того, щоб отримати і розрахувати найбільш коректну оцінку і в той же час врахувати можливі ризики у PERT методі застосовується оцінка за оптимістичною, песимістичною, очікуваною і найбільш вірогідною складовою. Оптимістичний час – це мінімальний час, необхідний для виконання роботи за умови, що все працюватиме ідеально. Песимістичний – це навпаки максимальний час за умови, коли виникають проблеми та негаразди. Найбільш вірогідний термін – це прогнозований термін виконання, коли все йде приблизно як завжди, а очікуваний – це розрахований термін виконання з допущенням деяких проблем.

Далі розраховуються середні значення з урахуванням вагових коефіцієнтів за формулою:

$$Y = \frac{t_0 + 4t_m + t_p}{6},$$

де  $t_0$  – мінімально можлива тривалість виконання завдання;

$t_m$  – тривалість виконання завдання в припущенні, що все йде так, як відбувається найчастіше;

$t_p$  – тривалість виконання завдання при найгіршій продуктивності.

За допомогою розрахованих значень визначається вірогідність виконання проекту у визначений термін на основі функції нормального розподілу.

Як значення, для якого буде розраховуватись функція нормального розподілу, будуть прийматись наші очікування щодо терміну виконання робіт по проекту. Він може співпадати з сумою розрахованих показників чи відрізнятись. Математичне сподівання (середнє значення) для розрахунків буде

взято як сума  $\sum_{i=1}^n \frac{t_0 + 4t_m + t_p}{6}$  розрахованих за PERT методом оцінок.

За допомогою вищезгаданих розрахунків можна отримати приблизний, але досить показовий результат, за яким можна визначити вірогідність реалізації проекту в очікуваний термін з урахуванням ризиків, що можуть виникнути при реалізації задач та робіт даного проекту [10].

Однією з великих переваг PERT є його обширне планування. Розробка мережі та аналіз критичного шляху виявляють взаємозалежності та проблеми, які не є очевидними за іншими методами планування. Тому PERT визначає, де слід зробити найбільший зусилля, щоб зберегти проєкт за розкладом.

Складність PERT ускладнює проблеми впровадження. Для системи звітності, організованої за методом PERT, існує більше вимог до даних, ніж для більшості інших систем. Тому PERT стає дорогим у плані підтримки та найчастіше використовується великими та складними програмами, коли необхідно дослідити велику систему з значною кількістю зв'язків.

## 2.5 Метод планування Agile (Scrum та Kanban)

Напевне, одним із найрозповсюдженіших та популярних серед методів планування проєктів є Agile. Майже кожен новий проєкт у ІТ-технологіях слідує саме цій моделі планування, яка показала себе достатньо непогано для динамічних та розвиваючихся проєктів, у яких тенденція постійних змін має критичне значення.

Термін «Agile» має велику кількість значень. Це і ітеративний підхід до планування проєктів, що передбачає розбиття проєкту на окремі етапи, і спосіб мислення та філософії, якій слідують декілька різних підходів (наприклад, таких як Scrum та Kanban), або в оригіналі взагалі маніфест гнучкої розробки програмного забезпечення, тобто певний документ, що містить у собі цінності та принципи гнучкої розробки проєктів (в особливості проєктів по створенню програмного забезпечення).

Основні принципи базуються на задоволенні потреб замовника завдяки регулярному виконанню поставлених задач, створенню такого ПЗ, що буде оновлюватись з періодичністю від двох тижнів до двох місяців. Над проєктом повинні працювати тільки вмотивовані професіонали, що будуть якісно та вчасно виконувати свої обов'язки.

Scrum є однією з методик керування проєктами та підвидом Agile. Запропонована Кеном Швабером та Джеффом Сазерлендом, ця методологія базується на тому, що створення ПЗ не є визначеним процесом, а цей процес є емпіричним із складними перетвореннями вхідних даних у вихідні дані, які можуть повторюватися чи не повторюватися у різних обставинах.

Основною ідеєю є те, що розробка включає в себе кілька екологічних і технічних факторів, які ймовірно змінюються протягом процесу (наприклад, вимоги, терміни, ресурси, технологія). Це робить процес розробки непередбачуваним і складним, вимагаючи гнучкості процесу розробки системи для відповіді на ці зміни. В результаті процесу розробки розробляється корисна система [11].

Перевагами Scrum є можливість кращого керування проєктом, розвиток проєкту є прозорим та видимим для кожного з членів команди, покращується комунікація, проєкт може розвиватися навіть в умовах нечітких вимог. Також слід зазначити те, що з плином часу процеси такого підходу можуть постійно вдосконалюватися завдяки ітеративності такого підходу. До недоліків можна віднести можливість некоректного розподілення робіт на одну ітерацію, особливо це може проявлятися на початку проєкту або при різкій зміні його розробників. Також такий підхід вимагає постійного моніторингу та часу для самовдосконалення та дискутування, яке розробники могли б витратити на реальні задачі, тобто вимагає ресурсів для того, щоб постійно покращувати цей процес.

Kanban є іншим підходом до планування та менеджменту проєкту, що полягає у візуалізації роботи проєкту за допомогою дошки (board) та завдань (tasks), які кожен розробник бере після завершення минулої задачі. Дошка має декілька критеріїв, до яких можна віднести завдання. Також це називають статусом завдання. До них відносять: «у процесі аналізу», «виконано», «на етапі перевірки» та багато інших у залежності від вимог. Завдяки такій моделі кожен учасник проєкту має краще поняття про етапи розвитку проєкту та окремого завдання, значно підвищується прозорість та групування усіх завдань.

Система була створена менеджерами автоконцерну Toyota у 1962 році.

Тайічі Оно, один із творців виробничої системи Toyota, говорив: «Два основних принципи системи виробництва Toyota – це «вчасно» і автоматизація з участю людини, або автономізація. Для управління системою використовується інструмент – це і є Kanban». Перша система для програмування, створена за допомогою Kanban була застосована у Microsoft у 2004 році.

Механізм роботи Kanban полягає у тому, що кожній роботі відповідає певна картка (одиниця якоїсь необхідної для проєкту сутності), новий елемент роботи може початись лише тоді, коли для нього ця картка є доступною. Ця доступна картка прикріплюється до робочого елемента під час його проходження через усю систему. Коли карток не залишається, нову роботу не



можна розпочинати. Будь-яка нова робота повинна залишатися в черзі, доки картка не звільниться. Коли виконана певна кількість роботи, картка звільняється і знову вводиться в обіг. Тепер можна розпочинати роботу над новим елементом у черзі. Цей механізм відомий як «витягуюча система», оскільки нова робота витягується системою, коли вона має достатню місткість для цього, а не нав'язується системі за вимогою. Витягуюча система не може бути перевантаженою, якщо місткість, визначена кількістю карточок у обігу, визначена правильно [12].

Задачею Kanban є баланс різних спеціалістів всередині команди задля забезпечення одночасної роботи кожного з розробників цього проєкту. Він потрібний для того, щоб усі в певний момент часу були зайняті різними видами діяльності для того, щоб не втрачався час та щоб працівники не сиділи без роботи.

Основними відмінностями у роботі між Kanban та Scrum є по-перше робочі цикли. Якщо у Scrum – це більш короткі ітерації з невеликим проміжком часу (два тижні), то ітерація за Kanban може тривати до двох місяців, після цього команда може підбити підсумки, аналізувати подальші плани та провести ретроспективу. Також слід зазначити й різний підхід до оцінювання робіт за складністю або часом. Якщо Scrum вимагає оцінювання робіт у story points (оцінка за складністю або часом) для визначення необхідної кількості задач на одну ітерацію, то для Kanban ці оцінки не потрібні і команда може сама вирішити, скільки часу потрібно на ту чи іншу активність, тобто у цьому випадку Kanban є більш гнучким.

До недоліків Scrum можна віднести більшу кількість нарад, які потрібно провести для того, щоб визначити продуктивність команди, проаналізувати кожну роботу та зробити аналіз помилок на кожній ітерації, на що йде достатньо багато часу, який розробники могли витратити на саме роботу. У Kanban же навпаки через те, що немає чіткого визначення часу, за скільки треба виконати ту чи іншу роботу, розробник може іноді викладатися не на повну силу та розтягувати задачі у часі.

Якщо ж у цілому вирішувати яка саме методологія підходить для проєкту, то тут треба визначитись з тим, наскільки жорсткими є умови розвитку такого проєкту. Якщо чітко поставлені терміни і команда знає кожен наступний крок, то тут підійде більше Kanban, де є чітка та логічна послідовність усіх задач на кожному етапі. Якщо ж проєкт постійно оновлюється та змінюється, то Scrum буде більш кращою альтернативою завдяки більшій кількості комунікацій.

## Висновки за розділом 2

Було розглянуто низку методів, що застосовуються для задач планування та управління проєктами, серед яких було виділено п'ять основних та найбільш застосованих на реальних проєктах – метод критичного шляху, метод діаграм Ганта, PERT-метод, методи Scrum та Kanban, які є одними з найбільш популярних для IT-проєктів сучасності.

У якості досліджуваного методу було взято метод критичного шляху через декілька причин. По-перше, він є достатньо зручним для визначення часу, що необхідно витратити на впровадження проєкту, по-друге дозволяє визначати можливі затримки під час створення проєкту. Також він є зручним для візуалізації отриманих результатів, що дозволяє легко пріоретизувати завдання.

Слід зазначити і те, що метод PERT (Project Evaluation and Review Technique) є дуже схожим до методу критичного шляху та використовує його принципи для планування часу розробки масштабних проєктів.

Дуже популярною зараз є методологія Agile, що включає у себе два підвиди – Scrum та Kanban. У першому підході основна ідея полягає у ітеративності процесу та постійне його вдосконалення від однієї ітерації до іншої, список робіт визначається зазвичай до старту наступної ітерації. Kanban же, навпаки, не має чітких прив'язок до часу та його ітерації є набагато більш довгими. Метод зосереджується на пріоретизації задач та виконанні їх чітко одна за одною без прив'язки до конкретного спеціаліста.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Вибір мови програмування

Зараз ми вступаємо в еру бурхливого розвитку штучного інтелекту, особливо, генеративного штучного інтелекту, який набуває неабиякого значення не тільки у сфері інформаційних технологій, але й загалом в усіх сферах життєдіяльності людини. Хтось бачить у цьому неабиякі можливості для подальшого прогресу, хтось бачить у цьому загрозу як для людства, так і для кожної людини окремо через те, що можливо у майбутньому штучний інтелект прийде на заміну багатьом нинішнім професіям.

Хай там як, але у подальшому штучний інтелект буде відігравати все більшу і більшу роль у житті людини незалежно від того, як швидко це станеться. Чималу роль у такому стрімкому розвитку відіграє одна з найпоширеніших та найпопулярніших мов програмування – Python. Вона має не тільки велике ком'юніті у світі, але й дуже різноманітний вибір бібліотек програмування, які здавалося б, можуть вирішити задачі будь-якого напрямку та складності.

Через величезну популярність цієї мови та простоту у використанні та різноманіття опцій для реалізації тієї чи іншої задачі було обрано саме цю мову програмування для підрахунку результатів проєкту та створення програмного продукту, який допоможе знайти будь-який критичний шляху у проєктах різної довжини та тривалості і розрахувати економічні параметри такого проєкту.

