

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів оцінки тривалості операцій ІТ-проекту з міграції ІС до Google Cloud Platform
(тема)

Виконав:
студент 2 курсу, групи УПГІТм-22-1

Шутько Віктор Валерійович
(прізвище, ім'я, по батькові)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва освітньої програми)

Керівник проф. каф. ІУС Ірина ПАНФЬОРОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри



(підпис)

Костянтин ПЕТРОВ

(власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Інформаційних управляючих систем


Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)

« 01 » квітня 20 24 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Шутьку Віктору Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів оцінки тривалості операцій ІТ-проєкту з міграції ІС до Google Cloud Platform
затверджена наказом університету від 01 квітня 2024 р. № 258См
2. Термін подання студентом роботи до екзаменаційної комісії 01.06.2024
3. Вихідні дані до роботи науково-технічні публікації та інтернет-джерела з тематики кваліфікаційної роботи
4. Перелік питань, що потрібно опрацювати в роботі аналіз існуючих методів оцінки тривалості операцій в контексті ІТ-проєктів з міграції інформаційних систем до Google Cloud Platform, формування мети та постановка задачі кваліфікаційної роботи, розробка/модифікація методу оцінки тривалості виконання операцій ІТ-проєкту з міграції інформаційних систем до Google Cloud Platform, розробка алгоритму оцінки тривалості операцій за розробленим/модифікованим методом, апробація методу для оцінки тривалості операцій ІТ-проєкту з міграції інформаційної системи до Google Cloud Platform, порівняння точності оцінок оригінального та розробленого/модифікованого методів, визначення придатності розробленого/модифікованого методу до оцінки тривалості операцій ІТ-проєкту з міграції ІС до Google Cloud Platform


КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз літератури та інтернет джерел	01.04.2024–20.04.2024	Виконано
2	Опис постановки задачі дослідження	20.04.2024–21.04.2024	Виконано
4	Аналіз існуючих методів оцінки тривалості операцій ІТ-проекту з міграції інформаційної системи до Google Cloud Platform	21.04.2024–05.05.2024	Виконано
5	Розробка методу параметричного оцінювання на базі методу PERT	05.05.2024–15.05.2024	Виконано
6	Розробка та опис інформаційної технології оцінки тривалості операцій ІТ-проекту з міграції інформаційної системи до Google Cloud Platform	15.05.2024–20.05.2024	Виконано
7	Апробація та порівняння розробленого методу параметричного оцінювання на прикладі кількох ІТ-проектів з міграції інформаційної системи Google Cloud Platform	20.05.2024–25.05.2024	Виконано
8	Підготовка пояснювальної записки до кваліфікаційної роботи	25.05.2024–26.05.2024	Виконано
9	Підготовка графічного матеріалу	25.05.2024–26.05.2024	Виконано
10	Підготовка презентаційних матеріалів	26.05.2024–27.05.2024	Виконано

Дата видачі завдання 01 квітня 2024 р.

Студент _____

 (підпис)

Керівник роботи _____

 (підпис) проф.каф.ІУС Ірина ПАНФЬОРОВА
 (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 127 стор., 18 рисунків, 21 таблиця, 42 джерела, 1 додаток.

МЕТОД PERT, МІГРАЦІЯ У ХМАРУ, ОЦІНКА ТРИВАЛОСТІ, ПЛАНУВАННЯ, ПРОЄКТ, РОЗРОБКА, ХМАРНА ПЛАТФОРМА.

Актуальність даної роботи проявляється в необхідності забезпечення аналітиків та менеджерів проєктів методом оцінювання тривалості операцій, точнішим за існуючі в контексті ІТ-проєктів з міграції інформаційних систем до хмарних платформ.

Об'єктом дослідження кваліфікаційної роботи є процес оцінки тривалості операцій ІТ-проєкту з міграції інформаційної системи до хмарної платформи.

Метою досліджень кваліфікаційної роботи є підвищення точності оцінки тривалості операцій в контексті ІТ-проєктів з міграції інформаційної системи до Google Cloud Platform.

Отримані наукові результати: розроблено метод параметричного оцінювання на базі методу PERT для підвищення точності оцінки тривалості операцій ІТ-проєктів з міграції інформаційних систем Google Cloud Platform. Запропоновано та реалізовано алгоритм використання, а також виконано апробацію синтезованого методу на прикладі кількох ІТ-проєктів з міграції інформаційної системи до Google Cloud Platform.

Результати кваліфікаційної роботи можуть бути впроваджені в діяльність ІТ-компаній, які займаються міграціями інформаційних систем до Google Cloud Platform.

ABSTRACT

The explanatory note of the qualification work: 127 pages, 18 figures, 21 tables, 42 sources, 1 appendix.

CLOUD PLATFORM, DEVELOPMENT, DURATION ESTIMATION, MIGRATION, PERT METHOD, PLANNING, PROJECT.

The relevance of this work is manifested in the need to provide analysts and project managers with a method for estimating the duration of operations, which is more accurate than the existing migration of information systems to cloud platforms in the context of IT projects.

The object of research of the qualification work is the process of assessing the duration of operations of the IT project of migration of the information system to the cloud platform.

The purpose of the qualification work research is to improve the accuracy of estimating the duration of operations in the context of IT projects for the migration of an information system to Google Cloud Platform.

Obtained scientific results: a method of parametric estimation based on the PERT method has been developed to improve the accuracy of estimating the duration of operations of IT projects for the migration of information systems to the cloud. An algorithm of use is proposed and implemented, as well as testing of the synthesized method is carried out on the example of several IT projects for migration of an information system to Google Cloud Platform.

The results of the qualification work can be implemented in the activities of IT companies that are engaged in the migration of information systems to Google Cloud Platform.

ЗМІСТ

	С.
Скорочення та умовні позначки.....	9
Вступ.....	10
1 Аналіз існуючих методів оцінки тривалості операцій ІТ-проєкту з міграції ІС до Google Cloud Platform	11
1.1 Аналіз відмінностей on-premises та хмарного підходів до розробки ІС ...	11
1.1.1 Особливості on-premises підходу до розробки ІС.....	11
1.1.2 Особливості розробки ІС з використанням хмарних технологій.....	13
1.1.3 Релевантність on-premises та хмарного підходу до розробки ІС.....	15
1.2 Місце міграції ІС до хмари на етапі експлуатації та супроводу ІС	16
1.3 Аналіз основних методів оцінки тривалості операцій ІТ-проєкту з міграції ІС до хмари	17
1.3.1 Основні методи оцінки тривалості операцій ІТ-проєкту.....	17
1.3.2 Метод експертної оцінки	17
1.3.3 Метод оцінювання за аналогами	18
1.3.4 Методи параметричного оцінювання.....	19
1.3.5 Метод PERT	20
1.3.6 Метод аналізу даних (альтернатив, резервів).....	21
1.3.7 Метод наради	22
1.4 Аналіз існуючих ІС оцінки тривалості операцій ІТ-проєкту з міграції ІС до хмари.....	23
1.4.1 ІС управління розкладом ІТ-проєкту	23
1.4.2 Microsoft Project.....	24
1.4.3 Primavera P6	26
1.4.4 Smartsheet	27
1.4.5 Trello	28

1.4.6 Asana	30
1.4.7 Atlassian JIRA.....	31
1.4.8 Basecamp	32
1.5 Основні хмарні провайдери	33
1.5.1 Порівняльна характеристика хмарних провайдерів	33
1.5.2 Google Cloud Platform	33
1.5.3 Amazon Web Services.....	35
1.5.4 Microsoft Azure	37
1.5.5 IBM Cloud.....	38
1.5.6 Oracle Cloud Infrastructure.....	39
1.5.7 Обґрунтування вибору провайдера Google Cloud Platform.....	40
1.6 Аналіз міграції ІС до хмарного сервісу Google Cloud Platform.....	41
1.6.1 Способи міграції ІС до хмарного сервісу GCP	41
1.6.2 Вибір сховища даних	42
1.6.3 Вибір обчислювальної платформи	46
1.6.4 Організація комунікації між застосунками.....	50
1.7 Особливість задачі оцінки тривалості операцій в контексті ІТ-проектів з міграції ІС до GCP.....	55
1.8 Формування мети та постановка задачі кваліфікаційної роботи	56
2 Розробка методу параметричного оцінювання на базі методу PERT.....	57
2.1 Опис методу PERT	57
2.1.1 Використання методу PERT для оцінки тривалості операцій	57
2.1.2 Недоліки методу PERT для оцінки тривалості операцій.....	58
2.2 Розробка методу параметричного оцінювання на базі методу PERT	60
2.3 Визначення критеріїв формування коефіцієнтів для методу параметричного оцінювання на базі методу PERT	61
2.4 Етапи оцінювання тривалості операцій ІТ-проєкту з використанням методу параметричного оцінювання на базі методу PERT	62

2.5 Висновки	63
3 Інформаційна технологія оцінки тривалості операцій ІТ-проекту з міграції ІС до Google Cloud Platform.....	64
3.1 Опис інформаційної технології.....	64
3.2 Впровадження інформаційної технології	69
3.3 Висновки	70
4 Експериментальне підтвердження результатів дослідження.....	71
4.1 Набір даних для порівняння точності оцінювання оригінального та синтезованого методів.....	71
4.2 Оцінка тривалості операцій за оригінальним методом PERT	79
4.3 Оцінка тривалості операцій за методом параметричного оцінювання на базі методу PERT	87
4.4 Порівняльний аналіз точності оцінювання оригінального та синтезованого методів.....	99
4.5 Висновки	106
Висновки	108
Перелік джерел посилання	110
Додаток А Графічний матеріал	114

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

- ВКО – ваговий коефіцієнт оптимізму
- ВКП – ваговий коефіцієнт песимізму
- ВКР – ваговий коефіцієнт реалізму
- ІС – інформаційна система
- Н – найімовірніша оцінка
- О – оптимістична оцінка
- П – песимістична оцінка
- API – Application Programming Interface
- AWS – Amazon Web Services
- CPM – Critical Path Method
- GAE – Google App Engine
- GCE – Google Compute Engine
- GCP – Google Cloud Platform
- GKE – Google Kubernetes Engine
- HTTP – Hypertext Transfer Protocol
- JBE – Junior Back-End Engineer
- JSON – JavaScript Object Notation
- MBE – Middle Back-End Engineer
- OCI – Oracle Cloud Infrastructure
- PERT – Program Evaluation and Review Technique
- REST – Representational State Transfer
- SA – Solutions Architect
- SBE – Senior Back-End Engineer
- TL – Software Engineering Team Lead
- VM – віртуальна машина

ВСТУП

В епоху цифрової трансформації міграція інформаційних систем (ІС) до хмарних платформ відокремлюється як один з ключових аспектів стратегічного розвитку для багатьох організацій. Це пояснюється тим, що хмари пропонують ряд переваг, які можуть значно покращити ефективність, гнучкість, безпеку та економічну ефективність ІТ-інфраструктури.

Міграція ІС до хмарної платформи виконується в межах спеціально організованих ІТ-проектів. ІТ-проекти з міграції ІС до хмари, як і усі інші проекти, потребують ретельного планування усіх видів ресурсів – часових, кадрових та фінансових. Планування часових ресурсів проекту включає в себе оцінку тривалості виконання проекту. Оцінка тривалості виконання проекту, в свою чергу, складається з оцінок тривалості усіх операцій проекту. Таким чином формується задача оцінки тривалості операцій ІТ-проекту з міграції ІС до хмари.

Для вирішення задачі оцінки тривалості операцій ІТ-проекту були розроблені спеціальні методи оцінювання. Однак, точність таких методів в контексті міграції ІС до хмари варіюється, в силу особливостей такого типу проектів.