Мова була створена та випущена вперше у 1991 році співробітником нідерландського інституту CWI Гвідо ван Россумом. Ім'я було вибрано не випадково і походить від комедійного телешоу «Монті Пайтоновці», що вважається одним з революційних жанрів абсурдного гумору. Назва мови була вибрана через її хорошу запам'ятовуваність.

Основна мета Python – зробити програмування простішим, лаконічним та доступним для розробників. Саме через це мова отримала свій читабельний

синтаксис, що робить її привабливою для початківців та досвідчених програмістів.

Python став однією з найпопулярніших мов програмування в світі завдяки своєму широкому спектру застосувань. Він використовується в різних сферах, включаючи веб-розробку, наукові дослідження, обробку даних, штучний інтелект, машинне навчання та багато іншого. Ця універсальність робить його привабливим для розробників усіх рівнів і галузей.

Python також знайшов своє застосування у великих технологічних компаніях. Google, Facebook, Instagram, Dropbox та багато інших використовують Python у своїх продуктах та сервісах.

Python також має велику та активну спільноту розробників, яка постійно розширює екосистему мови. Саме завдяки активній спільноті Python постійно розвивається та підтримується на високому рівні.

Python є вільною та відкритою мовою програмування, що означає, що ви можете використовувати його безкоштовно, розповсюджувати його та змінювати, якщо потрібно. Ця відкритість сприяє подальшому розвитку мови та створенню нових бібліотек і фреймворків, які допомагають розробникам у їх проєктах.

Існує безліч бібліотек та фреймворків, які допомагають розробникам вирішувати конкретні завдання.

Наприклад, Django і Flask – це популярні фреймворки для веб-розробки, NumPy та pandas – для обробки та аналізу даних, TensorFlow та PyTorch – для машинного навчання.

В управлінні та прогнозуванні термінів проєктів Python використовується з різноманітним інструментів для аналізу даних, візуалізації інформації, автоматизації завдань та спрощення взаємодії в команді.

Графіки та діаграми Ганта дозволяють відобразити важливі роботи у проєкті. Python має багато бібліотек для створення графіків, серед яких Matplotlib і Plotly. Вони дозволяють створювати діаграми Ганта для відображення завдань та строків виконання.

Також завдяки системі контролю версій Git, що може бути з легкістю налаштована під Python можна автоматизувати управління кодом та впровадити механізми версіонування.

Завдяки цим перевагам Python стає надійним інструментом для розробки та прогнозування проєктів.

Загалом, Python став однією з найпопулярніших та важливих мов програмування у світі завдяки своєму читабельному синтаксису, широкому спектру застосувань, активній спільноті та постійному розвитку. Його простота та потужність роблять його важливим інструментом для програмістів у різних галузях.

### 3.2 Бібліотека NetworkX для роботи з графами

Щоб відобразити мережу взаємопов'язаних робіт необхідно побудувати граф, а задля пошуку найдовшого маршруту цього графу необхідно знайти його критичний шлях і на основі цього вже розраховувати економічні параметри проєкту.

Одним із найкращих інструментів для цього буде бібліотека NetworkX, яка має широкий функціонал для роботи з графами та мереживими структурами. Вона може створити вершини з будь-яких вхідних даних будь то текст, XML-дані чи зображення та здатна отримати характеристики для будь-якої вершини графа, його висоту, діаметр тощо.

До речі, створена тим самим Гвідо ван Россумом за допомоги Девіда Хагбера у 2003 році вона була вперше опублікована і з тих пір постійно розвивалася. Основною їхньою метою було створення простого програмного продукту для побудови та редагування графів і імплементація алгоритмів для їхнього аналізу. NetworkX продовжує розвиватися і покращуватися, і вона залишається одним із найпопулярніших інструментів для роботи з графами та мережами в мові програмування Python. Її історія свідчить про важливість

відкритого вихідного коду і спільної розробки в сфері програмного забезпечення.

Бібліотека може бути легко інтегрована у проєкт за допомогою команди `import networkx as nx` за умови, якщо даний пакет вже присутній на комп'ютері користувача, у протилежному випадку необхідно виконати команду `pip install networkx`.

Основними можливостями бібліотеки є не тільки створення графів і отримання їх характеристик, але їх візуалізація та перетворення у зручні формати для подальшого використання, передачі та завантаження.

Слід зазначити, що бібліотека дозволяє працювати з будь-яким типами графів, як орієнтованими, так і неорієнтованими, мультиграфами.

Перевагами даної бібліотеки є її висока продуктивність та швидкість. Вона може без будь-яких проблем побудувати велику систему до 10 мільйонів вершин та порядку 100 мільйонів ребер між ними, це досягається завдяки низькорівневій структурі мови. Також однією з великих переваг бібліотеки є її простота та гарна документація, що дозволить будь-кому розібратися у її тонкощах та не викличе серйозних труднощів у її опануванні. Але навіть у випадку питань, що можуть виникнути при розв'язанні задач за допомогою бібліотеки, існує велика вірогідність того, що відповідь на це питання вже була знайдена завдяки активній спільноті користувачів.

Але на жаль, у неї є і недоліки, одним із яких є невелика продуктивність при роботі з дуже великими графами, і тому вибір цієї бібліотеки буде не завжди найкращим рішенням, коли аналізується дуже велика мережа. Також великим мінусом є не досить гарна підтримка можливостей візуалізації отриманих результатів та графів, що змушує використовувати більш спеціалізовані бібліотеки для розв'язання цієї проблеми.

У цілому ж можна сказати, що ця бібліотека є гарним вибором для розв'язання спектру задач, що пов'язані з аналізом проєктів (якщо вони не є занадто великими та розгалуженими) та надає широкий функціонал для роботи з графами.

### 3.3 Бібліотека GraphViz як засіб візуалізації даних

Бібліотека Graphviz була створена в рамках проєкту Graphviz, який розпочався в AT&T Labs у середині 1990-х років. Основною метою проєкту було створення інструменту для автоматизованої візуалізації графів та мереж. Graphviz був розроблений для вирішення завдань візуалізації даних у великих та складних програмах та системах.

Graphviz – це вільне програмне забезпечення для візуалізації графіків з відкритим вихідним кодом. Пакет Graphviz, який працює під Python 3.7+ у Python, надає чистий інтерфейс на Python для цього програмного забезпечення. Цей пакет дозволяє створювати як неорієнтовані, так і орієнтовані графи.

Побудова об'єкта графа чи орієнтованого графа за допомогою Graphviz схожа на те, як це робиться за допомогою NetworkX, в тому сенсі, що потрібно просто визначити вузли та ребра об'єкта графа і надати їм атрибути відповідно. Однак є також ключові відмінності. Наприклад, в пакеті NetworkX вузол може бути визначений як будь-який об'єкт, який можна хешувати (крім None). Але в Graphviz вузол може бути визначений лише як рядок.

Форма за замовчуванням вузла у графі, створеному за допомогою Graphviz, – це еліпс. За допомогою Graphviz можна отримати різноманітні форми для вузлів. Ці форми можуть бути основаними на полігонах, записах або визначеними користувачем. Крім того, можливо також отримати різні форми для різних вузлів [13].

У даній роботі можливості цієї бібліотеки були використані для створення орієнтованого графу з можливістю його більш коректної візуалізації із виділенням найдовшого маршруту у графі та відповідних вершин на ньому, а також збереження малюнків цих маршрутів на кожному ітераційному етапі підрахунку розглядуваних економічних параметрів.

До переваг можна також віднести простоту використання, можливість побудови будь-яких типів графів та видів їхніх вузлів, що вказує на високу

гнучкість даного засобу при візуалізації. Також великою перевагою є можливість збереження вихідних файлів бібліотеки у різні формати даних серед яких PNG, SVG, PDF та ін. Не можна не відмітити широку популярність бібліотеки, можливість поспілкуватися та задати будь-яке питання до активної спільноти користувачів та відкритий вихідний код. Усі ці можливості дають змогу швидко розвиватися цій бібліотеці та постійно модернізуватися.

Недоліками ж є також складність розміщення великих графів, що робить нелегким їх переглядання у межах одного екрану, великі графи також створюють ще одну проблему, пов'язану зі споживанням великої кількості ресурсів на комп'ютері дослідника. На жаль, обмежені анімаційні властивості бібліотеки. Graphviz не надає вбудованої підтримки для створення анімацій графів. Це означає, що створення анімованих візуалізацій може бути складним завданням.

Але, не дивлячись на це, Graphviz залишається важливим інструментом для візуалізації графів та мереж та продовжує впливати на різні галузі, де треба аналізувати та візуалізувати складні структури даних.

### 3.4 Gephi

Застосунок Gephi для візуалізації графів з великою кількістю ребер та вершин є дуже потужним засобом для візуалізації. Він був створений французьким розробником Мат'є Баст'єм під час його докторських досліджень у Швейцарії за допомоги Себастьяна Шана та Матіаса дель Фільго. Gephi задумувався як зручний засіб для візуалізації та аналізу складних графів, що має відкритий вихідний код, що б сприяло подальшому розвитку Gephi завдяки можливості дописувати до нього нові додатки та ділитися новими ідеями щодо його розвитку. Основні галузі, що використовують цей застосунок – це біологія, інформаційні технології, аналітичні компанії, що досліджують взаємозв'язки між певними подіями чи людьми та ін.



Зазвичай у біології його використовують для візуалізації біологічних мереж (наприклад білкових чи генетичних), у інформаційних технологіях він буде потужним інструментом для аналізу великих обсягів даних, коли потрібно знайти найсильніші зв'язки між окремими характеристиками, для виявлення складних патернів у даних.

Для даної атестаційної роботи Gephi має неабияке значення, через те, що він дозволяє не тільки відобразити саму структуру проєкту через відображення пов'язаних між собою робіт, але й тим, що він дає змогу позначити критичний шлях проєкту на кожній ітерації розрахунків, а також за допомогою вбудованих механізмів можна задати характеристики для кожного ребра (тобто його ідентифікаційний номер, колір, вагу цього ребра для графу) і для кожної вершини. У даному випадку можливості GraphViz є більш обмеженими, тому Gephi завдяки своїй гнучкості має суттєву перевагу.

Також до його переваг можна віднести дуже зрозумілий та інтуїтивний інтерфейс, за допомогою якого без великих зусиль можна кастомізувати граф, додати чи видалити вершини та ребра, створити новий чи імпортувати вже існуючий граф. Як вже зазначалося вище, відкритий вихідний код сприяє розширенню спільноти користувачів цього застосунку.

До недоліків можна віднести обмежену підтримку різних форматів даних для Gephi, тому потрібно визначитись із необхідним форматом заздалегідь, також він не дуже підходить для мереж, що змінюються в реальному часі і більше налаштований на роботу зі статичними даними. Ну і, на жаль, дуже розповсюдженою проблемою є і те, що необхідність працювати з дуже великою кількістю даних вимагає також і значної кількості ресурсів, що потрібно задіяти для роботи з Gephi.

### 3.5 Опис програми

Програма була створена за допомогою компілятора PyCharm Community Edition, що створений компанією JetBrains для написання програм на Python.

У першій частині програми імпортуються основні бібліотеки такі як `graphviz`, `networkx`, `tabulate`, `numpy` та інші. Далі йде визначення основного класу `Edge`, що буде потрібен нам для роботи з ребрами графу, а також основні функції: `calculate_weight_and_assign` – для підрахунку ітераційного кроку та вирахування коректного значення ваги ребра, а також `printGraph` – ця функція може бути застосована задля візуалізації графу при розробці програми у `Anaconda Navigator`.

У головному циклі програми визначені усі роботи проєкту за допомогою конструктора класу `Edge`. Першим параметром слугує номер вихідної вершини, другим – вхідної, далі параметр початкової ваги ребра, мінімальної та максимальної. У додатку А з вихідним кодом програми частина робіт буде закоментована (це визначення ребер графу для першого проєкту). Це зроблено для того, щоб уникнути дуплікації імен змінних та помістити два види проєктів у одній програмі.

Після цього кожному ребру також надається свій колір для того, щоб явно вказати критичний шлях на графі. Ребра, що будуть належати цьому шляху будуть мати червоний колір, усі інші – чорний.

У кінці програми створюється таблиця з усіма результатами та її заголовок, а також явно задається масив з кроками ітераційного процесу від 0 до 1 та усі дані щодо ітераційного процесу виводяться у консоль.

Після запуску програми вона генерує крім таблиці з результатами у консолі також ще 11 файлів з розширенням `.graphml`. Ці файли надалі можуть імпортуватися у додаток `Gerhi` задля візуалізації графів на кожній ітерації.

### Висновки за розділом 3

Однією з найпопулярніших мов програмування для розв'язання математичних задач наразі, безсумнівно, є `Python`. У цій мові присутня велика кількість бібліотек задля роботи з графами (`NetworkX`), з великими

багатовимірними масивами й матрицями (NumPy), з таблицями (tabulate) та побудові моделей для машинного навчання (sklearn).

Саме тому ця мова була обрана для досліджуваної задачі. Методи бібліотеки NetworkX дозволяють не тільки побудувати сам граф будь-якого типу і розмірності, але і знайти критичний шлях, що є однією з головних задач при знаходженні тривалості усього проекту.

Не менш важливим є і візуалізація отриманих результатів, бо саме за допомогою неї можна зрозуміти якість результатів, а також з легкістю порівнювати їх між собою, що дасть змогу краще розібратися, чи треба вносити правки до процесу або знайти кореляційний зв'язок між певними характеристиками та, якщо потрібно, вдосконалити процес.

Задля цього було проведено аналіз як бібліотеки GraphViz, що є простою у використанні та може будувати будь-які типи графів, так і одного з найпотужніших застосунків для візуалізації – Gephi. Він дозволяє дуже гнучко налаштовувати параметри графу такі як вага ребра, його колір та задавати власні характеристики, що є вкрай важливим для коректної презентації найважливіших аспектів дослідження.

## 4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

### 4.1 Обчислення економічних характеристик мережевого графіку для проєкту створення Backend-частини сайту електронної комерції

Основною задачею дослідження є пошук основних економічних параметрів проєкту по створенню Backend-частини сайту електронної комерції (припускається, що сайт буде створено за допомогою однієї з платформ електронної комерції – SAP Commerce/Magento), тож, по-перше, необхідно визначитися з тим, які роботи необхідно виконати для того, щоб досягнути мети, а потім завдяки послідовності робіт побудувати граф.

Весь список робіт графу представлено у таблиці 4.1.

Таблиця 4.1 – Дані про кількість робіт та логічні зв'язки між ними

№	Назва роботи	Попередні роботи
1	Підбір команди розробників та аналітиків	–
2	Створення плану робіт та аналіз бізнес-вимог	–
3	Вибір інструментів розробки та основних фреймворків	1
4	Розгортання програмного середовища для подальшої розробки та тестування	3
5	Побудова архітектури	1, 2
6	Створення DB-моделі даних та аналіз зв'язків між таблицями	5
7	Налаштування серверу для швидкого пошуку продуктів, індексація даних	5
8	Створення каталогу продуктів, призначення цін, категоризація	6

Кінець таблиці 4.1

№	Назва роботи	Попередні роботи
9	Створення головної сторінки сайту	6
10	Створення сторінки для відображення деталей продукту	8
11	Створення сторінки для відображення категорій продуктів, налаштування пагінації	7, 8
12	Створення медіаданих та налаштування CDN	8
13	Створення акаунту користувача	9, 10
14	Налаштування Security, авторизація та аутентифікація, створення сесій	13
15	Налаштування інтернаціоналізації/локалізації	9, 10
16	Створення кошика продуктів	14
17	Створення механізму для розрахунку остаточних цін та застосування можливих акцій та дисконтів	14
18	Підключення Payment-системи	17
19	SEO-налаштування	15
20	Створення admin-сторінки	12, 16, 19
21	Створення Unit та інтеграційних тестів	11, 18, 20
22	Розгортання кінцевого продукту на Production-середовищі	21

Однією з найважливіших характеристик проекту є час його виконання, але заздалегідь неможливо точно спрогнозувати терміни його виконання, тож задля планування застосовуються методи нечіткої логіки.

Можна використовувати різні підходи до того, як саме інтерпретувати результат, наприклад оцінити можливість виконання проекту з чітко визначеним терміном його закінчення (наприклад, через рік) або оцінити час, що необхідний для завершення роботи над проектом з певною мірою

впевненості. Саме цей критерій є одним із найбільш розповсюджуваних і використовується для прийняття рішення про початок роботи над проектом або відмови від його реалізації.

За допомогою даних з таблиці 4.1 створимо мережевий графік даного проекту.

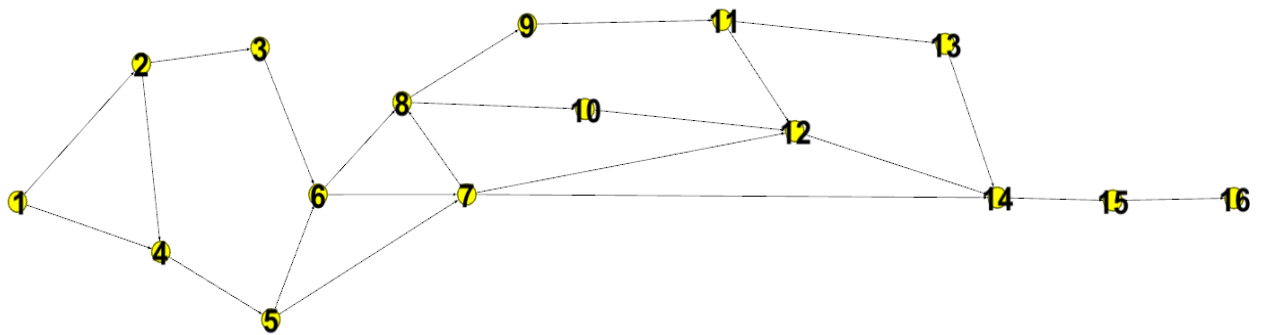


Рисунок 4.1 – Мережевий графік проекту

Також необхідно задати тривалість робіт, тобто визначити нечіткі числа для кожної роботи  $d_{ij} = \langle a_{ij}, b_{ij}, c_{ij} \rangle$ , де  $a_{ij}$  – мінімальний час виконання роботи,  $b_{ij}$  – час, коли робота точно буде виконана,  $b_{ij} = c_{ij}$ . Функція приналежності  $\mu_{ij}$  у цього нечіткого числа буде показувати впевненість у тому, що робота буде дійсно виконана у період між початковою та кінцевою подіями цієї роботи. Дані з тривалістю робіт були зведені до таблиці 4.2.

Таблиця 4.2 – Дані тривалостей робіт проекту

№	Назва роботи	Номери вузлів роботи $(i, j)$	Тривалість	
			Час $t_{\min}$	Час $t_{\max}$
1	Підбір команди розробників та аналітиків	(1,2)	30	60

Продовження таблиці 4.2

№	Назва роботи	Номери вузлів роботи ( <i>i, j</i> )	Тривалість	
			Час $t_{\min}$	Час $t_{\max}$
2	Створення плану робіт та аналіз бізнес-вимог	(1,4)	14	30
3	Вибір інструментів розробки та основних фреймворків	(2,3)	3	7
4	Розгортання програмного середовища для подальшої розробки та тестування	(3,6)	7	14
5	Побудова архітектури	(4,5)	14	21
6	Створення DB-моделі даних та аналіз зв'язків між таблицями	(5,6)	5	10
7	Налаштування серверу для швидкого пошуку продуктів, індексація даних	(5,7)	7	15
8	Створення каталогу продуктів, призначення цін, категоризація	(6,7)	14	21
9	Створення головної сторінки сайту	(6,8)	15	30
10	Створення сторінки для відображення деталей продукту	(7,8)	10	20
11	Створення сторінки для відображення категорій продуктів	(7,14)	10	20
12	Створення медіаданих та налаштування CDN	(7,12)	20	30

Кінець таблиці 4.2

№	Назва роботи	Номери вузлів роботи ( <i>i, j</i> )	Тривалість	
			Час $t_{\min}$	Час $t_{\max}$
13	Створення акаунту користувача	(8,9)	7	14
14	Налаштування Security, авторизація та аутентифікація, створення сесій	(9,11)	21	30
15	Налаштування інтернаціоналізації/локалізації	(8,10)	14	21
16	Створення кошика продуктів	(11,12)	7	10
17	Створення механізму для розрахунку остаточних цін та застосування можливих акцій та дисконтів	(11,13)	21	30
18	Підключення Payment-системи	(13,14)	30	45
19	SEO-налаштування	(10,12)	7	21
20	Створення admin-сторінки	(12,14)	30	60
21	Створення Unit та інтеграційних тестів	(14,15)	30	60
22	Розгортання кінцевого продукту на Production-середовищі	(15,16)	7	10
Усього			323	579

Розв'язком задачі буде перетин множини допустимих планів та множини нечіткої цілі. Допустимим планом буде мережевий графік, що побудований з певним коефіцієнтом впевненості. Мінімальне значення такого коефіцієнту впевненості (функції приналежності допустимих планів) буде рівним нулю і відповідатиме мережевому графіку з мінімальними можливими тривалостями



робіт, тобто це найкращий, але найменш вірогідний сценарій. Максимальне ж значення функції приналежності буде дорівнювати 1, і відповідатиме графіку з максимальними тривалостями робіт, тобто коли ми точно будемо впевнені, що роботи будуть виконані. Основною задачею буде пошук такого значення функції приналежності в діапазоні від 0 до 1, що воно б відповідало максимальному значенню перетину функції приналежності допустимих планів та нечіткої цілі.

У якості функції приналежності нечіткої цілі візьмемо функцію  $\mu_{\hat{c}}(x) = \frac{T^{\max} - t_k(x)}{T^{\max} - T^{\min}}$  ( $k$  – номер ітерації), яка показуватиме наближення до найкращого з можливих планів, тобто найкоротшого, але і найменш вірогідного.