Головною метою роботи є підвищення точності оцінювання тривалості операцій ІТ-проектів саме в контексті міграції ІС до Google Cloud Platform. Для цього планується створення або модифікація існуючого методу оцінки тривалості операцій, використання розробленого методу на реальних проектних даних та порівняння нового методу з оригінальним для визначення придатності розробки до використання на реальних проектах.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОЦІНКИ ТРИВАЛОСТІ ОПЕРАЦІЙ ІТ-ПРОЄКТУ З МІГРАЦІЇ ІС ДО GOOGLE CLOUD PLATFORM

1.1 Аналіз відмінностей on-premises та хмарного підходів до розробки ІС

1.1.1 Особливості on-premises підходу до розробки ІС

On-premises підхід до розробки ІС означає, що всі необхідні ресурси та інфраструктура для розробки, випробування, розгортання та експлуатації системи знаходяться у власності та під контролем самої організації.

У такому підході всі сервери, засоби зберігання даних, мережеві компоненти, апаратне та програмне забезпечення розташовані на внутрішніх серверних приміщеннях або дата-центрах організації. Команда розробників має повний контроль над всіма аспектами ІС, включаючи апаратне забезпечення, операційну систему, мережу, бази даних та додаткове програмне забезпечення.

Основні переваги on-premises підходу:

- контроль і безпека даних;
- гнучкість та налаштування;
- доступ до даних;
- контроль над середовищем;
- відсутність залежності від зовнішніх постачальників.

Організація має повний контроль над своїми даними, оскільки вони зберігаються та обробляються на внутрішніх серверах. Це може бути особливо важливо для організацій, які працюють з конфіденційною чи регульованою інформацією.

З on-premises розробкою, організація може налаштувати середовище розробки та експлуатації системи згідно зі своїми потребами. Це дозволяє керувати рівнем продуктивності, масштабованістю та налаштуваннями забезпечення.

Організація також має прямий фізичний доступ до своїх серверів та даних. Це може бути корисним для задач резервного копіювання, відновлення даних та проведення досліджень. Організація може встановити власні політики щодо зберігання та резервного копіювання даних, що дозволяє забезпечити високий рівень безпеки і надійності.

Також, організація має повний контроль над конфігурацією та розгортанням інфраструктури. Це дозволяє гнучко налаштовувати сервери, мережу та інші компоненти згідно з потребами проекту або організації.

Організація не залежить від зовнішніх постачальників хмарних послуг або інших провайдерів. Це може бути важливою перевагою в ситуаціях, коли доступ до хмарної інфраструктури обмежений або необхідно дотримуватись певних внутрішніх політик організації.

Однак, on-premises підхід також має свої недоліки:

- високі витрати;
- обмежена масштабованість;
- відповідальність за підтримку;
- відсутність географічної гнучкості.

Розгортання та підтримка власної інфраструктури вимагає значних капітальних витрат на придбання та підтримку обладнання, програмного забезпечення та персоналу.

Власна інфраструктура може мати обмежені межі масштабованості порівняно з хмарними рішеннями, що можуть привести до складнощів у випадку необхідності розширення або зменшення потужності системи. Встановлення додаткового обладнання або зміна конфігурації може бути трудомістким і вимагати часу.

Організація, яка обирає on-premise підхід, повинна мати власні засоби і експертизу для підтримки та адміністрування інфраструктури. Це включає в себе завдання, такі як моніторинг, планування резервного копіювання, оновлення програмного забезпечення та вирішення технічних проблем.

У випадку on-premises підходу система розгортається в одному фізичному місці або в декількох розташованих рядом місцях. Це може бути обмеженням, якщо необхідно мати доступ до системи з різних географічних регіонів або для забезпечення стійкості системи.

1.1.2 Особливості розробки ІС з використанням хмарних технологій

Підхід до розробки ІС з використанням хмарних технологій базується на використанні хмарних обчислювальних ресурсів та послуг для розробки, тестування, розгортання та експлуатації системи. У цьому підході інфраструктура та ресурси розташовуються на хмарних платформах, які надають доступ до обчислювальних потужності, мережі, сховищ даних та інших послуг через Інтернет.

Основні переваги підходу з використанням хмарних технологій:

- масштабованість;
- економічність;
- гнучкість;
- висока доступність та надійність;
- швидкість розгортання;
- постійні оновлення та підтримка;
- глобальний доступ.

Хмарні платформи дозволяють легко масштабувати ресурси в залежності від потреб системи. Є можливість збільшувати або зменшувати потужність обчислювальних ресурсів, мережі та зберігання даних згідно з вимогами проекту, що забезпечує гнучкість та оптимальне використання ресурсів;

Замість інвестування у власну інфраструктуру, можна використовувати хмарні ресурси по моделі плати за використання (pay-as-you-go), тобто сплата

йде лише за ті ресурси, які використовуються, що дозволяє знизити початкові витрати та оптимізувати витрати на обслуговування інфраструктури.

Є можливість швидко розгорнути та випробувати нові системи або функції без необхідності придбання та налаштування нового обладнання. Можна використовувати готові сервіси та рішення, які надаються хмарними постачальниками, що спрощує розробку та зменшує час до введення в експлуатацію.

Хмарні платформи зазвичай мають високий рівень доступності та надійності. Вони забезпечують резервне копіювання даних, реплікацію серверів та автоматичне масштабування для забезпечення безперебійної роботи системи. Використання географічно розподілених центрів обробки даних забезпечує високу стійкість системи.

Розгортання системи може бути значно прискорене. Можливим та доцільним є використовувати шаблони конфігурацій та автоматизовані процеси розгортання, що дозволяє швидко створювати та запускати нові середовища для розробки та тестування.

Хмарні постачальники активно розвивають свої послуги та пропонують постійні оновлення. Використання нових функцій є доцільним та зумовлене покращеннями безпеки та вдосконаленими сервісами, необхідними для ефективної розробки та експлуатації системи. Постачальники також забезпечують підтримку та технічну допомогу у разі потреби.

З хмарними технологіями, доступ до систем можна отримати з будь-якого місця, де є Інтернет. Це спрощує роботу з розподіленими командами, дозволяє віддалений доступ для співробітників та забезпечує мобільність розробників та адміністраторів системи.

Проте, підхід з використанням хмарних технологій також має свої недоліки:

- залежність від постачальників;
- проблеми безпеки;
- обмежений контроль над інфраструктурою;

– залежність від Інтернет-з'єднання.

Якщо постачальник зазнає проблем, може виникнути перебої в роботі системи. Також існує ризик, що постачальник може змінити свої умови або послуги, що може вплинути на систему.

Збереження конфіденційності та захисту даних є важливими аспектами розробки ІС. Використання хмарних технологій означає передачу та зберігання даних на серверах постачальника, що може викликати певні стурбованості щодо безпеки.

Використання хмарних послуг надає обмежений контроль над фізичною інфраструктурою та налаштуваннями. При такому підході до розробки ІС можна розраховувати лише на функціонал та обмеження, надані хмарним постачальником.

Для доступу до хмарних послуг необхідне стабільне та швидке Інтернет-з'єднання. Якщо наявне з'єднання є обмеженим або нестабільним, це може призвести до затримок та проблем з доступом до системи.

1.1.3 Релевантність on-premises та хмарного підходу до розробки ІС

Вибір між on-premise підходом та хмарним підходом до розробки ІС залежить від конкретних потреб, вимог та обставин організації. On-premise підхід слід обирати, коли:

- є специфічні вимоги щодо безпеки та контролю над даними, особливо в регульованому середовищі;
- є потреба в повному контролі над інфраструктурою та налаштуваннями системи;
- обсяги даних великі, і потрібен прямий доступ до них без залежності від мережі.

Хмарний підхід є сенс обирати, коли:

- потрібна гнучкість та масштабованість, особливо якщо потрібно швидко збільшити або зменшити ресурси залежно від потреб системи;
- є необхідність знизити початкові витрати та оптимізувати витрати на обслуговування інфраструктури;
- потрібен глобальний доступ до системи та можливість працювати з розподіленими командами.

Потрібно ретельно провести аналіз вимог, бюджету, рівня безпеки та контролю, а також оцінити наявні ресурси та експертизу організації для оптимального вибору між on-premise підходом та хмарним підходом. Необхідно врахувати, що також можлива комбінація обох підходів, використовуючи гібридні рішення для досягнення найкращого результату. Деякі організації можуть використовувати комбінацію обох підходів, використовуючи хмарні рішення для певних функцій або застосунків, а on-premises інфраструктуру для більш критичних чи регульованих вимог. [1]

1.2 Місце міграції ІС до хмари на етапі експлуатації та супроводу ІС

Експлуатація та супровід ІС – один з основних етапів життєвого циклу ІС. Супровід ІС – це набір організаційних і технічних заходів, спрямованих на забезпечення нормальної роботи і підтримку ІС протягом їх життєвого циклу. Це включає в себе такі аспекти, як оновлення, розвиток та вирішення проблем, що виникають під час експлуатації системи. Міграція ІС до хмари торкається кожного з перелічених аспектів супроводу ІС, а саме: оновлення та адаптація системи під середовище обраного хмарного провайдера, розвиток системи з врахуванням тих можливостей, що надає обраний хмарний провайдер та вирішення проблем, які з'явилися та потенційно можуть з'явитися при експлуатації системи після міграції ІС до середовища обраного хмарного провайдера. Отже, аналізуючи міграцію ІС до хмари з точки зору життєвого

циклу ІС, можна зробити висновок, що вона є може бути додатковим аспектом етапу супроводу ІС.

1.3 Аналіз основних методів оцінки тривалості операцій ІТ-проєкту з міграції ІС до хмари

1.3.1 Основні методи оцінки тривалості операцій ІТ-проєкту

Оцінка тривалості операцій проєкту є однією з найважливіших складових в плануванні та управлінні будь-яким проєктом. Для досягнення успішних результатів та вчасного виконання проєкту, важливо точно визначити тривалість кожної операції. Для цього використовуються різні методи оцінки, кожен з яких має свої переваги та обмеження. Основні методи оцінки тривалості операцій проєкту:

- експертна оцінка;
- оцінювання за аналогами;
- методи параметричного оцінювання;
- метод PERT;
- аналіз даних (альтернатив та резервів);
- наради.

1.3.2 Метод експертної оцінки

Метод експертної оцінки тривалості операцій у ІТ-проєкті є стратегічно важливим інструментом для точного прогнозування часу, який необхідний для успішного завершення різних завдань та операцій, що включені до проєкту.

Цей метод базується на знаннях і досвіді фахівців, які володіють відповідною компетентністю у сфері інформаційних технологій і мають розуміння специфіки проєктного управління.

Основна ідея полягає в тому, що експерти, зазвичай члени проєктної команди або фахівці з конкретної області ІТ, надають оцінки часу, який потрібний для виконання окремих операцій чи завдань. Ці оцінки можуть виражатися у годинах, днях, тижнях або інших відповідних одиницях часу.

Важливим кроком є формування експертної групи, в яку входять особи з різним досвідом та компетентністю з метою визначення тривалості операцій. Спільне обговорення та обґрунтування оцінок має на меті досягнення консенсусу між експертами.

Після отримання оцінок для кожної окремої операції, може бути проведено обчислення середнього значення, а також розглянуті можливі ризики та невизначеності, що можуть вплинути на тривалість операцій. Результатом є деталізована і достовірна інформація про тривалість кожної операції, яка служить основою для подальшого планування і управління ІТ-проєктом [2].

1.3.3 Метод оцінювання за аналогами

Метод оцінювання за аналогами тривалості операцій в ІТ-проєкті є аналітичним підходом, який базується на порівнянні схожих проєктів або операцій, що вже були виконані в минулому, з поточним завданням з метою прогнозування тривалості останнього. Цей метод ґрунтується на припущенні, що операції зі схожими характеристиками матимуть схожі тривалості виконання.

Під час використання методу оцінювання за аналогами, спочатку визначаються схожість між поточним проєктом і аналогічними проєктами чи операціями. Схожість може включати такі параметри, як обсяг робіт, складність

завдань, використані технології, склад персоналу тощо. Після цього аналогічні операції або проекти аналізуються на предмет історичних даних про тривалість їхнього виконання.

Для отримання прогнозу тривалості поточного проекту на основі аналогів проводять аналіз цих історичних даних та роблять висновки. Важливо враховувати можливі відмінності та унікальні аспекти поточного проекту, які можуть вплинути на тривалість операцій та зробити корекції в прогнозах.

Перевагою методу оцінювання за аналогами є можливість використання вже існуючого досвіду та даних для створення більш обґрунтованих прогнозів тривалості операцій в ІТ-проектах. Однак важливо пам'ятати, що точність прогнозів може залежати від якості вибору аналогів та аналізу даних [3].

1.3.4 Методи параметричного оцінювання

Методи параметричного оцінювання тривалості операцій в ІТ-проекті є математичними та статистичними підходами, які базуються на аналізі структурних параметрів операцій та використанні певних математичних моделей для прогнозування часових рамок. Ці методи ґрунтуються на припущенні, що існує певний математичний зв'язок між тривалістю операцій і їхніми характеристиками.

Прикладами методів параметричного оцінювання тривалості операцій ІТ-проекту є:

- метод лінійної регресії;
- метод аналізу статистичних розподілів;
- метод Монте-Карло;
- метод аналізу кривих настання подій.

У методі лінійної регресії використовується лінійна математична модель, що виражає залежність між тривалістю операції і одним або кількома

факторами, такими як обсяг робіт, кількість ресурсів, обладнання тощо. Наприклад, може бути встановлено, що час виконання операції зростає лінійно зі збільшенням обсягу робіт.

В аналізі статистичних розподілів тривалість операції оцінюється на основі відомого статистичного розподілу, такого як нормальний, експоненціальний, Вейбулла тощо. Використання цих розподілів дозволяє моделювати величину тривалості і прогнозувати його параметри.

Метод Монте-Карло використовує стохастичну симуляцію для прогнозування тривалості операцій. Він передбачає генерацію випадкових чисел для параметрів, які впливають на тривалість, і повторення цього процесу багато разів для отримання розподілу можливих тривалостей.

За допомогою методів параметричного оцінювання можна отримати більш точні прогнози тривалості операцій, але для їхнього успішного використання необхідно мати великий обсяг якісних даних і розуміти припущення, які лягли в основу вибору математичної моделі чи статистичного розподілу. Окрім того, важливо постійно оновлювати та коригувати моделі відповідно до змін у проєкті.

1.3.5 Метод PERT

Метод PERT (Program Evaluation and Review Technique) є ефективним для прогнозування тривалості операцій в IT-проєктах. Цей метод передбачає використання трьох оцінок для кожної операції: оптимістичної, песимістичної і найімовірнішої. Ці оцінки використовуються для створення більш об'єктивного і надійного прогнозу тривалості операції.

Оптимістична оцінка відображає той сценарій, в якому всі обставини сприяють швидкому виконанню операції. Це може бути, наприклад, той час,

який потрібний, якщо всі ресурси доступні і працюють на повну потужність, а ризики мінімізовані.

Песимістична оцінка відображає найгірший можливий сценарій, де всі можливі ризики та фактори, що заважатимуть, враховуються. Ця оцінка відображає найбільший час, який може бути потрібний для виконання операції.

Найімовірніша оцінка враховує реалістичний сценарій на основі досвіду та експертних знань. Вона відображає час, який ймовірно буде витрачено на операцію при нормальних умовах.

За допомогою цих трьох оцінок, можна вирахувати середнє значення і дисперсію для тривалості операції. Ці значення використовуються для розрахунку імовірнісного розподілу тривалості операції.

Метод PERT дозволяє краще враховувати ризики та невизначеності, які можуть вплинути на тривалість операцій, і створює більш об'єктивні та надійні прогнози, що допомагає управляти проектом ефективніше та планувати ресурси [4].

1.3.6 Метод аналізу даних (альтернатив, резервів)

Метод аналізу даних для оцінки тривалості операцій в ІТ-проекті – це системний процес, який передбачає збір, обробку і інтерпретацію існуючої інформації з метою визначення тривалості операцій. Цей метод базується на аналізі реальних даних, що стосуються подібних операцій або проектів. Важливим є збирання якісної та кількісної інформації, яка може включати в себе історичні дані про тривалість операцій, ресурси, які були задіяні, а також умови, за яких проводилися роботи.

Перший крок у використанні методу аналізу даних – це систематичний збір і підготовка існуючої інформації. Це може включати в себе огляд

документації, аналіз історичних записів проєктів, опитування експертів та вивчення даних, пов'язаних з подібними завданнями.

Далі проводиться аналіз зібраних даних для виявлення закономірностей та паттернів, що можуть впливати на тривалість операцій. Цей аналіз може включати в себе використання статистичних методів, таких як регресійний аналіз, аналіз варіації, а також визначення середніх та медіанних значень.

На основі аналізу даних формується прогноз тривалості операцій в ІТ-проєкті, а також оцінка ризиків і можливостей, які можуть вплинути на цю тривалість. Оцінка проводиться з урахуванням варіабельності і невизначеності, що можуть бути пов'язані з операціями проєкту.

Метод аналізу даних дозволяє підтвердити оцінки тривалості операцій на підставі реальних даних, що забезпечує більш точну та обґрунтовану основу для планування та управління ІТ-проєктом.

1.3.7 Метод наради

Метод наради в оцінці операцій ІТ-проєкту є одним із способів залучення експертів та зацікавлених сторін до процесу визначення тривалості та успішності операцій. Цей метод передбачає проведення зустрічей або нарад, на яких експерти та члени проєктної команди обговорюють та оцінюють різні аспекти операцій, спираючись на свій досвід та знання.

Нарада починається з визначення конкретних цілей і завдань. Це може включати в себе обговорення того, що очікується від операцій, які ресурси доступні, які ризики і можливості існують. Далі учасники обговорюють різні альтернативні підходи та стратегії для виконання операцій.

Експерти та учасники наради можуть внести свої власні оцінки та пропозиції щодо тривалості, технічних аспектів, бюджету та інших параметрів

операцій. Цей обмін ідеями та думками допомагає збагатити процес прийняття рішень та прийняти більш обґрунтовані рішення.

Крім того, нарада може бути використана для вирішення конфліктних ситуацій або роз'яснення питань, які можуть виникнути в процесі оцінки операцій. Учасники наради мають можливість обговорити різні точки зору та прийти до консенсусу щодо важливих аспектів операцій.

Один з важливих аспектів методу наради – це документація обговореного та прийнятих рішень. Це дозволяє зберегти історію прийнятих рішень та аргументи, які враховувалися при оцінці операцій.

У підсумку, метод наради в оцінці операцій ІТ-проєкту відіграє важливу роль у залученні експертів і зацікавлених сторін для спільного прийняття обґрунтованих рішень щодо тривалості та успішності операцій, що сприяє досягненню поставлених цілей проєкту [5].

1.4 Аналіз існуючих ІС оцінки тривалості операцій ІТ-проєкту з міграції ІС до хмари

1.4.1 ІС управління розкладом ІТ-проєкту

Управління розкладом проєктів включає в себе використання різних інструментів і методів для планування, виконання та контролю проєктів. Основні ІС, які можуть бути використані для управління розкладом проєктів, включають:

- Microsoft Project;
- Primavera P6;
- Smartsheet;
- Trello;
- Asana;
- Atlassian JIRA;

– Basecamp.

Microsoft Project дозволяє створювати, редагувати та відстежувати розклади проєктів. Вона надає можливості для визначення завдань, ресурсів і залежностей між ними.

Primavera P6 використовується для великих та складних проєктів, зокрема в будівництві та інфраструктурних роботах. Вона дозволяє детально планувати, відстежувати та аналізувати розклади.

Smartsheet надає інструменти для планування проєктів, включаючи створення графіків робіт, спільну роботу команди та відстеження виконання завдань.

Trello в основному використовується для керування завданнями та проєктами на дошках. Також може бути використана для створення і відстеження розкладів проєктів у формі карточок та списків.

Asana дозволяє створювати завдання, проєкти та графіки виконання, а також відстежувати прогрес у реальному часі.

JIRA зазвичай використовується для управління розробкою програмного забезпечення, але також може бути використана для створення і відстеження розкладів проєктів.

Basecamp спрямована на спільну роботу команди та включає інструменти для створення розкладів та відстеження завдань.

1.4.2 Microsoft Project

Microsoft Project – це ІС, призначена для управління проєктами, яка розроблена корпорацією Microsoft. Ця ІС відома своєю здатністю надавати інструменти для створення, аналізу та керування розкладами та завданнями в рамках різноманітних проєктів. Microsoft Project використовується в широкому спектрі сфер, включаючи інженерію, будівництво, інформаційні технології,

дослідження та інші галузі, де ефективне управління часом та ресурсами є ключовими факторами успіху.

Ця ІС дозволяє користувачам створювати докладні розклади завдань, визначати послідовності та залежності між ними, призначати ресурси для виконання завдань і контролювати виконання проєкту в реальному часі. Microsoft Project забезпечує можливість візуалізації графіка виконання завдань у вигляді діаграми Ганта, яка допомагає в зрозумінні та аналізі ходу проєкту.

Однією з ключових особливостей Microsoft Project є можливість оптимізації використання ресурсів та контролю над бюджетом проєкту. Користувачі можуть встановлювати пріоритети завдань, розподіляти ресурси ефективніше і відстежувати витрати.

Microsoft Project підтримує інтеграцію з іншими ІС, що дозволяє обмінюватися даними з іншими програмами, такими як Microsoft Excel і Microsoft SharePoint. Ця можливість сприяє зручному обміну інформацією та спільній роботі команди над проєктом.

Окрім цього, Microsoft Project дозволяє використовувати різні методи управління проєктами, такі як Waterfall, Agile, Scrum і багато інших. Це дозволяє підлаштовувати підходи до керування проєктом під конкретні вимоги та характеристики проєкту.

Ще однією значущою функцією Microsoft Project є можливість генерувати звіти та аналітику проєкту, що спрощує процес прийняття рішень та сприяє ефективному комунікації зі всіма зацікавленими сторонами. Ця здатність допомагає забезпечити прозорість та взаєморозуміння всередині команди проєкту і з зовнішніми зацікавленими сторонами.

У великих організаціях Microsoft Project може інтегруватися з іншими корпоративними системами для забезпечення синхронізації даних та більшого рівня автоматизації процесів управління проєктами.

Отже, Microsoft Project є важливим інструментом для ефективного управління проєктами, який допомагає організаціям досягати своїх цілей,

визначати пріоритети та забезпечувати успішне виконання проєктів будь-якої складності [6].

1.4.3 Primavera P6

Primavera P6 – це ІС, спеціально призначена для управління проєктами та портфелями проєктів, яка розроблена корпорацією Oracle. Ця ІС відома своєю високою функціональністю та придатністю для великих та складних проєктів, особливо в галузі будівництва, інфраструктурних робіт, нафти та газу, досліджень та розробки, де керування ресурсами, витратами та часом є критичними факторами успіху.

Primavera P6 надає засоби для створення та управління докладними розкладами проєктів, що включають в себе визначення завдань, ресурсів, їх розподіл, установлення залежностей між завданнями, а також врахування обмежень та ризиків. Ця ІС дозволяє розробникам проєкту побудувати високодеталізовані графіки виконання за допомогою методологій, таких як метод Критичного Шляху (Critical Path Method, СРМ), і забезпечує можливість контролювати процес виконання завдань у режимі реального часу.

Однією з ключових переваг Primavera P6 є його здатність до оптимізації використання ресурсів та витрат, а також визначення критичних шляхів і резервних можливостей в графіку проєкту. Ця система дозволяє встановлювати пріоритети, проводити аналіз впливу змін та забезпечувати максимальний контроль над виконанням проєкту.

Primavera P6 також надає можливості для створення та спільної роботи над проєктами в режимі онлайн, що дозволяє командам з різних регіонів спільно працювати над проєктами, обмінюватися інформацією та вносити зміни у реальному часі. Ця можливість сприяє підвищенню ефективності комунікації та співпраці всередині команди проєкту.

Primavera P6 також включає в себе засоби для аналізу та моделювання різних сценаріїв проекту, що дозволяє розробникам проекту визначити оптимальний план дій та приймати обґрунтовані рішення щодо ресурсів, термінів та витрат.