Так як наші значення функції приналежності варіюються у діапазоні від 0 до 1, то будемо рухатись із кроком 0,1 і знаходити тривалість критичного шляху на кожному кроці та будемо знаходити значення нечіткої цілі.

Розглянемо яким чином буде відбуватись ітераційний процес завдяки методу критичного шляху.

Ітерація 1. Візьмемо  $\mu_{\hat{x}} = 0$ . Знайдемо найдовший шлях у графі з найменшими тривалостями робіт. Побудуємо мережевий графік та позначимо вагу кожного ребра відповідно до даних з таблиці 4.2. За допомогою функції `dag_longest_path` бібліотеки `networkx` знайдемо критичний шлях. Він проходить через вершини [1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16]. Відповідно до даних з таблиці 4.2 можемо зрозуміти, які саме роботи будуть виконані. Наприклад ребру (1, 2) відповідає робота з номером 1, її тривалість на даному шляху буде дорівнювати 30, ребру (2, 4) відповідатиме фіктивна робота, основна мета якої показати, що вона використовується тільки для логічної зв'язки декількох робіт і не вимагає затрати ресурсів, тож її тривалість буде дорівнювати нулю. Далі ребру (4, 5) буде відповідати робота №5 з тривалістю 14, ребру (5, 6) відповідає робота №6 з тривалістю 5 і так далі. Нарешті ребру (15, 16) буде відповідати робота №22 з тривалістю 7 робочих днів. Таким чином, просумувавши

тривалості робіт на критичному шляху отримаємо число  $t_0(x) = 189$ . Слід зауважити, що воно буде дорівнювати значенню  $T^{\min}$ , тобто найкоротшому серед можливих шляхів. Тобто, у найкращому випадку тривалість проєкту буде становити 189 днів.

Тепер необхідно знайти значення функції приналежності нечіткої цілі. Для її підрахунку необхідно розрахувати значення  $T^{\max}$ , тобто плану з найдовшими тривалостями робіт. Функція `dag_longest_path` у цьому випадку дає такий самий результат щодо вершин критичного шляху – [1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16]. У цей раз робота, що відповідає ребру між вершинами 1 і 2 буде дорівнювати 60, між вершинами 2 і 4 все так же 0, між вершинами (4, 5) – 21, між (15, 16) – 10. Просумувавши тривалості усіх робіт отримаємо, що  $T^{\max} = 321$ . Підставивши всі значення у формулу, отримаємо

$$\mu_{\hat{c}}(x) = \frac{T^{\max} - t_1(x)}{T^{\max} - T^{\min}} = \frac{321 - 189}{321 - 189} = 1. \text{ Цей коефіцієнт дорівнює одиниці тому, що}$$

функція нечіткої цілі показує близькість до найкращого з можливих планів, тобто саме того який і розглядався на цій ітерації. Тепер необхідно знайти  $\mu_{\hat{D}}(x) = \min\{\mu_{\hat{x}}(x), \mu_{\hat{c}}(x)\} = \min\{0, 1\} = 0$ . Перейдемо до наступної ітерації.

На рисунку 4.2 зображено мережевий графік з вагами ребер, що відповідають тривалостям робіт для  $\mu_{\hat{x}} = 0$ .

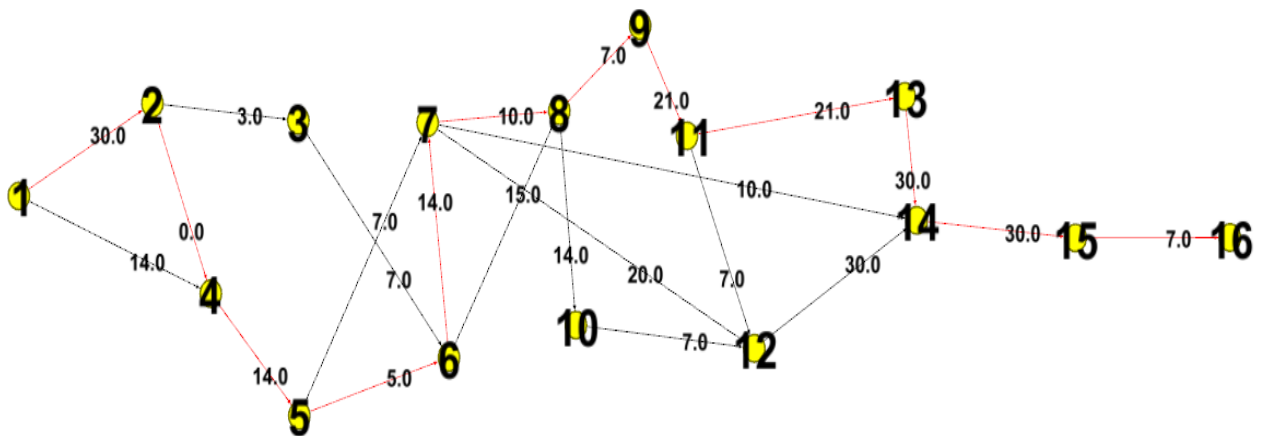


Рисунок 4.2 – Мережевий графік з тривалостями робіт при  $\mu_{\hat{x}} = 0$

Ітерація 2. Тепер необхідно взяти  $\mu_{\hat{x}} = 0,1$  та перерахувати ваги ребер (тривалості робіт) згідно цього коефіцієнту. У такому випадку тривалість роботи між вершинами 1 і 2 буде підраховуватись за формулою  $(T_i^{\max} - T_i^{\min}) \cdot \mu_{\hat{x}} + T_i^{\min} = (60 - 30) \cdot 0,1 + 30 = 33$ , ребру між вершинами 2 і 4 відповідатиме фіктивна робота, тож її тривалість буде дорівнювати нулю. Застосуємо цю ж формулу для ребра між вершинами 4 і 5, тоді  $(21 - 14) \cdot 0,1 + 14 = 14,7$ .

Підрахуємо таким чином тривалості кожної з робіт. Таким чином вийде, що роботі з номером 1 відповідатиме тривалість 33, роботі 2 – 15,6, роботі 3 – 3,4, роботі 4 – 7,7, роботі 5 – 14,7, роботі 6 – 5,5, роботі 7 – 7,8, роботі 8 – 14,7, роботі 9 – 16,5, роботі 10 – 11, роботі 11 – 11, роботі 12 – 21, роботі 13 – 7,7, роботі 14 – 21,9, роботі 15 – 14,7, роботі 16 – 7,3, роботі 17 – 21,9, роботі 18 – 31,5, роботі 19 – 8,4, роботі 20 – 33, роботі 21 – 33, роботі 22 – 7,3.

Далі за допомогою тієї ж функції `dag_longest_path` знайдемо найдовший шлях у графі – ним буде все той же шлях [1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16]. Номери робіт, що лежать на цьому шляху – 1, 5, 6, 8, 10, 13, 14, 17, 18, 21, 22.

Тепер необхідно знайти довжину критичного шляху, для цього просумуємо тривалості кожної з робіт на цьому шляху  $33 + 14,7 + 5,5 + 14,7 + 11 + 7,7 + 21,9 + 21,9 + 31,5 + 33 + 7,3 = 202,2$ .

Далі необхідно розрахувати коефіцієнт функції приналежності нечіткої цілі  $\mu_{\hat{c}}(x) = \frac{T^{\max} - t_2(x)}{T^{\max} - T^{\min}} = \frac{321 - 202,2}{321 - 189} = \frac{118,8}{132} = 0,9$ . І нарешті знайдемо  $\mu_{\hat{D}}(x) = \min\{\mu_{\hat{x}}(x), \mu_{\hat{c}}(x)\} = \min\{0,1; 0,9\} = 0,1$ .

Результати співпадають з результатами, які отримані програмно, отже можемо сказати, що всі розрахунки вірні. Мережевий графік для цієї ітерації зображено на рисунку 4.3.

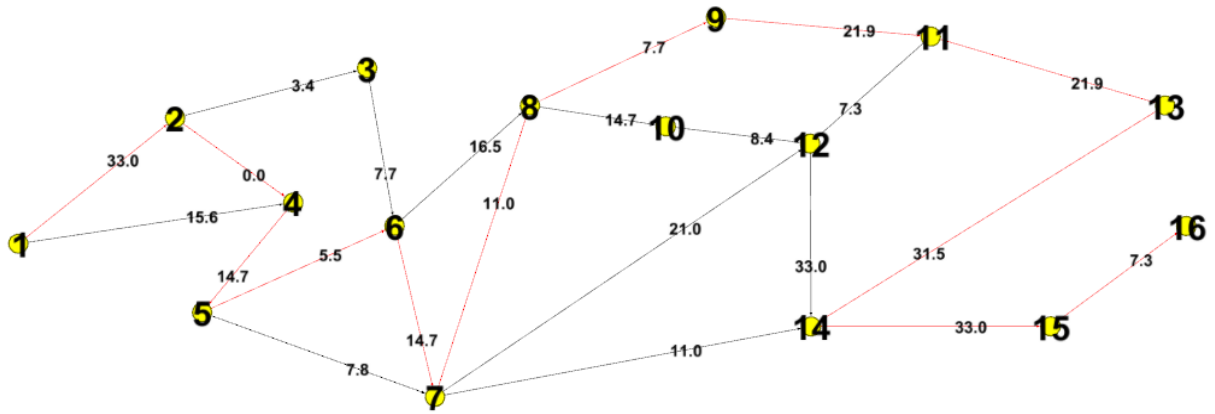


Рисунок 4.3 – Мережевий графік з тривалостями робіт при  $\mu_{\hat{x}} = 0,1$

Ітерація 3. Знову збільшуємо функцію приналежності  $\mu_{\hat{x}}$  допустимих планів на 0,1 і перераховуємо тривалості робіт. Для роботи №1 вона буде дорівнювати  $(60 - 30) \cdot 0,2 + 30 = 36$ , для роботи №2 –  $(30 - 14) \cdot 0,2 + 14 = 17,2$  і так далі. Після знаходження усіх тривалостей робіт знову відшукаємо критичний шлях графу, який і на цій ітерації залишиться тим самим – [1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16]. Тривалість критичного шляху: 215,4. Тоді

$$\mu_{\hat{c}}(x) = \frac{T^{\max} - t_3(x)}{T^{\max} - T^{\min}} = \frac{321 - 215,4}{321 - 189} = \frac{105,6}{132} = 0,8.$$

Далі  $\mu_D(x) = \min\{\mu_{\hat{x}}(x), \mu_{\hat{c}}(x)\} = \min\{0,2; 0,8\} = 0,2$ . Мережевий графік зображено на рисунку 4.4.