У великих корпораціях Primavera P6 може бути інтегрована з іншими корпоративними системами для забезпечення обміну даними та синхронізації інформації між різними додатками та платформами.

Загалом, Primavera P6 є потужною ІС для управління проектами, яка допомагає організаціям ефективно планувати, виконувати та контролювати проекти будь-якої складності, забезпечуючи надійний інструментарій для досягнення цілей проекту [7].

1.4.4 Smartsheet

Smartsheet – це ІС, розроблена для управління проектами та завданнями, яка поєднує у собі можливості електронної таблиці та спеціалізованого програмного забезпечення для планування та спільної роботи над проектами. Ця ІС надає зручні та ефективні інструменти для створення, оновлення та відстеження розкладів проектів, завдань та ресурсів, що дозволяє командам працювати спільно та в режимі реального часу.

Smartsheet забезпечує можливість створювати докладні графіки виконання завдань, які можуть бути представлені у вигляді гантівських діаграм або звичайних таблиць, що полегшує відстеження ходу проекту та призначення завдань. Користувачі можуть встановлювати залежності між завданнями, призначати відповідальних осіб та спостерігати за їх виконанням.

Однією з ключових переваг Smartsheet є можливість спільної роботи над проектами, де кожен член команди може вносити зміни та оновлення в

реальному часі. Це сприяє підвищенню комунікації та співпраці всередині команди та сприяє вирішенню завдань швидше та ефективніше.

Smartsheet також надає можливість створення спеціалізованих звітів та аналітики проекту, що допомагає командам отримувати обґрунтовану інформацію та визначати стан проекту. Ця система дозволяє налаштовувати та автоматизувати процеси сповіщень та нагадувань, що сприяє вчасному виконанню завдань та важливих строків.

Smartsheet також може інтегруватися з іншими ІС, такими як Microsoft Office, Google Apps, і іншими, що робить можливим обмін даними та інформацією між різними додатками та платформами.

У великих організаціях Smartsheet може використовуватися для управління декількома проектами та портфелями проектів, дозволяючи керівникам проектів отримувати загальний огляд та контролювати виконання різних проектів в одній системі.

В загальному плані Smartsheet – це потужний інструмент для управління проектами та завданнями, який спрощує процес планування, спільної роботи та контролю над проектами, що дозволяє організаціям ефективно досягати своїх цілей та управляти ресурсами[8].

1.4.5 Trello

Trello – це ІС, призначена для управління завданнями, проектами та спільною роботою команд, яка заснована на концепції «канбан-дошки». Ця ІС розроблена командою Fog Creek Software та пізніше була розвинута та популяризована серед користувачів з усього світу. Trello відома своєю простотою використання та здатністю ефективно організувати завдання та проекти.

Основним елементом Trello є дошка, яка візуально поділяється на колонки, що представляють різні етапи завдань або проєкту. Завдання або картки розташовуються на цих колонках і можуть бути пересувані між ними для відстеження статусу та прогресу. Кожну картку можна розширити, додати до неї докладну інформацію, прикріпити файли, встановити дедлайни та призначити відповідальних осіб.

Однією з ключових переваг Trello є зручність спільної роботи. Кілька користувачів можуть спільно працювати над однією дошкою, додавати та оновлювати завдання, коментувати картки та спілкуватися у реальному часі. Це сприяє ефективній комунікації та співпраці в команді.

Trello надає можливість налаштовувати завдання та додавати до них різні мітки, кольорові позначки та теги, що полегшує організацію та пошук завдань. Крім того, користувачі можуть створювати окремі дошки для різних проєктів, завдань або команд, що робить систему більш універсальною для різних цілей.

Trello також підтримує інтеграцію з іншими ІС, такими як Google Drive, Dropbox, Slack, та багатьма іншими, що дозволяє обмінюватися даними та інформацією між різними додатками та сервісами. Ця можливість робить Trello більш гнучким та зручним інструментом для керування завданнями та проєктами.

У великих організаціях Trello може використовуватися для управління багатьма проєктами та завданнями, спрощуючи організацію та координацію роботи між різними командами та відділами. Загалом, Trello є потужним інструментом для організації роботи, що сприяє зручному управлінню завданнями та проєктами[9].

1.4.6 Asana

Asana – це ІС, призначена для управління завданнями, проектами та робочими процесами в організаціях. Розроблена командою засновників Facebook та багатьма іншими компаніями в 2008 році Asana стала однією з найпопулярніших інструментів для спільної роботи та керування проектами у різних галузях бізнесу.

Основним компонентом Asana є централізована дошка, на якій користувачі можуть створювати завдання, пріоритизувати їх, призначати відповідальних осіб та встановлювати терміни виконання. Кожне завдання може містити детальний опис, прикріплені файли та коментарі, що полегшує розуміння вимог та дій, необхідних для виконання завдання.

Однією з ключових особливостей Asana є здатність створювати проекти та завдання індивідуально, а також групувати їх в проектні комірки та завдання більшого масштабу. Користувачі можуть визначати залежності між завданнями, встановлювати пріоритети та визначати послідовність виконання.

Asana надає можливість відстежувати прогрес виконання завдань та проектів у режимі реального часу. Користувачі можуть створювати звіти, графіки виконання та аналізувати дані, що допомагає керівникам та командам приймати обґрунтовані рішення та вносити корективи в планування.

Asana також сприяє ефективній комунікації та спільній роботі команд. Користувачі можуть обговорювати завдання, ділитися коментарями, спільно редагувати завдання та спілкуватися через інтегрований чат та систему сповіщень.

У великих організаціях Asana може використовуватися для керування багатьма проектами та завданнями, а також для організації спільної роботи між різними відділами та командами. Загалом, Asana є потужним інструментом для управління завданнями та проектами, що сприяє підвищенню ефективності та організації робочих процесів[10].

1.4.7 Atlassian JIRA

JIRA – це ІС, розроблена компанією Atlassian, яка призначена для керування проєктами та завданнями, а також для ведення звітності та управління робочими процесами. JIRA є однією з найпопулярніших систем управління завданнями та проєктами серед розробників програмного забезпечення, а також у багатьох інших галузях.

Основними складовими JIRA є задачі, які можуть бути створені, відстежувані та оновлювані користувачами. Кожна задача має назву, опис, статус, призначених виконавців та інші атрибути. Задачі групуються в проєкти, що допомагає структурувати роботу та відстежувати прогрес по кожному проєкту окремо.

Однією з ключових особливостей JIRA є можливість налаштування робочих процесів та створення власних завдань та проєктних шаблонів. Користувачі можуть визначати правила та етапи виконання завдань, а також автоматизувати багато рутинних операцій.

JIRA надає можливість вести звітність та аналізувати дані про проєкти та завдання. Користувачі можуть створювати звіти, графіки та діаграми для відображення статистики та стану проєктів, що допомагає керівникам приймати обґрунтовані рішення та визначати пріоритети.

JIRA також інтегрується з багатьма іншими інструментами розробки програмного забезпечення, такими як GitHub, Bitbucket, Confluence та інші. Ця можливість спрощує процес розробки та спільної роботи команд, дозволяючи автоматично відстежувати зміни в коді, зв'язувати їх з завданнями та вести звітність.

У великих організаціях JIRA може використовуватися для керування багатьма проєктами та великими командами розробників. Загалом, JIRA є потужним інструментом для управління завданнями та проєктами, що сприяє організації та координації робочих процесів у різних сферах діяльності[11].

1.4.8 Basecamp

Basecamp – це ІС, розроблена компанією Basecamp, LLC, призначена для організації та управління проектами, спільною роботою та комунікацією в командах та організаціях. Заснована у 2004 році, Basecamp відома своєю простотою використання та наголошує на принципах ефективної спільної роботи та комунікації.

Основною одиницею організації в Basecamp є проєкт. В кожному проєкті користувачі можуть створювати завдання, спільно працювати над ними, додавати коментарі, призначати відповідальних осіб та встановлювати терміни виконання. Кожен проєкт також має електронну дошку, на якій можна організовувати та відстежувати завдання та їх прогрес.

Однією з ключових особливостей Basecamp є засіб спільної роботи, який дозволяє користувачам обмінюватися повідомленнями, файли та документами у зручний та структурований спосіб. Ця можливість сприяє покращенню комунікації та співпраці всередині команди.

Basecamp надає можливість планування подій та зустрічей, ведення списків завдань, встановлення завдань та завдань шляхом перетягування та оптимізації робочих процесів. Користувачі також можуть відстежувати зміни та оновлення завдань та сповіщень через електронну пошту.

Однією з інтересних особливостей Basecamp є «Campfire» – внутрішній чат для комунікації в реальному часі, що спрямований на полегшення обговорення та спільної роботи над завданнями та проєктами.

У великих організаціях Basecamp може використовуватися для керування багатьма проєктами та командами, забезпечуючи централізовану платформу для організації та координації роботи. Загалом, Basecamp є ефективним інструментом для організації спільної роботи, керування проєктами та полегшення комунікації в організаціях різних масштабів та галузях[12].

1.5 Основні хмарні провайдери

1.5.1 Порівняльна характеристика хмарних провайдерів

Основними хмарними провайдерами сьогодні є Google Cloud Platform, Amazon Web Services AWS, IBM Cloud, Microsoft Azure, Oracle Cloud Infrastructure. Порівняння основних хмарних провайдерів за критеріями наведено у таблиці 1.1.

1.5.2 Google Cloud Platform

Google Cloud Platform (GCP) є передовою хмарною платформою, розробленою компанією Google, яка надає розширену інфраструктуру для будівництва та розгортання сучасних застосунків. Вона пропонує широкий спектр хмарних послуг, включаючи обчислення, сховища даних та мережеві сервіси, для задоволення потреб бізнесу різних масштабів.

Одним з ключових компонентів GCP є інфраструктура обчислення, яка надає велику гнучкість та розширюваність. Вона включає в себе віртуальні машини, контейнери та сервери без обладнання, що дозволяє користувачам легко розгортати та масштабувати обчислювальні ресурси залежно від потреб проекту.

Для забезпечення потреб зберігання даних GCP пропонує різні типи сховищ, включаючи об'єктні, блокові та файлові сховища. Вони забезпечують високий рівень надійності та доступності, дозволяючи користувачам зберігати та керувати своїми даними в безпечному середовищі.

Хмарна платформа GCP також надає розширені мережеві послуги для побудови та керування мережевою інфраструктурою. Вона пропонує віртуальні мережі, балансування навантаження, віртуальні приватні хмари та Virtual

Таблиця 1.1 – Порівняння хмарних провайдерів

Критерій	GCP	AWS	IBM Cloud	Microsoft Azure	OCI
Масштабованість	Висока	Висока	Висока	Висока	Висока
Продуктивність	Висока	Висока	Висока	Висока	Висока
Використання штучного інтелекту	AI/ML, TensorFlow, AutoML	AI/ML, SageMaker	Watson, AI/ML	Azure AI, Azure Machine Learning	OCI AI Services
Зрозумілість інтерфейсу	Інтуїтивний інтерфейс, добра документація	Різноманітність інтерфейсів, може вимагати більше часу для навчання	Інтуїтивний інтерфейс, добра документація	Інтуїтивний інтерфейс, добра документація	Різноманітність інтерфейсів, може вимагати більше часу для навчання
Політика ціноутворення	Гнучкість, можливість обрання оптимальних опцій	Різноманітність, можливість обрання оптимальних опцій	Різноманітність, можливість обрання оптимальних опцій	Гнучкість, можливість обрання оптимальних опцій	Різноманітність, можливість обрання оптимальних опцій
Рівень безпеки	Високий	Високий	Високий	Високий	Високий

Private Network (VPN), що дозволяє користувачам налаштувати безпеку та забезпечувати ефективну комунікацію між компонентами системи.

Безпека є однією з основних пріоритетів GCP. Платформа надає різноманітні заходи безпеки, такі як керування доступом, шифрування даних, мережеві файрволи та системи моніторингу, що допомагають захистити дані та ресурси від зовнішніх загроз.

Крім того, GCP підтримує розвиток аналітики та штучного інтелекту. Вона пропонує інструменти для аналізу даних, машинного навчання та глибокого навчання, дозволяючи користувачам отримувати цінні інсайти та застосовувати розумові алгоритми для розв'язання складних завдань.

GCP також надає зручні інструменти для управління хмарною інфраструктурою, включаючи консоль керування, командний рядок та Application Programming Interface (API), що дозволяють легко адмініструвати та контролювати ресурси та послуги.

Завдяки своїй географічній доступності GCP має регіональні центри даних по всьому світу, що дозволяє користувачам вибирати найближчий регіон для досягнення оптимальної продуктивності та зниження затримок при обміні даними.

GCP є високотехнологічною хмарною платформою, яка надає розширені можливості для розгортання, масштабування та керування застосунками в хмарі. Ця платформа демонструє високу надійність, продуктивність і безпеку, що робить її перспективним вибором для організацій будь-якого розміру та галузі[13].

1.5.3 Amazon Web Services

Amazon Web Services (AWS) є хмарною платформою, розробленою компанією Amazon, яка надає розширену інфраструктуру для будівництва та

розгортання сучасних застосунків. Вона пропонує широкий спектр хмарних послуг, включаючи обчислення, сховища даних, мережеві сервіси та інструменти штучного інтелекту, для задоволення потреб бізнесу різних масштабів.

Ядром інфраструктури обчислення AWS є Elastic Compute Cloud, яке забезпечує маштабовані віртуальні сервери залежно від потреб користувача. EC2 надає гнучкість вибору конфігурації, операційної системи та доступу до різних типів обчислювальних ресурсів, що дозволяє швидко розгортати та маштабувати застосунки.

AWS пропонує різні типи сховищ даних, включаючи Amazon Simple Storage Service, Elastic Block Store та Amazon Elastic File System. Ці сховища забезпечують надійне та безпечне зберігання даних з можливістю резервного копіювання, шифрування та реплікації.

У сфері мережевих послуг AWS пропонує Amazon Virtual Private Cloud, що дозволяє налаштувати ізольовану віртуальну мережу з контролем доступу та безпекою. Вона також надає мережеві файрволи, балансувальники навантаження та інші інструменти для ефективного керування мережевою інфраструктурою.

AWS має широкий набір послуг для аналітики, включаючи Amazon Athena, Amazon Redshift та Amazon QuickSight, які дозволяють користувачам аналізувати та отримувати цінні інсайти зі своїх даних. Крім того, AWS пропонує послуги штучного інтелекту, такі як Amazon Rekognition та Amazon SageMaker, для розробки та виконання розумових алгоритмів.

AWS має високий рівень безпеки, забезпечуючи різні заходи, включаючи керування доступом, шифрування даних, аудит та моніторинг, що допомагають захистити дані та ресурси від зовнішніх загроз.

Ця хмарна платформа також надає потужні інструменти для управління і моніторингу ресурсів, включаючи Amazon CloudWatch, AWS CloudTrail та AWS Management Console. Вони дозволяють користувачам ефективно керувати

своїми хмарними ресурсами та отримувати детальну інформацію про їх стан та використання.

AWS є однією з найбільших та найпопулярніших хмарних платформ у світі, завдяки своїм передовим технологіям, надійності та гнучкості. Ця платформа дозволяє бізнесам будь-якого розміру розгорнути та керувати своїми застосунками в хмарі з високим рівнем продуктивності та безпеки [14].

1.5.4 Microsoft Azure

Microsoft Azure – це хмарний сервіс, розроблений компанією Microsoft, який надає широкий спектр послуг обчислення, сховища даних, аналітики, штучного інтелекту та інших рішень для розробки, розгортання та керування застосунками у хмарному середовищі.

Azure пропонує широкий набір інструментів та сервісів, які допомагають організаціям будувати та керувати різноманітними застосунками та інфраструктурою.

Azure дозволяє розгорнути віртуальні машини з різними операційними системами, що дозволяє розробникам використовувати власні стеки технологій. Azure App Service – платформа для розгортання та керування веб-застосунками, мобільними застосунками та службами швидкого розгортання. Azure Storage – пропонує різні типи сховищ даних, включаючи об'єктне сховище, файлові сховища, Binary Large Object сховища та таблиці. Azure SQL Database – керована реляційна база даних, яка працює на хмарних серверах. Azure Cognitive Services – набір сервісів штучного інтелекту, таких як впізнавання образів, обробка мови, машинне навчання та інші. Azure Functions – сервіс безсерверного (serverless) обчислення, який дозволяє виконувати код на вимогу, без необхідності керування інфраструктурою. Azure DevOps – набір інструментів для розробки, тестування, розгортання та керування циклом

розробки програмного забезпечення. Azure також пропонує послуги мережі, безпеки, аналітики, Інтернету речей та багато інших інструментів та послуг для розробки та управління застосунками в хмарному середовищі.

Microsoft Azure має широку клієнтську базу та вважається одним з провідних хмарних сервісів на ринку, забезпечуючи високу масштабованість, надійність та безпеку [15].

1.5.5 IBM Cloud

IBM Cloud – це хмарний сервіс, розроблений компанією IBM, який надає широкий спектр послуг обчислення, сховища даних, штучного інтелекту, аналітики, блокчейну та інших рішень для розробки, розгортання та керування застосунками у хмарному середовищі.

IBM Cloud пропонує різноманітні сервіси та інструменти, які допомагають організаціям будувати, розгортати та керувати застосунками в хмарному середовищі. IBM Cloud дозволяють розгортати віртуальні сервери з різними конфігураціями та операційними системами. IBM Watson надає набір інструментів штучного інтелекту для розпізнавання образів, обробки мови, розуміння природної мови та інші. IBM Cloud Object Storage – об'єктне сховище даних, яке дозволяє зберігати та отримувати доступ до великих обсягів даних з високою масштабованістю та безпекою. IBM Cloud Databases – набір керованих баз даних, включаючи реляційні бази даних, NoSQL-бази даних та бази даних часових рядів. IBM Cloud Kubernetes Service – сервіс керованого контейнеризованого розгортання з використанням Kubernetes. IBM Blockchain Platform – платформа для розробки та використання блокчейн-рішень з підтримкою Hyperledger Fabric та інших блокчейн-протоколів. IBM Cloud Functions – це безсерверний сервіс, який дозволяє запускати функції на вимогу, реагуючи на події та запити.

IBM Cloud також пропонує послуги безпеки, мережі, аналітики, розробки та багато інших інструментів для розробки та управління застосунками у хмарному середовищі [16].

1.5.6 Oracle Cloud Infrastructure

Oracle Cloud Infrastructure (OCI) є передовою хмарною платформою, розробленою компанією Oracle, яка надає розширену інфраструктуру для будівництва та розгортання сучасних застосунків. Вона пропонує широкий спектр хмарних послуг, включаючи обчислення, сховища даних та мережеві сервіси, для задоволення потреб бізнесу різних масштабів.

Інфраструктура обчислення OCI забезпечує гнучкість та розширюваність. Вона надає віртуальні машини (VM), контейнери, сервери без обладнання (Bare Metal) та графічні обчислення, що дозволяє розгортати та масштабувати обчислювальні ресурси залежно від потреб користувача.

Для забезпечення потреб зберігання даних OCI надає різні типи сховищ, включаючи блокові, файлові та об'єктні сховища. Вони дозволяють зберігати, керувати та реплікувати дані в безпечному та надійному середовищі з високою доступністю.

Хмарна платформа OCI також пропонує потужні мережеві послуги для побудови та керування мережевою інфраструктурою. Вона забезпечує можливості віртуальних мереж, балансування навантаження, віртуальних приватних мереж та VPN, що дозволяє забезпечити безпеку та ефективну комунікацію між компонентами системи.

Однією з ключових особливостей OCI є її висока ступінь безпеки. Вона надає різноманітні заходи безпеки, такі як керування доступом, шифрування даних, мережеві файрволи та моніторинг, що допомагають захистити дані та ресурси від зовнішніх загроз.

Платформа ОСІ також підтримує розвиток аналітики та штучного інтелекту. Вона надає інструменти для аналізу даних, машинного навчання та глибокого навчання, дозволяючи користувачам отримувати цінні інсайти зі своїх даних та застосовувати розумові алгоритми для розв'язання складних завдань.

Управління хмарною інфраструктурою ОСІ спрощено завдяки набору інструментів для керування, таких як консоль керування та API, що дозволяють легко адмініструвати та контролювати ресурси.

Завдяки своїй географічній доступності ОСІ присутня в різних регіонах по всьому світу. Це дозволяє користувачам вибрати найближчий регіон для забезпечення оптимальної продуктивності та зниження затримок при обміні даними[17].

1.5.7 Обґрунтування вибору хмарного провайдера Google Cloud Platform

GCP пропонує кілька переваг, що роблять його привабливим вибором порівняно з його конкурентами, такими як AWS, IBM Cloud, Microsoft Azure та ОСІ саме для міграції ІС.

GCP має потужну, тобто широку за функціоналом та можливостями інтеграції, глобальну інфраструктуру з розташованими по всьому світу центрами даних. Це дозволяє користувачам розгорнути свої застосунки в близьких до них регіонах, забезпечуючи швидку передачу даних та зниження затримок.

GCP володіє потужними інструментами та сервісами для машинного навчання, розподіленого обчислення та аналізу даних. Наприклад, TensorFlow – популярний фреймворк для розробки моделей глибокого навчання – розроблено саме в Google і інтегровано з GCP сервісами. У багатьох хмарних сервісах Google Cloud наявний високий рівень масштабованості. Це означає, що

вони можуть легко розширювати ресурси горизонтально чи вертикально, в залежності від потреб проєкту.

GCP пропонує різні моделі ціноутворення, що дозволяють користувачам вибрати оптимальний варіант для своїх потреб. Наприклад, можна використовувати плату за використання (pay-as-you-go), зарезервовані інстанси або різні пакети, що дозволяють економити витрати.

GCP приділяє велику увагу безпеці своїх послуг. Він пропонує інструменти для керування доступом, шифрування даних, аудиту та моніторингу, що допомагають захистити дані та ресурси користувачів від зовнішніх загроз.

GCP активно працює над зменшенням впливу своїх центрів даних на навколишнє середовище. Вони забезпечують ефективне використання енергії, використовують відновлювані джерела енергії.

В цілому, GCP пропонує високий рівень технологій, гнучкість, безпеку та інноваційність, що робить його привабливим вибором для бізнесів та розробників, які шукають надійну та ефективну хмарну платформу для своїх проєктів. Саме цей фактор став ключовим для вибору даної платформи при у якості постійного хостингу при міграції ІС з on-premises до хмарної платформи.

1.6 Аналіз особливостей міграції ІС до хмарного сервісу Google Cloud Platform

1.6.1 Способи міграції ІС до хмарного сервісу GCP

Існує декілька способів міграції ІС до хмарних сервісів. Є можливість мігрувати як окремі компоненти ІС, так і всі її компоненти в цілому. Вибір способу міграції залежить від вимог та можливостей замовника.

Для повноцінної міграції ІС до GCP необхідно вирішити наступний перелік питань:

- визначитися з вибором сховища даних;
- визначитися з вибором обчислювальної платформи;
- визначитися з організацією комунікації між застосунками.

Сучасні застосунки можуть мати кілька сховищ даних, якщо вони розробляються на базі мікросервісної архітектури. Тобто доцільним є говорити про реалізацію паттерну «Database per service»).

У хмарному сервісі GCP є можливість розгортання написаного коду «as is» або ж з використанням засобів контейнеризації. Вибір того чи іншого способу вже залежить від конкретних вимог до ІС в цілому.

Якщо необхідно організувати взаємодію одних застосунків з іншими, це можна зробити синхронно або асинхронно. Для обох способів комунікації GCP надає перелік інструментальних засобів для побудови відповідних типів комунікацій.