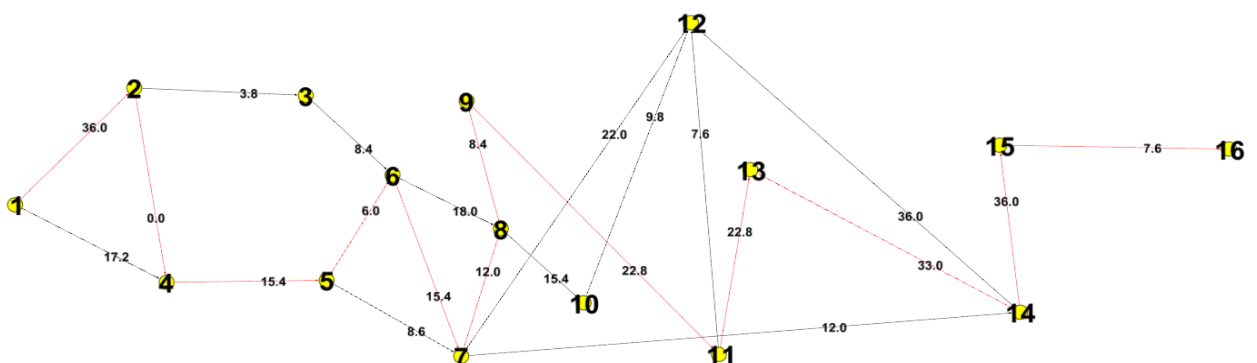


Рисунок 4.4 – Мережевий графік з тривалостями робіт при  $\mu_{\hat{x}} = 0,2$

Таким же чином продовжуємо ітераційний процес у діапазоні  $\mu_{\hat{x}}$  від 0 до 1 із кроком 0,1. Усього буде виконано 11 ітерацій. Мережеві графіки для 10 та 11 ітерацій наведені на рисунках 4.5 та 4.6 відповідно. Дані щодо коефіцієнтів та довжини критичного шляху на кожній ітерації зведені до таблиці 4.3.

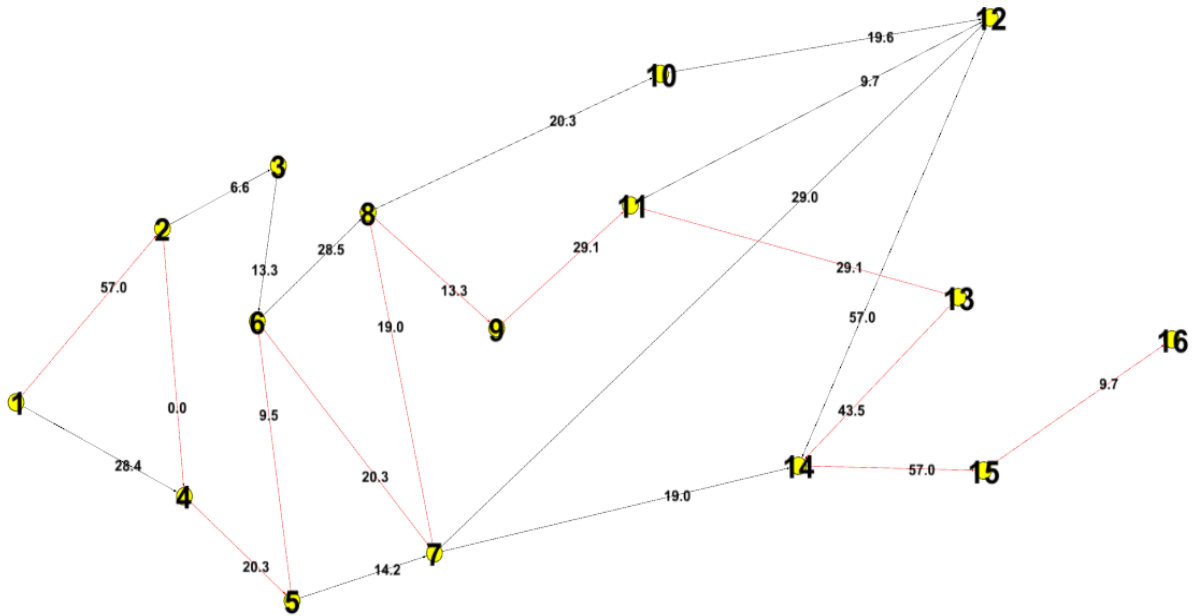


Рисунок 4.5 – Мережевий графік з тривалостями робіт при  $\mu_{\hat{x}} = 0,9$

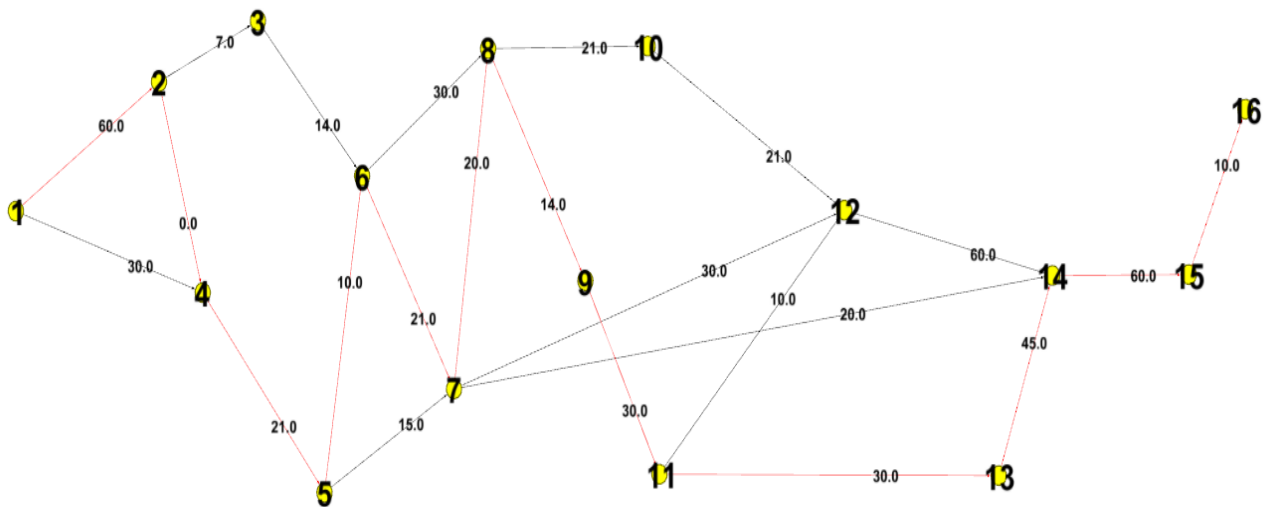


Рисунок 4.6 – Мережевий графік з тривалостями робіт при  $\mu_{\hat{x}} = 1$

Таблиця 4.3 – Результати ітераційного процесу

№	k	$\mu_{\hat{x}}$	Тривалість критичного шляху	$\mu_{\hat{c}}$	$\mu_{\hat{D}}$
1	0	0,0	189	1,0	0,0
2	1	0,1	202,2	0,9	0,1
3	2	0,2	215,4	0,8	0,2
4	3	0,3	228,6	0,7	0,3
5	4	0,4	241,8	0,6	0,4
6	5	0,5	255	0,5	0,5
7	6	0,6	268,2	0,4	0,4
8	7	0,7	281,4	0,3	0,3
9	8	0,8	294,6	0,2	0,2
10	9	0,9	307,8	0,1	0,1
11	10	1,0	321	0,0	0,0

Отже, згідно з наведеною таблицею результатів можна стверджувати, що максимум функції приналежності нечіткого розв'язку досягається на 6 ітерації при  $\mu_{\hat{x}} = 0,5$ .

А отже можна сказати, що найбільш оптимальним та вірогідним для початку проєкту буде план, при якому проєкт буде тривати 255 днів.

Можна по-різному інтерпретувати отримані результати в залежності від того, яка саме задача стоїть перед менеджером. Наприклад, якщо необхідно визначити вірогідність (або максимальне значення надійності), при якій план проведення робіт проєкту буде завершено за час, що не є більшим за 230 діб, то згідно таблиці така надійність буде дорівнювати близько 0,3 (або 30 відсотків). Або ж навпаки, якщо необхідно визначити скільки діб необхідно витратити, щоб досягти надійності 0,8 (або 80 відсотків), то з таблиці бачимо, що це значення буде дорівнювати 295 діб.

#### 4.2 Обчислення економічних характеристик мережевого графіку для модифікованого проєкту створення Backend-частини сайту електронної комерції

Задля порівняння результатів спробуємо модифікувати наш минулий проєкт та перебудуємо граф трішки іншим чином, враховуючи, що більшість робіт можуть бути виконані паралельно. Це значно збільшує кількість фіктивних робіт (для подальшого зв'язку робіт між собою), тож змінює саму структуру графу. Також були додані декілька робіт для більшої деталізації задач, що виконуються протягом усього проєкту.

Список усіх робіт було додано до таблиці 4.4.

Таблиця 4.4 – Дані про кількість робіт та логічні зв'язки між ними

№	Назва роботи	Попередні роботи
1	Підбір команди розробників та аналітиків	–
2	Створення плану робіт та аналіз бізнес-вимог	–
3	Вибір інструментів розробки та основних фреймворків	–
4	Розгортання програмного середовища для подальшої розробки та тестування	1
5	Побудова архітектури	1
6	Створення DB-моделі даних та аналіз зв'язків між таблицями	2, 5
7	Налаштування серверу для швидкого пошуку продуктів	1
8	Створення каталогу продуктів, призначення цін, категоризація	6
9	Створення головної сторінки сайту	3, 4, 5, 6

Кінець таблиці 4.4

№	Назва роботи	Попередні роботи
10	Створення сторінки для відображення деталей продукту	3, 4, 5, 6
11	Створення сторінки для відображення категорій продуктів, налаштування пагінації	3, 4, 5, 7, 8
12	Створення медіаданих та налаштування CDN	8
13	Створення акаунту користувача	3, 4, 5, 6
14	Налаштування Security, авторизація та аутентифікація, створення сесій	13
15	Створення кошика продуктів	8, 10
16	Налаштування інтернаціоналізації/локалізації	9, 10, 11, 13
17	Створення механізму для розрахунку остаточних цін та застосування можливих акцій та дисконтів	15
18	Підключення Payment-системи	15
19	SEO-налаштування	16
20	Створення admin-сторінки	14, 17, 18
21	Створення Unit та інтеграційних тестів	12, 18, 19, 20
22	Створення документації	12, 18, 19, 20
23	Виявлення та усунення потенційних ризиків	12, 18, 19, 20
24	Розгортання кінцевого продукту на Production-середовищі	21, 22, 23
25	Оптимізація продуктивності Production-середовища	24

У порівнянні з таблицею для минулого проєкту можна відразу відмітити, що зросла кількість робіт (вершин), до графу додалися задача по оптимізації продуктивності Production-середовища, створення документації і виявлення та усунення потенційних ризиків. Ці задачі дозволять удосконалити проєкт та зменшити ризики, що можуть виникнути при розгортанні та початковому



використанні Production-середовища.

Також значно зросла кількість попередніх робіт, що виконуються перед певними роботами.

Це було зроблено для того, щоб максимально деталізувати та правильно описати процес створення сайту електронної комерції. Слід зазначити також, що кількість фіктивних робіт зросла з 1 до 14 у порівнянні з минулим проєктом. Тривалості робіт для цього графу були наведені у таблиці 4.5.