1.6.2 Вибір сховища даних

Google Cloud BigQuery є хмарною аналітичною платформою, яка відкриває нові можливості для міграції ІС та вирішення різноманітних завдань аналізу та обробки даних. Цей інноваційний інструмент від Google Cloud надає науковий підхід до вирішення таких завдань, і ось деякі з контекстуальних аспектів та сфер використання Google Cloud BigQuery в рамках міграції ІС:

- аналіз великих обсягів даних;
- міграція та консолідація даних;
- аналітика в реальному часі;
- аналіз геопросторових даних;
- аналіз користувацької поведінки та персоналізація;
- безпека та доступність;
- інтеграція з іншими послугами Google Cloud.

Google Cloud BigQuery наділяє організації можливістю виконувати складні аналітичні операції над великими обсягами структурованих та навіть неструктурованих даних. Це сприяє отриманню важливих інсайтів та підтримує прийняття стратегічних рішень.

Google Cloud BigQuery може служити інструментом для міграції даних з різних джерел, незалежно від формату чи обсягу. Це спрощує завдання інтеграції та консолідації даних під час міграції ІС, що дозволяє забезпечити єдність даних.

За допомогою потокового завантаження даних Google Cloud BigQuery може надавати реальний час аналізу даних. Це корисно для систем моніторингу та реагування, таких як системи контролю за кібербезпекою або моніторингу виробництва.

Враховуючи підтримку геопросторових даних, Google Cloud BigQuery підходить для аналізу географічної інформації, що є важливим в контексті міграції географічно розподілених систем.

Google Cloud BigQuery може допомагати аналізувати дані про користувачів, їх взаємодію та представлення персоналізованого контенту або послуг, що підвищує задоволеність користувачів та прибутковість.

Платформа надає рівень безпеки, який відповідає високим стандартам, а також гарантує доступність даних та послуг. Це особливо важливо в контексті міграції систем, де забезпечення безпеки та доступності є критичним завданням.

Google Cloud BigQuery має широкий спектр можливостей для інтеграції з іншими сервісами Google Cloud, такими як Google Cloud Storage, Google Data Studio та інші. Це дозволяє створювати комплексні та масштабовані рішення.

Google Cloud BigQuery є стратегічним інструментом для впровадження аналітики великих даних, міграції даних та побудови ІС з високою продуктивністю та надійністю. Ця платформа відкриває перед сучасними організаціями можливості для збільшення ефективності та прийняття обґрунтованих стратегічних рішень на основі даних [18].

Google Cloud Firestore, як мінімалістична, хмарна, документ-орієнтована система керування базами даних, пропонує значний потенціал для використання в контексті міграції ІС. Вона вирізняється високою гнучкістю та швидкістю доступу до даних, а також можливістю взаємодії в режимі реального часу, що робить її привабливим вибором при міграції ІС. Серед наявного функціоналу:

- зберігання та синхронізація даних користувачів;
- аналітика та обробка подій у реальному часі;
- міграція та інтеграція даних;
- масштабування та розподілена обробка;
- інтеграція з іншими сервісами Google Cloud.

Firestore є ідеальним рішенням для зберігання та синхронізації даних користувачів у мобільних додатках та веб-сервісах. Забезпечуючи можливість зберігання структурованих даних у документах та миттєву доставку оновлень, вона забезпечує споживачів актуальною інформацією.

Firestore дозволяє обробляти потокові події та миттєво реагувати на них. Це корисно для створення систем аналізу реального часу, таких як моніторинг додатків, системи тревов та прогнозування на основі реальних даних.

Firestore відкриває можливість для міграції даних з різних джерел та їх інтеграції у єдину базу даних. Це спрощує процес об'єднання інформації з різних джерел та створення єдиної точки доступу до неї.

Firestore пропонує розподілену архітектуру, що сприяє масштабуванню та обробці завдань з великими обсягами даних. Це важливо для забезпечення високої доступності та продуктивності під час міграції ІС.

Firestore інтегрується з іншими сервісами Google Cloud, такими як Google Cloud Functions, що дозволяє автоматизувати обробку подій та реагувати на них. Це дозволяє створювати комплексні рішення для обробки даних.

Google Cloud Firestore, завдяки своїй гнучкості, можливостям реального часу та інтеграції з екосистемою Google Cloud, може служити ефективним

інструментом для міграції ІС та розв'язання різноманітних задач зберігання, обробки та аналізу даних у великих масштабах[19].

Google Cloud Storage – це хмарний сервіс для зберігання та керування об'єктами даних в Google Cloud Platform, яка відкриває безліч можливостей для вирішення різних завдань зберігання та обробки даних [20]. З використанням наукового стилю, розглянемо сфери застосування Google Cloud Storage:

- зберігання даних великого обсягу;
- архівація та резервне копіювання;
- спільний доступ до об'єктів;
- аналіз даних та обробка;
- міграція даних;
- інтеграція з іншими хмарними послугами;
- географічно розподілені додатки.

Google Cloud Storage відмінно підходить для зберігання великих обсягів даних, таких як мультимедійні файли, резервні копії, журнали тощо. Це дозволяє організаціям зберігати дані без обмежень обсягу на інфраструктурі хмари.

Google Cloud Storage може служити засобом для зберігання архівних даних та резервних копій, забезпечуючи надійне зберігання та доступ до них у випадку втрати основних даних.

Спільний доступ дозволяє організаціям спільно працювати над об'єктами даних, забезпечуючи доступ для декількох користувачів або систем у безпечний спосіб.

Google Cloud Storage може використовуватися для міграції даних з інших джерел, зокрема із власних центрів обробки даних в хмару Google.

Завдяки можливості зберігання структурованих та неструктурованих даних, Google Cloud Storage є ідеальним для обробки та аналізу великих обсягів даних за допомогою інших інструментів, таких як Google BigQuery чи Apache Spark.

Google Cloud Storage легко інтегрується з іншими сервісами Google Cloud, такими як Google Cloud Dataflow, Google Cloud Functions і Google Cloud Machine Learning, що дозволяє створювати комплексні рішення для обробки та аналізу даних.

Для додатків, що працюють у різних регіонах, Google Cloud Storage надає можливість зберігати дані в різних місцях для оптимальної продуктивності та доступності.

Google Cloud Storage є важливим елементом екосистеми Google Cloud та служить для різноманітних задач зберігання та обробки даних у хмарному середовищі. Вона надає надійну та ефективну інфраструктуру для розв'язання сучасних завдань зберігання та управління об'єктами даних.

1.6.3 Вибір обчислювальної платформи

Google App Engine (GAE) представляє собою значущу платформу для розгортання та управління програмними застосунками у хмарному середовищі GCP. Сервіс надає високорівневий підхід до розгортання програмного коду, де інфраструктурні аспекти обслуговуються автоматично, надаючи розробникам можливість зосередитися на написанні програмного коду та функціональних аспектах своїх застосунків.

GAE надає абстракцію від інфраструктури, включаючи операційну систему та серверне обладнання. Розробники можуть завантажити свій код та налаштування, а сервіс автоматично керує розміщенням та масштабуванням застосунку.

GAE підтримує різні мови програмування, включаючи Python, Java, Node.js, Go та PHP, що дає можливість розробникам використовувати мову, з якою вони найбільш зручні.

Під час збільшення навантаження GAE автоматично розгортає додаткові інстанції застосунку, забезпечуючи стійкість та відмовостійкість.

GAE інтегрується з Google Identity Platform, що дозволяє використовувати механізми аутентифікації та авторизації, що відповідають стандартам безпеки.

Застосунки, розгорнуті на GAE, виконуються на розподілених серверах Google у світовому масштабі, що гарантує високу доступність і низький рівень відмов.

GAE дозволяє швидко розгорнути застосунки без необхідності управління віртуальними машинами або інфраструктурними деталями.

Застосунки, розгорнуті на GAE, легко інтегруються з іншими сервісами GCP, такими як бази даних, хмарне сховище, моніторинг і багато інших, для розширення функціональності.

GAE надає зручну та високорівневу платформу для розгортання та керування веб-застосунками без необхідності управління інфраструктурою. Це спрощує процес розробки, розгортання та масштабування застосунків, дозволяючи розробникам фокусуватися на розробці функціональності своїх застосунків[21].

Google Kubernetes Engine (GKE) є важливою частиною інфраструктури хмарного обчислення GCP, яка надає масштабовану та керовану платформу для контейнеризованих застосунків. Цей сервіс базується на відкритому джерелі Kubernetes та доповнює його глибокими інтеграціями з іншими сервісами GCP, створюючи потужне середовище для розгортання, керування та масштабування контейнерів.

Основними аспектами GKE є:

- автоматизоване розгортання;
- оркестрація контейнерів;
- інтеграція з іншими сервісами GCP;
- безпека: GKE надає захист на рівні платформи, включаючи доступ до ідентифікації та контроль доступу;
- масштабованість;

– зручність управління.

GKE дозволяє автоматизувати процес розгортання контейнерів, дозволяючи розробникам концентруватися на розробці застосунків, а не на інфраструктурних завданнях.

Використовуючи Kubernetes, GKE надає потужні засоби для оркестрації та керування контейнерами, включаючи автоматичне масштабування, вирівнювання навантаження та самодостатність.

GKE інтегрується з іншими сервісами GCP, такими як Cloud Storage, Cloud Logging, та Cloud Monitoring, що спрощує розгортання та управління застосунками.

За допомогою GKE можливо легко масштабувати контейнери, щоб забезпечити стійкість та низький рівень відмов.

GKE надає інтерфейси для керування та моніторингу застосунків, що допомагає розробникам та операторам ефективно управляти середовищем.

У висновку, Google Kubernetes Engine є потужною та високорівневою платформою для розгортання та керування контейнеризованими застосунками. Цей сервіс забезпечує автоматизацію, масштабованість та інтеграцію з іншими сервісами GCP, роблячи його ідеальним вибором для оркестрації контейнерів у хмарному середовищі.

Google Compute Engine (GCE) представляє собою важливий складник інфраструктури хмарного обчислення GCP, що надає можливість віртуального розгортання і управління VM. Цей сервіс розширює інфраструктурні можливості користувачів, дозволяючи їм створювати та управляти VM з різними конфігураціями і операційними системами в хмарному середовищі.

GCE надає ряд важливих характеристик. Користувачі отримують повний контроль над віртуальними машинами з можливістю вибору конфігурації та операційної системи. Сервіс пропонує високопродуктивні обчислювальні ресурси, що дозволяє запускати високонавантажені та обчислювально інтенсивні застосунки з мінімальними затримками.

Користувачі можуть легко змінювати розмір та кількість віртуальних машин в залежності від власних потреб. Забезпечення безпеки та контроль доступу здійснюється за допомогою служби Identity and Access Management. GCE інтегрується з іншими хмарними послугами GCP, що дозволяє створювати комплексні рішення для різних завдань.

GCE є потужним сервісом для створення, керування та використання VM у хмарному середовищі. Цей сервіс дозволяє користувачам отримати доступ до обчислювальних ресурсів з високою продуктивністю, гнучкістю та контролем, необхідними для вирішення різних завдань і вимог [22].

Google Cloud Run – передова платформа для розгортання та виконання контейнеризованих застосунків в хмарному середовищі Google Cloud. Цей сервіс надає виняткову можливість створювати, розгортати та керувати контейнерами з програмним кодом безпосередньо з веб-сервером, дозволяючи розробникам швидко та ефективно надавати свої застосунки в інтернеті.

Google Cloud Run дозволяє розробникам розгортати контейнери, створені з використанням Docker, та автоматично керувати їх масштабуванням в залежності від навантаження. Ця платформа надає середовище, де розробники можуть концентруватися на розробці свого програмного коду та бізнес-логіці, не витрачаючи час на управління інфраструктурою.

Google Cloud Run використовує контейнери для забезпечення відокремленості та консистентності робочого середовища застосунків. Ця платформа інтегрується з іншими службами Google Cloud, що сприяє покращенню продуктивності та можливостям застосунків. До її основних переваг належать гнучкість, швидкість розгортання, автоматичне масштабування та простота управління контейнеризованими застосунками.

У підсумку, Google Cloud Run є потужною платформою для розгортання і керування контейнеризованими застосунками, яка відкриває перед розробниками широкі можливості у сфері розробки та розгортання веб-застосунків у хмарному середовищі [23].

Google Cloud Functions – це сучасна інтернет-платформа в рамках Google Cloud, яка надає можливість розгорнути та виконувати індивідуальні функції безпосередньо в хмарному середовищі. Цей сервіс базується на концепції «функціонального обчислення» і дозволяє розробникам виконувати код лише відповідно до певних подій або тригерів без необхідності управління інфраструктурою серверів.

Google Cloud Functions дозволяє розробникам завантажувати код функції та налаштовувати тригери, які викликають цю функцію у відповідь на певні події або дії. Інтеграція з різними сервісами Google Cloud дозволяє реагувати на події, що стаються в цих сервісах, і виконувати відповідні дії.

Функції, створені за допомогою Google Cloud Functions, автоматично масштабуються в залежності від навантаження, забезпечуючи високу доступність та стійкість. Платформа підтримує різні мови програмування, що дає розробникам вибір мови, з якою вони найбільш зручні. Інтеграція з іншими сервісами Google Cloud розширює можливості функцій та дозволяє створювати комплексні рішення.

У підсумку, Google Cloud Functions є потужною інтернет-платформою для розгортання та виконання індивідуальних функцій в хмарному середовищі, що дозволяє розробникам легко створювати, налаштовувати та автоматизувати завдання на основі подій і тригерів [24].

1.6.4 Організація комунікації між застосунками

Синхронна комунікація між застосунками має свої переваги і недоліки, і вибір використання цього підходу повинен базуватися на конкретних вимогах та сценаріях застосування. Переваги синхронної комунікації:

- простота розробки і розуміння;
- миттєва відповідь;

– зручність в налагодженні.

Синхронна комунікація зазвичай є легшою для розробки та розуміння, оскільки вона моделює взаємодію в стилі «запит-відповідь». Це дозволяє розробникам простіше відслідковувати потік даних і вирішувати проблеми в коді.

Синхронна комунікація дозволяє отримувати миттєву відповідь від отримувача. Це корисно в сценаріях, де час очікування відповіді є критичним (наприклад, у веб-застосунках).

У випадку помилок або проблем зі синхронною комунікацією, налагодження коду може бути спрощено через послідовний характер взаємодії.

Недоліки синхронної комунікації:

- блокування та затримки;
- складність масштабування;
- вразливість до відмов;
- обмін великими об'ємами даних.

Однією з головних недоліків синхронної комунікації є можливість блокування, коли одна сторона чекає на відповідь від іншої. Це може призводити до затримок у виконанні і загального зниження продуктивності системи.

Синхронна комунікація може бути складною для масштабування, оскільки потребує обслуговування багатьох одночасних з'єднань і може становити тяжкість при великому обсязі даних або великій кількості користувачів.

Якщо один з компонентів системи відмовляє, то це може призвести до недоступності всієї системи, оскільки інші компоненти, що залежать від нього, будуть заблоковані.

Синхронна комунікація може бути неефективною при обміні великими об'ємами даних, оскільки вона може спричиняти великі навантаження на мережу та ресурси.

Загалом, синхронна комунікація має свої переваги, але також має свої недоліки, і її використання вимагає ретельного аналізу конкретних вимог проекту та збалансованого підходу до розробки системи. У деяких випадках може бути доцільним використання синхронної та асинхронної комунікації в поєднанні для досягнення найкращих результатів.

Синхронна комунікація в GCP використовується для обміну даними між різними компонентами або застосунками у реальному часі, коли відправник та отримувач очікують один на одного для завершення обміну інформацією. Цей підхід дозволяє створювати високопродуктивні, інтерактивні та надійні системи в хмарному середовищі GCP.

Основні складові синхронної комунікації в GCP включають:

- Hypertext Transfer Protocol (HTTP)-запити [25];
- Google Cloud Pub/Sub;
- Google Cloud Storage;
- Google Cloud SQL.

В GCP є можливість використовувати служби, такі як Google Cloud Endpoints або Google App Engine, для створення веб-застосунків, які обмінюються HTTP-запитами. Це дозволяє різним компонентам системи взаємодіяти через стандартний HTTP-протокол.

Це служба для асинхронної комунікації, але використовується для синхронних випадків також. За допомогою Google Cloud Pub/Sub можна створювати теми та підписники для обміну повідомленнями між компонентами системи [26]. Отримувачі можуть підписатися на теми та отримувати повідомлення в режимі реального часу.

Ця служба дозволяє зберігати та обмінюватися файлами між різними частинами системи. GCP можна використовувати для завантаження, зберігання та обміну файлами за допомогою синхронних операцій.

Якщо система потребує роботи з базами даних, Google Cloud SQL надає можливість синхронної комунікації з реляційними базами даних, такими як MySQL, PostgreSQL та інші [27].

Асинхронна комунікація між застосунками також має свої переваги і недоліки, які варто враховувати при розробці і використанні таких систем.

Переваги асинхронної комунікації:

- масштабованість;
- відмінність часу обробки;
- висока доступність;
- обмін складними структурами даних.

Однією з основних переваг асинхронної комунікації є її здатність до масштабування. Системи, що використовують асинхронний обмін повідомленнями, можуть легко розширюватися шляхом додавання нових компонентів або обробників повідомлень.

Асинхронна комунікація дозволяє компонентам системи обробляти повідомлення власним темпом. Це особливо корисно в сценаріях, де обробка повідомлень може займати різну кількість часу.

У випадку відмови одного з компонентів системи інші можуть продовжити обробку повідомлень. Це сприяє високій доступності системи.

Асинхронна комунікація може бути корисною при обміні великими об'ємами даних або складними структурами даних, оскільки вона дозволяє ефективно передавати і обробляти ці дані.

Недоліки асинхронної комунікації:

- складність розробки і налагодження;
- потенційна складність управління порядком повідомлень;
- затримки в обробці;
- підвищена складність у вирішенні проблем з безпекою.

Асинхронна комунікація може бути складнішою для розробки і налагодження порівняно зі схожими синхронними системами. Вона може призводити до більшої складності логіки програми і ускладнювати налагодження.

У системах, де порядок обробки повідомлень є важливим, управління цим порядком може стати складною задачею.

Якщо велика кількість повідомлень накопичується в черзі, це може призвести до затримок в обробці, особливо якщо обробка кожного повідомлення займає багато часу.

У деяких випадках асинхронна комунікація може викликати додаткові проблеми з безпекою, такі як можливість атак на чергу повідомлень.

Загалом, вибір між синхронною і асинхронною комунікацією залежить від конкретних потреб і вимог проекту. Обидва підходи мають свої місце та використовуються відповідно до контексту і завдань системи.

GCP надає ряд служб та інструментів для підтримки асинхронної комунікації між застосунками.

Google Cloud Pub/Sub надає можливість відправляти та отримувати повідомлення асинхронно. Він дозволяє створювати теми та підписники для обміну повідомленнями між різними компонентами системи. Pub/Sub можна використовувати для створення розподілених систем, де компоненти можуть взаємодіяти безпосередньо через повідомлення.

Google Cloud Storage дозволяє зберігати та обмінюватися файлами асинхронно. Наявний функціонал дозволяє завантажувати файли в обліковий запис Cloud Storage та обмінюватися ними між різними компонентами системи.

Google Cloud Functions дозволяє створювати легкі функції, які відповідають на події в системі безпосередньо асинхронно.

Google Cloud Dataflow використовується для обробки даних в реальному часі або пакетному режимі. Він дозволяє створювати потокові та пакетні обробники даних, які можуть взаємодіяти асинхронно з різними джерелами даних і призначеннями [28].

Google Cloud Tasks використовується для створення та обробки завдань асинхронно. Є опція використовувати Cloud Tasks для планування та виконання робіт асинхронно між різними компонентами застосунків [29].

Ці інструменти дозволяють розробникам створювати розподілені системи в GCP, які можуть обмінюватися даними та взаємодіяти асинхронно. Це особливо корисно для розподілених та високонавантажених застосунків, де

асинхронна комунікація може сприяти підвищенню ефективності та масштабованості системи.

1.7 Особливість задачі оцінки тривалості операцій в контексті ІТ-проектів з міграції ІС до GCP

При виконанні ІТ-проекту з міграції ІС до GCP є такі фактори, які не залежать від виконавців проекту та які неможливо врахувати заздалегідь. В своїй більшості вони пов'язані з доступністю та обмеженнями хмарної інфраструктури а також стабільністю її роботи на великих об'ємах виробничих даних. Ці фактори породжують часову невизначеність, яку також необхідно враховувати (наскільки це можливо) при оцінці тривалості виконання операцій ІТ-проекту. Найкраще адаптованим до часової невизначеності є метод PERT, завдяки використанню 3 можливих сценаріїв. В свою чергу, його недоліком в контексті проектів з міграції ІС до GCP є орієнтованість на найімовірніший сценарій, що може не покривати всіх можливих випадків і, як результат, демонструвати помірну точність оцінки (для деяких випадків).

1.8 Формування мети та постановка задачі кваліфікаційної роботи

Об'єктом дослідження в рамках магістерської роботи є оцінка тривалості операцій ІТ-проекту з міграції ІС до GCP.

Предметом дослідження є методи оцінки тривалості операцій ІТ-проекту.

Метою дослідження є підвищення точності оцінки тривалості виконання операцій в контексті ІТ-проєкту з міграції ІС до GCP.

Для досягнення поставленої мети будуть вирішуватися такі завдання дослідження:

- аналіз предметної галузі та сучасного стану проблеми;
- синтез методу, що враховує часову невизначеність та не орієнтований на найімовірніший сценарій;
- опис практичного вирішення задачі оцінки тривалості операцій ІТ-проєкту;
- апробація синтезованого методу та порівняння з методом PERT.

2 РОЗРОБКА МЕТОДУ ПАРАМЕТРИЧНОГО ОЦІНЮВАННЯ НА БАЗІ МЕТОДУ PERT

2.1 Опис методу PERT

2.1.1 Використання методу PERT для оцінки тривалості операцій

Метод PERT застосовується для оцінки тривалості операцій або завдань в проєктах. Основна ідея полягає в тому, що тривалість операції може бути невизначеною і варіюватися в залежності від умов. Для оцінки тривалості операції в методі PERT застосовується формула:

$$T = \frac{a + 4b + c}{6}, \quad (2.1)$$

де T – середньозважений час виконання операції;
 a – оптимістичний час виконання операції;
 b – найімовірніший час виконання операції;
 c – песимістичний час виконання операції.

Метод PERT формула (2.1) враховує оптимістичний, середній і песимістичний сценарії тривалості операції, що дозволяє отримати більш об'єктивну оцінку тривалості проєкту або його частини.

Функція розподілу часу виконання операції має вигляд:

$$f(t) = \frac{(t-a)^{\alpha-1} (c-t)^{\beta-1}}{B(\alpha, \beta)(c-a)^{\alpha+\beta-1}}, \quad (2.2)$$

де $f(t)$ – функція розподілу часу виконання операції;
 t – найімовірніший час виконання операції;
 $B(\alpha, \beta)$ – бета-функція;
 α, β – параметри бета-розподілу;

- a – оптимістичний час виконання операції;
- b – найімовірніший час виконання операції;
- c – песимістичний час виконання операції.

Параметри бета-розподілу α та β розраховуються за формулами:

$$\alpha = \frac{4b + c - 5a}{c - a} = 1 + 4 \frac{b - a}{c - a}, \quad (2.3)$$

$$\beta = \frac{5c - a - 4b}{c - b} = 1 + 4 \frac{c - b}{c - a}, \quad (2.4)$$

- де
- a – оптимістичний час виконання операції;
 - b – найімовірніший час виконання операції;
 - c – песимістичний час виконання операції.

2.1.2 Недоліки методу PERT для оцінки тривалості операцій

Метод PERT є корисним інструментом для оцінки тривалості операцій у проєктах, проте він має свої недоліки та обмеження:

- суб'єктивність експертних оцінок;
- необхідність ретельної експертизи;
- важкість реалізації;
- важкість підтримки;
- орієнтованість на найімовірніший сценарій.