Таблиця 4.5 – Дані тривалостей робіт проєкту

№	Назва роботи	Номери вузлів роботи $(i, j)$	Тривалість	
			Час $t_{\min}$	Час $t_{\max}$
1	Підбір команди розробників та аналітиків	(1, 4)	30	60
2	Створення плану робіт та аналіз бізнес-вимог	(1, 2)	22	30
3	Вибір інструментів розробки та основних фреймворків	(1, 3)	3	7
4	Розгортання програмного середовища для подальшої розробки та тестування	(4, 10)	7	14
5	Побудова архітектури	(4, 5)	14	21
6	Створення DB-моделі даних та аналіз зв'язків між таблицями	(2, 6)	5	10
7	Налаштування серверу для швидкого пошуку продуктів	(4, 7)	7	15
8	Створення каталогу продуктів, призначення цін, категоризація	(6, 8)	14	21

Продовження таблиці 4.5

№	Назва роботи	Номери вузлів роботи ( $i, j$ )	Тривалість	
			Час $t_{\min}$	Час $t_{\max}$
9	Створення головної сторінки сайту	(6, 9)	15	30
10	Створення сторінки для відображення деталей продукту	(6, 11)	15	20
11	Створення сторінки для відображення категорій продуктів, налаштування пагінації	(8, 12)	15	20
12	Створення медіаданих та налаштування CDN	(8, 13)	15	30
13	Створення акаунту користувача	(6, 14)	7	14
14	Налаштування Security, авторизація та аутентифікація, створення сесій	(14, 15)	21	30
15	Створення кошика продуктів	(8, 16)	8	10
16	Налаштування інтернаціоналізації/локалізації	(12, 17)	14	21
17	Створення механізму для розрахунку остаточних цін та застосування можливих акцій та дисконтів	(16, 18)	8	12
18	Підключення Payment-системи	(16, 19)	15	45
19	SEO-налаштування	(17, 20)	7	30
20	Створення admin-сторінки	(18, 21)	40	60

Кінець таблиці 4.5

№	Назва роботи	Номери вузлів роботи ( $i, j$ )	Тривалість	
			Час $t_{\min}$	Час $t_{\max}$
21	Створення Unit та інтеграційних тестів	(20, 22)	45	60
22	Створення документації	(20, 23)	45	50
23	Виявлення та усунення потенційних ризиків	(20, 24)	25	40
24	Розгортання кінцевого продукту на Production-середовищі	(24, 25)	7	10
25	Оптимізація продуктивності Production-середовища	(25, 26)	14	18
Усього			375	678

Тепер, коли є всі дані про тривалість робіт, про зв'язки між цими роботами, можна буде побудувати граф та розпочати ітераційний процес. Усі параметри ітераційного процесу буде збережено і він буде відбуватися аналогічно минулому пошуку.

Спочатку буде знайдено найменший та найдовший критичні шляхи у графі при ступені надійності 0 та 1, а далі з кожним підвищенням значення ступеню надійності на 0,1 будуть перераховуватися тривалості робіт, а отже і критичний шлях у графі може змінитись. Наприкінці кожної ітерації так само будемо знаходити значення функції приналежності та функції нечіткої цілі та знаходити оптимальне значення. На рисунку 4.7 зображено граф з усіма фіктивними та дійсними роботами.

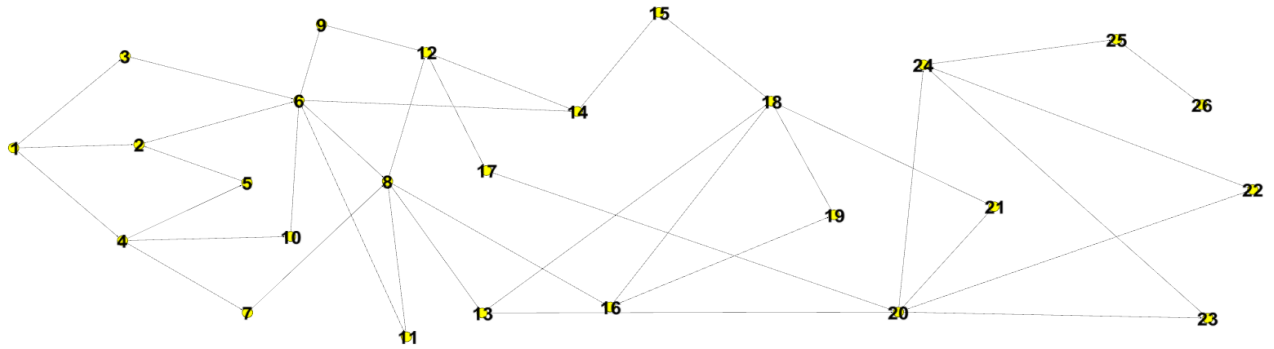


Рисунок 4.7 – Мережевий графік проєкту

Розрахуємо критичний шлях при мінімальному та максимальному значенні надійності для знаходження  $\mu_{\hat{c}}$  та  $\mu_{\hat{d}}$ , а саме коефіцієнти  $T^{\max}$  і  $T^{\min}$ . При  $\mu_{\hat{x}} = 0$  за допомогою бібліотеки networkx було знайдено критичний шлях, що проходить через вершини [1, 4, 5, 2, 6, 11, 8, 16, 19, 18, 21, 20, 23, 24, 25, 26]. Загальна тривалість мінімального шляху – 193 дні. Максимальний же шлях у цьому випадку буде проходити через інші вершини – [1, 4, 5, 2, 6, 8, 16, 19, 18, 21, 20, 22, 24, 25, 26] і його довжина становитиме 315 днів. Тобто у даному випадку незважаючи на те, що кількість робіт була збільшена, але оптимізовано мережевий графік, збільшивши паралельність робіт, то можемо стверджувати, що це відіграло суттєву роль для зменшення максимального критичного шляху проєкту.

Так як знайдено мінімальний критичний шлях проєкту, тож першу ітерацію можна пропустити і перейти відразу до другої ітерації. У даному випадку перерахуємо усі тривалості робіт із надійністю 0,1. Тоді вийде, що робота 1 буде тривати  $(60 - 30) \cdot 0,1 + 30 = 33$  дні, робота 2  $(30 - 14) \cdot 0,1 + 14 = 15,6$ , робота 3 –  $(7 - 3) \cdot 0,1 + 3 = 3,4$  і так далі. Нарешті робота 25 триватиме  $(18 - 10) \cdot 0,1 + 10 = 10,8$  днів. У такому випадку критичний шлях буде проходити також через ті ж вершини [1, 4, 5, 2, 6, 11, 8, 16, 19, 18, 21, 20, 23, 24, 25, 26]. Зобразимо графік для другої ітерації (рис. 4.8).

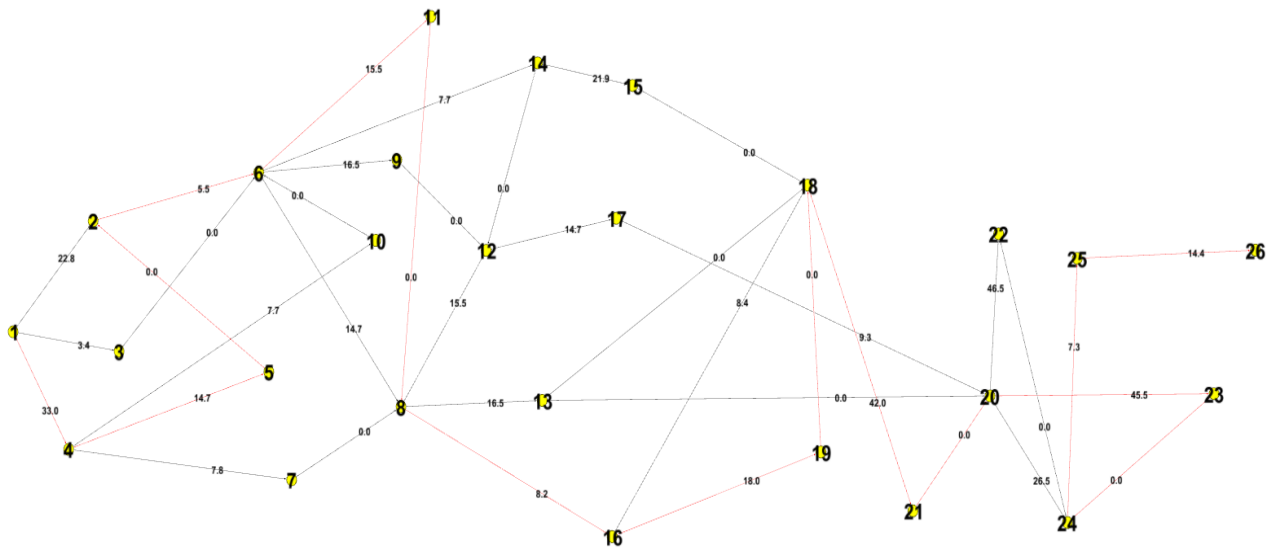


Рисунок 4.8 – Мережевий графік проєкту на другій ітерації

Усі інші ітерації виконуються аналогічно з минулим проєктом, але тут слід відмітити, що на 7 ітерації змінюється критичний шлях проєкту. Тепер він буде проходити через вершини [1, 4, 5, 2, 6, 8, 16, 19, 18, 21, 20, 23, 24, 25, 26]. Якщо порівнювати його з попереднім шляхом, то можна відзначити, що роботу, що йшла з вершини 6 до вершини 11 (тобто №10 – створення сторінки для відображення деталей продукту) замінила робота, що виходить із вершини 6 до вершини 8 (№ 8 – створення каталогу продуктів, призначення цін, категоризація). Якщо робота заміняє одна іншу на критичному шляху, це означає що вона матиме більшу тривалість. Щоб перевірити це, необхідно розрахувати цю тривалість на даній ітерації. У даному випадку  $\mu_{\bar{x}} = 0,6$ , тоді для роботи №10 вона буде рівною  $(20-15) \cdot 0,6 + 15 = 18$ , а для роботи №8 –  $(21-14) \cdot 0,6 + 14 = 18,2$ .

Насправді, цікавою деталлю є ще і те, що на минулій, або ж шостій, ітерації у нас є відразу два критичних шляхи – [1, 4, 5, 2, 6, 8, 16, 19, 18, 21, 20, 23, 24, 25, 26] і [1, 4, 5, 2, 6, 11, 8, 16, 19, 18, 21, 20, 23, 24, 25, 26] через те, що тривалість робіт №8 і №10 є однаковою на цій ітерації. Для роботи №8 тривалість буде розраховуватись за формулою  $(21-14) \cdot 0,5 + 14 = 17,5$ , а для роботи №10 –  $(20-15) \cdot 0,5 + 15 = 17,5$ . Мережеві графіки для шостої та сьомої

ітерацій буде зображено на рисунках 4.9 та 4.10 відповідно.

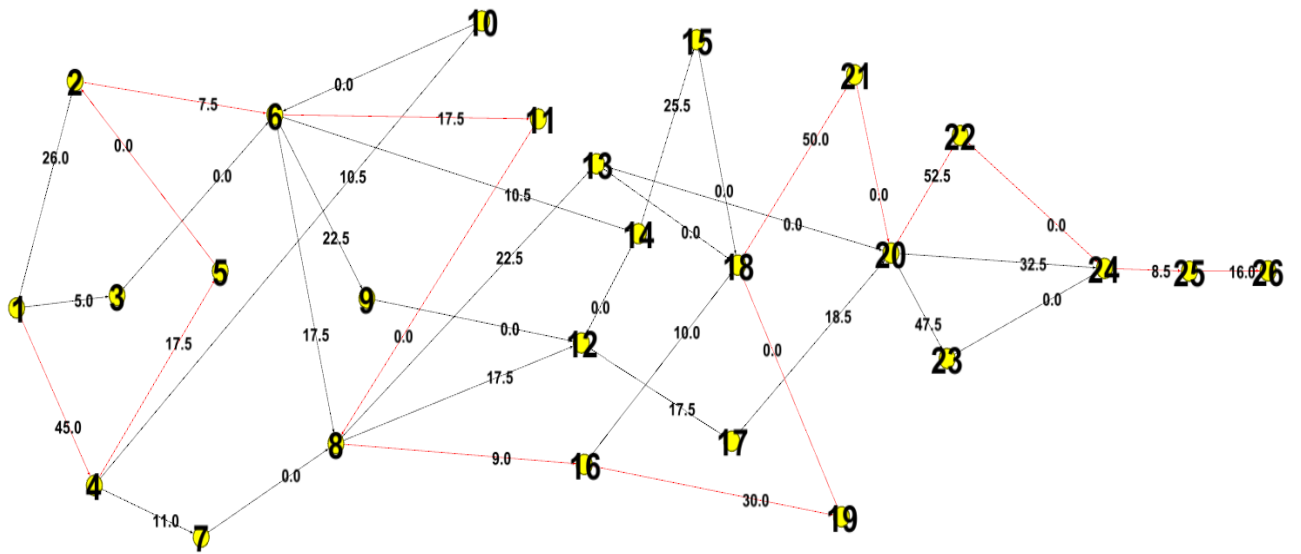


Рисунок 4.9 – Мережевий графік проекту на шостій ітерації

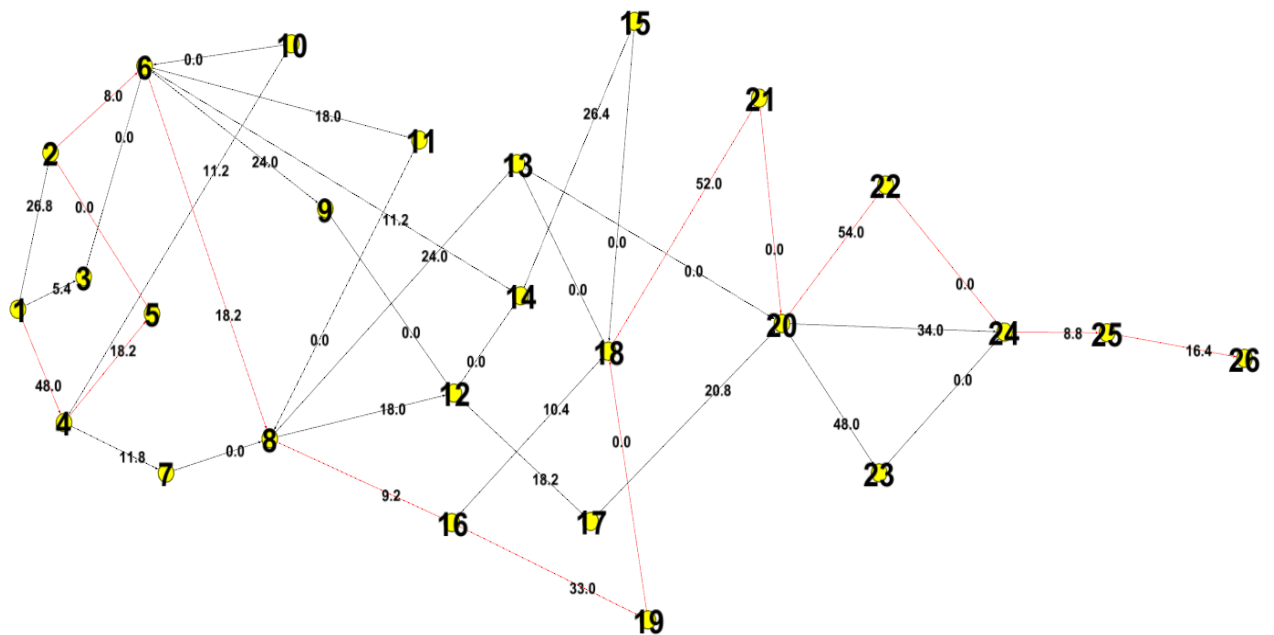


Рисунок 4.10 – Мережевий графік проекту на сьомій ітерації

Після сьомої ітерації критичний шлях вже залишається сталим і для останньої ітерації, як вже було зазначено раніше, він матиме тривалість 315 діб. Усі значення коефіцієнтів  $\mu_{\hat{x}}$ ,  $\mu_{\hat{c}}$  та  $\mu_{\hat{d}}$  та тривалості критичних шляхів було зведено до таблиці 4.6.

Таблиця 4.6 – Результати ітераційного процесу

№	k	$\mu_{\hat{x}}$	Тривалість критичного шляху	$\mu_{\hat{c}}$	$\mu_{\hat{D}}$
1	0	0,0	193	1,0	0,0
2	1	0,1	205,1	0,901	0,1
3	2	0,2	217,2	0,802	0,2
4	3	0,3	229,3	0,702	0,3
5	4	0,4	241,4	0,603	0,4
6	5	0,5	253,5	0,504	0,5
7	6	0,6	265,8	0,403	0,403
8	7	0,7	278,1	0,302	0,302
9	8	0,8	290,4	0,202	0,202
10	9	0,9	302,7	0,101	0,101
11	10	1,0	315	0,0	0,0

Аналізуючи результати, можемо сказати, що максимум функції приналежності для цього проєкту досягається на 6 ітерації при  $\mu_{\hat{x}} = 0,5$ . У даному випадку оптимальним коефіцієнтом надійності залишився той же показник (тобто середнє значення тривалості проєкту між мінімальним та максимальним значенням).

Тож оптимальним планом, що відповідає максимальному значенню перетину між функцією допустимих планів та функцією нечіткої цілі на ітераційному процесі є план з надійністю 50 відсотків та тривалістю 253,5 діб.

Аналізуючи значення надійності, як і у випадку з першим проєктом, можна стверджувати, що коли стоїть задача розрахувати надійність проєкту з 70% впевненістю, тоді проєкт буде завершено не менше, ніж за 278 діб. І навпаки, якщо стоїть задача розрахувати впевненість у тому, що проєкт буде завершено через 230 діб, то надійність буде дорівнювати приблизно 30%.

### 4.3 Розрахунок та порівняння вартості проєктів

Не секрет, що одним із найважливіших економічних параметрів для планування та прийняття рішення про початок проєкту є не тільки час його виконання, але і його вартість. Цей параметр є одним із ключових для менеджерів, а отже, йому дійсно варто приділити особливу увагу задля успішного кінцевого результату та правильного розподілу ресурсів.

Але для того, щоб правильно розрахувати вартість та кошторис усього проєкту, необхідно крім часу, знати також і кількість працівників, що будуть залучені до його реалізації. Цей параметр буде доречно вибрати якраз згідно з кількістю робіт, що можуть бути виконані одночасно для зменшення затримки між виконанням робіт.

Для визначення кількості робіт, що можуть виконані одночасно, використаємо мережеві графіки обох проєктів, що зображені на рисунках 4.2 для першого проєкту та 4.7 для другого проєкту, щоб бачити не тільки кількість інцидентних ребер тій чи іншій вершині, але і тривалість роботи, для того, щоб відрізнити фіктивні і дійсні роботи між собою. Фіктивними роботами можна нехтувати через те, що їхня тривалість є нульовою. Для перевірки також можна використовувати таблиці 4.1 та 4.4, щоб точно розуміти послідовність робіт між собою.

Після аналізу мережевих графіків можна зробити висновок, що для першого проєкту максимальна кількість робіт, що можуть бути виконані одночасно, дорівнює чотирьом. Вони можуть бути представлені у вигляді масивів. Один масив відповідатиме номерам робіт, що можуть виконуватися одночасно. Згідно графіку, таких масивів буде два – це роботи [9, 10, 11, 12], а також [11, 12, 13 і 15]. У такому випадку можемо зробити висновок, що для своєчасного виконання завдань необхідно як мінімум 4 розробники.

Для другого проєкту максимальна кількість одночасних робіт буде дорівнювати 5 роботам. Через те, що цей графік є більш розгалуженим, тож і



кількість масивів буде більшою – [2, 3, 4, 5, 7], [9, 11, 12, 13, 15], [11, 12, 14, 17, 18]. У цьому випадку буде необхідно залучити 5 розробників.

Навіть якщо намагатися йти принципом мінімізації витрат та збереження ресурсів, задля успішної роботи всієї команди проєкту необхідні бізнес-аналітик, проджект-менеджер, DevOps-інженер та тестувальники.

Однозначну відповідь на те, яку кількість тестувальників треба наймати в залежності від розміру команди та кількості розробників, але у середньому ця кількість варіюється у рамках 50 відсотків. Тож нехай для першого проєкту буде залучено два тестувальники (Junior та Senior), а для третього – троє (Junior, Middle та Senior).

Задля більш швидкої роботи над проєктом та можливих форс-мажорних ситуацій можна збільшувати кількість персоналу.

Для комплектування команди розробників першого проєкту необхідний, по-перше, Team Lead спеціаліст, який буде виконувати більш управлінську роботу та вирішувати архітектурні проблеми, іншими трьома розробниками будуть Senior, Middle та Junior спеціалісти.

Для другого ж проєкту можна додатково найняти ще одного інженера Middle рівня для більш усередненого вибору, але ще раз слід підкреслити, що цей вибір залишається на розсуд засновника.

Дані про значення розподілу зарплат можемо взяти на одному з найвідоміших українських сайтів про ІТ – dou.ua. Результати щодо оптимальних планів, які отримані у попередніх пунктах, надають дані про тривалість усього проєкту, а отже можна буде порахувати кількість коштів, що необхідно буде витратити на утримання зазначеної команди спеціалістів.

Іншими витратами, такими як інтеграціями з іншими сервісами та платою за ту чи іншу Еcommerce-платформу враховувати не будемо через те, що ці параметри є дуже специфічними та будуть залежати лише від вимог керівництва, які можуть як збільшити, так і зменшити загальну вартість.

Дані про середні зарплати спеціалістів, що будуть залучені для реалізації проєктів станом на червень 2023 року приведені у таблиці 4.7.

Таблиця 4.7 – Значення середніх місячних зарплат спеціалістів проєкту

Назва роботи	Середня зарплата, \$ (USD)
Team Lead	5500
Senior Software Engineer	5125
Middle Software Engineer	2500
Junior Software Engineer	1000
DevOps	3300
Project Manager	1600
Business Analyst	2200
Junior QA	800
Middle QA	1700
Senior QA	3313

Варто зазначити, що для аналізу були вибрані розробники за спеціалізацією Java та Ecommerce, через те, що більшість Ecommerce-платформ написані саме на мові Java.

Згідно даним у таблиці 4.7 можемо стверджувати, що середня місячна вартість утримання команди спеціалістів для першого проєкту буде дорівнювати  $5500+5125+2500+1000+3300+1600+2200+800+3313=25338\$$ . До команди другого проєкту буде додано ще одного Middle Software Engineer та Middle QA, тож середня вартість зросте на 4200\$ буде дорівнювати 29538\$.

Тепер можна підрахувати вартості мінімального можливого (в залежності від часу виконання), оптимального та максимального планів. У цьому випадку, згідно результатів у таблиці 4.3 мінімальна тривалість проєкту буде дорівнювати 189 діб. Важливою деталлю також буде і те, що при естимачії тривалості роботи враховувалися лише робочі дні, а у графіках середніх зарплат на сайті DOU вказана саме місячна зарплата (з врахуванням вихідних). Тож у нашому випадку необхідно визначити, скільки коштів витрачається на утримання команди за один робочий день, враховуючи, що середня кількість

робочих днів у 2023 році в Україні приблизно дорівнює 22 дням. Тож, у середньому за один робочий день команда витратить  $\frac{25338}{22} = 1151,73$  \$. Тоді мінімально можлива вартість першого проєкту буде становити  $189 \cdot 1151,73 = 217676,97$  \$. Так як тривалість оптимального шляху дорівнює 255 днів, то вартість цього плану дорівнює  $255 \cdot 1151,73 = 293691,15$  \$, а максимального –  $321 \cdot 1151,73 = 369705,33$  \$.

Середньодобова вартість утримання команди для другого проєкту становитиме  $\frac{29538}{22} = 1342,64$  \$. Згідно цього показника таким же чином були підраховані характеристики другого проєкту. Усі результати були зведені до таблиці 4.8.

Таблиця 4.8 – Порівняння економічних характеристик проєктів

Назва проєкту	План		Вартість усього проєкту
	Назва плану	Тривалість	
Створення Backend-частини сайту електронної комерції з послідовним виконанням робіт	Найменш вірогідний	189	217676,97\$
	Оптимальний	255	293691,15\$
	Найбільш вірогідний	321	369705,33\$
Створення Backend-частини сайту електронної комерції з паралельним виконанням робіт	Найменш вірогідний	193	259129,52\$
	Оптимальний	253,5	340359,24\$
	Найбільш вірогідний	315	422931,60\$

Згідно з даними цієї таблиці можемо відмітити те, що різниця у вартості між мінімальними планами буде дорівнювати  $259129,52 - 217676,97 = 41452,55\$$ , між оптимальними планами –  $340359,24 - 293691,15 = 46668,09\$$ , а максимальними –  $422931,60 - 369705,33 = 53226,27\$$ . Але якщо враховувати різницю між цими проєктами, то як вже зазначалося вище, другий проєкт має на три роботи більше (№22, 23 і 25 у другому проєкті), а саме на 52 доби при мінімальній надійності. У такому випадку різниця б склала  $52 \cdot 1151,73 = 59889,96\$$  (сума на утримання команди першого проєкту за 52 доби). Тоді  $59889,96 - 41452,55 = 18437,41\$$ . У найбільш вірогідному випадку  $99 \cdot 1151,73 = 114021,27\$$  (сума на утримання команди першого проєкту за 99 діб).  $114021,27 - 53226,27 = 60795\$$ . Тож економія коштів при правильному плануванні є достатньо значною, особливо у випадку тривалого проєкту.

#### Висновки за розділом 4

У якості досліджуваних проєктів було взято два приклади створення сайту електронної комерції з різними типами зв'язків робіт між ними. Перший проєкт мав незначну кількість паралельних зв'язків та був розрахований на невелику кількість команди та його розробників. Після аналізу послідовності робіт та проведення естимації (визначення мінімальної та максимальної тривалості кожної з робіт) було створено мережевий графік, який у подальшому використовувався для визначення критичного шляху. Ребрами цього графіку були роботи, а кожна вага показувала час виконання такої роботи.

Після побудови графіку було розпочато ітераційний процес з метою визначення найбільш оптимального плану для різних ступенів надійності проєкту від найменш до найбільш вірогідного шансу виконання усіх робіт. Після проведення розрахунків було з'ясовано, що найбільш оптимальним планом є план з надійністю 0,5 (або 50%), що може свідчити до близькості з оцінками, які зазвичай даються для прогнозування реального терміну роботи,

коли між найменшим та найбільшим терміном зазвичай обирають «золоту середину». Тривалість такого плану склала 255 днів, критичний шлях не змінювався впродовж ітераційного процесу.

У другому проєкті було не тільки змінено мережевий графік (тепер використовувався підхід з паралелізацією робіт), але й було додано інші роботи, щоб вдосконалити сам процес створення сайту та подивитися, наскільки критичним ця зміна буде у порівнянні з першим проєктом.

Аналогічним чином було проведено ітераційний процес для другого проєкту. Через більшу кількість розгалужень у графі критичний шлях змінювався впродовж ітераційного процесу, а роботи, що мали більшу продовжуваність покривали термін виконання робіт з низькою продовжуваністю. У результаті навіть з додаванням трьох нових робіт загальна тривалість проєкту зменшилася до 253,5 діб, що свідчить про значний вплив того, як саме побудовано мережевий графік. Кількість розробників проєкту при цьому була збільшена на 2, оптимальний план також мав надійність 0,5.

Також було проведено порівняння вартості обох проєктів для найменш вірогідного, оптимального та найбільш вірогідного згідно даним порталу dou.ua щодо середнього розподілу зарплат по Україні. Звісно ж, через збільшення кількості працівників вартість проєкту №2 зросла, але якщо не враховувати три роботи, що були додані до другого проєкту, то можна дійти висновку, що проєкт є не тільки більш вигідним, але і його реалізація буде набагато швидшою.

## ВИСНОВКИ

У даній кваліфікаційній роботі були розглянуті основні та найпопулярніші методи, що можуть бути застосовані для підрахунку та визначення основних економічних параметрів будь-якого проєкту, у якості прикладу було взято проєкт створення Backend-частини сайту електронної комерції. Основними досліджуваними характеристиками були час виконання проєкту, надійність виконання цього проєкту за відведений час та вартість утримання команди для реалізації такого проєкту, було знайдено оптимальні плани реалізації проєкту з врахуванням виконання усіх наявних робіт до кінця проєкту.

Також детально були розглянуті методи планування та управління проєктами, такі як метод критичного шляху (CPM), метод діаграм Ганта, метод PERT (Project Evaluation and Review Technique), найпопулярніші Agile методи, що використовуються у IT-сфері (Scrum та Kanban), проаналізовані їхні переваги та недоліки.

Для дослідження були використані теорія нечіткої логіки, що використовувалась для представлення часу виконання робіт (естимачії проєкту) за допомогою нечітких чисел, побудови множин допустимих планів та функцій приналежності допустимих планів та нечіткої цілі задля знаходження оптимального плану виконання проєкту. За допомогою теорії графів на основі зв'язків між роботами було побудовано орієнтований граф, що дав уявлення про те, яким саме чином може бути побудовано критичний шлях, дав можливість візуалізувати отримані результати та відобразити усі допустимі плани.

Основні економічні параметри були підраховані для двох варіантів проєкту створення сайту електронної комерції, що може бути створений у двох різних варіаціях – послідовному, коли команда не має багато розробників для його реалізації (або достатньо ресурсів), і паралельному виконанню робіт, що дозволяє оптимізувати час виконання за допомогою залучення додаткових

працівників. Також було підраховано вартість обох проєктів, що у подальшому може допомогти менеджерам зробити висновки щодо впровадження того чи іншого варіанту планування такого типу проєктів.

У роботі також було створено програмний продукт мовою Python і бібліотеки NetworkX, що допомагає швидко розрахувати дані про функції приналежності нечіткої цілі для критичного шляху з різними рівнями надійності, а також візуалізувати отримані результати за допомогою застосунку Gephi.

Слід також зазначити, що даний метод може бути використаний не тільки у сфері інформаційних технологій, але і у багатьох інших галузях діяльності. Він може бути використаний при будівництві складних будівель, планування логістичних завдань, багатоетапних наукових та освітніх проєктах, маркетингових кампаніях та ін.

Отримані результати дослідження мають велику цінність для планування проєкту, для вирішення про доцільність його початку, для прогнозування ресурсів, що знадобляться для його реалізації. Ці показники є вкрай необхідними задля покращення процесів та коректного розподілення робіт, що може допомогти посилити конкурентоспроможність та підвищити прибуток проєкту.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Закутній С. В. Визначення економічних параметрів виконання проєктів, представлених мережею взаємопов'язаних робіт. *27-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті»* : зб. матеріалів форуму. Т. 7. Харків : ХНУРЕ, 2023. С. 133–134.
2. Zadeh L. A. Fuzzy sets and information granularity. Amsterdam : “North-Holland Publishing Co”, 1979. P. 3–18.
3. Bellman R. E., Zadeh L. A. Decision-Making in a Fuzzy Environment. *Management Science*, 1970. Vol. 17, no. 4. P. 142–146.
4. Коньшева Л.К., Назаров Д.М. Основы теории нечетких множеств: Учебное пособие. Санкт-Петербург : “Питер”, 2011. 192 с.
5. Флегонтов А.В., Вилков В.Б., Черных А.К. Моделирование задач принятия решений при нечетких исходных данных. Санкт-Петербург : “Лань”, 2020. 332 с.
6. Алексеева В.А. Теория графов и математическая логика. Практикум: Учебное пособие. Ульяновск : УлГТУ, 2014. 127 с.
7. Зак Ю.А. Принятие решений в условиях нечетких и размытых данных: Fuzzy-технологии. Москва : “Либроком”, 2013. 352 с.
8. Project Management Institute. A guide to project management body of knowledge (PMBOK guide). Pennsylvania, Project Management Institute, 2017. 168 p.
9. Atlassian. URL: <https://www.atlassian.com/ru/agile/project-management/gantt-chart> (дата звернення: 10.11.2023).
10. Бортульов Є.С., Іванов Р.В. Визначення вірогідності реалізації проєкту на основі PERT-методу. Сучасні проблеми моделювання соціально-економічних систем. *Матеріали XI міжнародної науково-практичної Інтернет-конференції 11-12 квітня 2019 р.* : Мультимедійне наук. електрон. вид. Братислава – Харків : ВШЕМ – ХНЕУ ім. С. Кузнеця. 2019.
11. Brewer J.L., Dittman K.C. Methods of IT project management. West Lafayette. Purdue University Press, 2023. 555 p.



12. Андерсон Д. Канбан. Альтернативный путь в Agile ; пер. с англ. А. Коробейникова. Москва : “Манн, Иванов и Фербер”, 2017. 335 с.
13. Medium. URL: <https://towardsdatascience.com/graph-visualisation-basics-with-python-part-iii-directed-graphs-with-graphviz-50116fb0d670> (дата звернения: 13.11.2023).
14. Ross T. J. Fuzzy logic with engineering applications. Chichester : “John Wiley & Sons, Ltd”, 2010. 585 p.
15. Kerzner H. Project Management. A Systems Approach to Planning, Scheduling and Controlling. New Jersey : “John Wiley & Sons, Ltd”, 2010. 1094 p.
16. Gephi. URL: <https://uk.wikipedia.org/wiki/Gephi> (дата звернения: 21.11.2023).