Метод PERT вимагає оцінок від експертів щодо тривалості операцій. Однак такі оцінки є суб'єктивними і залежать від індивідуального досвіду та точки зору кожного експерта. Різні експерти надають різні оцінки для однієї й тієї ж операції, що впливає на результати методу.

При використанні методу PERT важливо провести докладний аналіз і експертизу, щоб визначити оптимальні значення для кожної операції. Експертиза враховує не лише тривалість операцій, але й можливі ризики, вплив зовнішніх факторів та інші аспекти.

Впровадження методу PERT вимагає певних зусиль. Наприклад, необхідно враховувати специфіку проєкту, підготувати персонал до використання методу та забезпечити належний контроль за процесом.

Метод PERT базується на статистичних розрахунках, які орієнтовані на найімовірніший сценарій. Такий спосіб оцінювання має й свої недоліки. По-перше, адекватність оцінки тривалості операції залежить від адекватності формування кожного сценарію. Неадекватні оцінки сприяють неадекватному результату, на який не можна орієнтуватися. Для формування адекватних сценаріїв необхідно витратити додатковий час, якщо порівнювати з методами експертного оцінювання або оцінювання за аналогами, адже тут експертам необхідно врахувати три можливі сценарії розвитку подій (або залучити більше експертів, що призведе до збільшення витрат). По-друге, адекватність оцінки залежить не тільки від адекватності сформованих сценаріїв, а й від коефіцієнтів, які визначають вплив кожного сценарію на кінцевий результат. Тобто, орієнтованість на найімовірніший сценарій з мінімальним урахуванням впливу інших сценаріїв може не відобразити усі можливі випадки при оцінюванні тривалості операцій ІТ -проєктів.

Особливістю ІТ-проєктів з міграції ІС до GCP є залежність від доступності сервісів обраного хмарного провайдера, на що неможливо вплинути зі сторони учасників проєкту. Завжди є актуальним внесення поправки на проблеми з доступністю хмарних сервісів та їх обмеження на реальних (не тестових) об'ємах даних, що призводить до реалізації песимістичного сценарію.

2.2 Розробка методу параметричного оцінювання на базі методу PERT

Для вирішення орієнтованості на найімовірніший сценарій, яка наявна в оригінальному методі PERT, середнє значення часу виконання операції пропонується розраховувати за наступною формулою:

$$\mu = \frac{\chi a + \delta b + \varepsilon c}{\chi + \delta + \varepsilon}, \quad (2.5)$$

де μ – середнє значення часу виконання операції;
 χ – ваговий коефіцієнт оптимістичного сценарію;
 a – час виконання операції за оптимістичним сценарієм;
 δ – ваговий коефіцієнт найімовірнішого сценарію;
 b – час виконання операції за найімовірнішим сценарієм;
 ε – ваговий коефіцієнт песимістичного сценарію;
 c – час виконання операції за песимістичним сценарієм.

У методі параметричного оцінювання на базі методу PERT формула (2.5) можливим сценаріям часу виконання операції a , b та c призначено деякі вагові коефіцієнти χ , δ та ε , що визначаються експертами, виходячи з вимоги $\chi, \delta, \varepsilon \in R$.

Розроблений метод параметричного оцінювання дає змогу визначати вклад кожного сценарію у кінцеву оцінку тривалості операції, використовуючи такі вагові коефіцієнти, які на думку експертів будуть найрелевантнішими в контексті виконання специфічної операції.

Виходячи з формули (2.5), параметри бета-розподілу α та β розраховуються за формулами

$$\alpha = (\chi + \delta + \varepsilon) \frac{\mu - a}{c - a}, \quad (2.6)$$

$$\beta = (\chi + \delta + \varepsilon) \frac{c - \mu}{c - a}, \quad (2.7)$$

де χ – ваговий коефіцієнт оптимістичного сценарію;
 δ – ваговий коефіцієнт найімовірнішого сценарію;
 ε – ваговий коефіцієнт песимістичного сценарію;
 μ – середнє значення часу виконання операції;
 a – час виконання операції за оптимістичним сценарієм;
 c – час виконання операції за песимістичним сценарієм.

2.3 Визначення критеріїв формування коефіцієнтів для методу параметричного оцінювання на базі методу PERT

Задача визначення критеріїв для розробленого методу є унікальною для кожної операції, для якої таке оцінювання буде виконуватися. Критерії та їх оцінки необхідно визначати, базуючись на експертних знаннях для кожної операції індивідуально. Таким чином формується задача оптимізації витрат часу розрахунку критеріїв для можливих сценаріїв у методі параметричного оцінювання на базі методу PERT, для вирішення якої необхідно застосувати розроблений метод на деякому кінцевому переліку ІТ-проектів з міграції ІС до GCP емпірично визначити перелік деяких стандартних операцій, їх тип та сформувати шкалу критеріїв та оцінок, за якими для даних стандартних та подібних їм операцій будуть розраховуватися вагові коефіцієнти сценаріїв.

2.4 Етапи оцінювання тривалості операцій ІТ-проєкту з використанням методу параметричного оцінювання на базі методу PERT

Процес оцінювання тривалості операцій ІТ-проєкту з використанням методу параметричного оцінювання на базі методу PERT:

- визначення експертами оцінок тривалості операцій за оптимістичним, найімовірнішим та песимістичним сценаріями;
- узгодження експертами шкали вагових коефіцієнтів та відповідних їм оцінок для кожного сценарію;
- визначення значень вагових коефіцієнтів χ , δ та ε для кожної операції згідно узгодженої шкали критеріїв;
- розрахунок середнього часу тривалості виконання операції за формулою (2.5).

На етапі визначення тривалості операцій за оптимістичним, найімовірнішим та песимістичним сценаріями в рамках розробленого методу параметричного оцінювання, виконується комплекс дій з формуванням експертних оцінок тривалості операції для відповідного сценарію. Зазначений етап включає активне залучення фахівців та учасників проєкту, обладнаних належним досвідом у виконанні аналогічних завдань.

Узгодження експертами шкали вагових коефіцієнтів для кожного сценарію формує єдину шкалу критеріїв та відповідних їм оцінок індивідуально для кожної окремої операції ІТ-проєкту.

Визначення значень вагових коефіцієнтів χ , δ та ε виконується для кожної операції згідно узгодженої раніше шкали критеріїв.

Розрахунок тривалості операцій виконується згідно формули (2.5) та визначених раніше вагових коефіцієнтів для кожної операції окремо.

2.5 Висновки

Було проаналізовано оригінальний метод PERT та визначено його переваги та недоліки. За результатами аналізу визначено, що використання оригінального методу PERT не є доцільним, коли контекст операції завідомо не спирається на найімовірніший сценарій.

При розробці методу параметричного оцінювання на базі методу PERT було зосереджено увагу на виправленні орієнтованості на найімовірніший сценарій з мінімальним урахуванням впливу оптимістичного та песимістичного сценаріїв на кінцеву оцінку тривалості операції.

Як результат, розроблено метод параметричного оцінювання, що дозволяє ставити вагові коефіцієнти у відповідність сценаріям тривалості операцій. Розрахунок вагових коефіцієнтів є процесом, що потребує додаткового аналізу та індивідуального підходу до контексту кожної операції ІТ-проєкту з міграції ІС до GCP.

3 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОЦІНКИ ТРИВАЛОСТІ ОПЕРАЦІЙ ІТ-ПРОЄКТУ З МІГРАЦІЇ ІС ДО GOOGLE CLOUD PLATFORM

3.1 Опис інформаційної технології

Для часткової автоматизації розрахунків оцінки тривалості операцій за методом параметричного оцінювання на базі методу PERT було розроблено спеціальний веб-сервіс – PERT-based parametric estimation calculator [30].

Розроблений веб-сервіс являє собою першу пілотну версію веб-застосунку, написану мовою програмування Java [31] з використанням компонентів веб-фреймворку Spring [32], таких як Spring MVC [33], Spring Boot [34]. Веб-застосунок спроектований з використанням архітектурного підходу Representational State Transfer (REST) [35]. Веб-сервіс відповідає першому рівню Richardson Maturity Model і є окремим бекенд компонентом, що здатний приймати HTTP запити з даними, необхідними для розрахунків тривалості операцій за модифікованим методом PERT у форматі JavaScript Object Notation (JSON) [36].

Функціонал поточної версії веб-сервісу дозволяє виконувати розрахунки тривалості операцій використовуючи метод параметричного оцінювання на базі методу PERT. Для виконання розрахунків надано можливість робити запити напряму до сервісу або використовувати HTTP клієнт.

Надсилаючи запит напряму до веб-сервісу, необхідно вказати тіло HTTP запиту як масив JSON об'єктів, що містять у собі метайнформацію операції: ідентифікатор, назву та попередні оцінки тривалості операцій за відповідним сценарієм – оптимістичним, песимістичним на найімовірнішим. Кожна така оцінка містить у собі власне попередню оцінку тривалості операцій та її вплив на формування кінцевої оцінки. У тілі запиту можна відправляти від 1 до 1000 операцій включно. Приклад тіла запиту наведено на рисунку 3.1.

```

{
  "operations": [
    {
      "operationId": "string",
      "operationName": "string",
      "estimates": {
        "OPTIMISTIC": {
          "duration": "float",
          "influence": "float"
        },
        "PESSIMISTIC": {
          "duration": "float",
          "influence": "float"
        },
        "EXPECTED": {
          "duration": "float",
          "influence": "float"
        }
      }
    }
  ]
}

```

Рисунок 3.1 – Тіло запиту для PERT-based parametric estimation calculator

Існує можливість надсилати запит за допомогою окремого компоненту – користувацького інтерфейсу. Для цього необхідно відкрити сайт компоненту PERT-based parametric estimation та натиснути кнопку Begin (рисунок. 3.2).

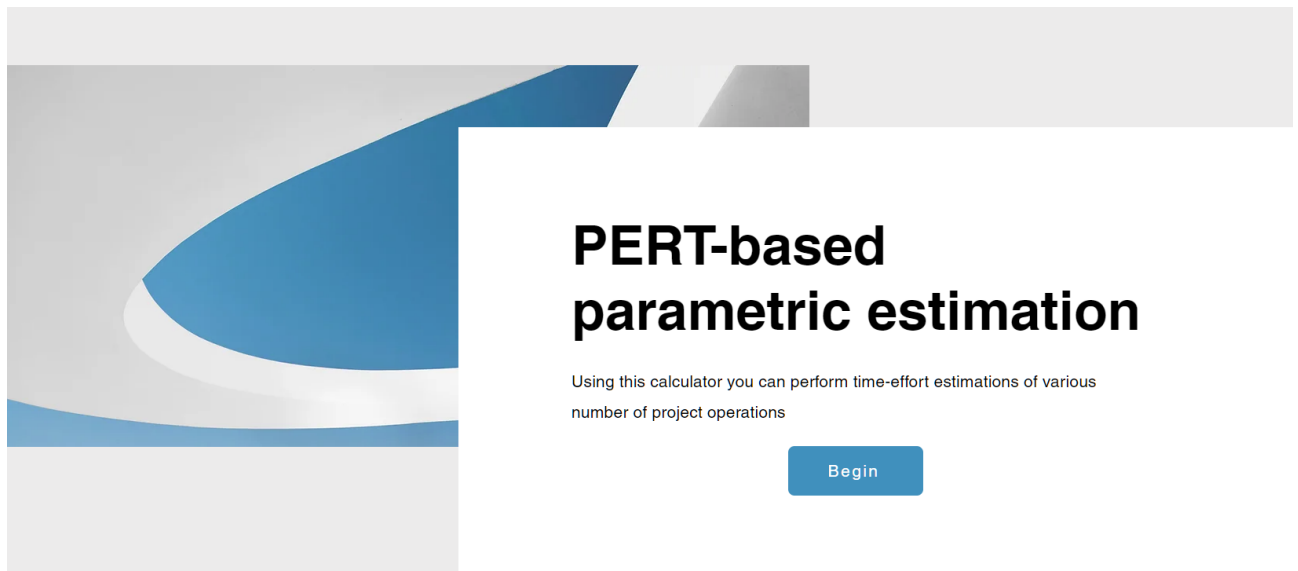
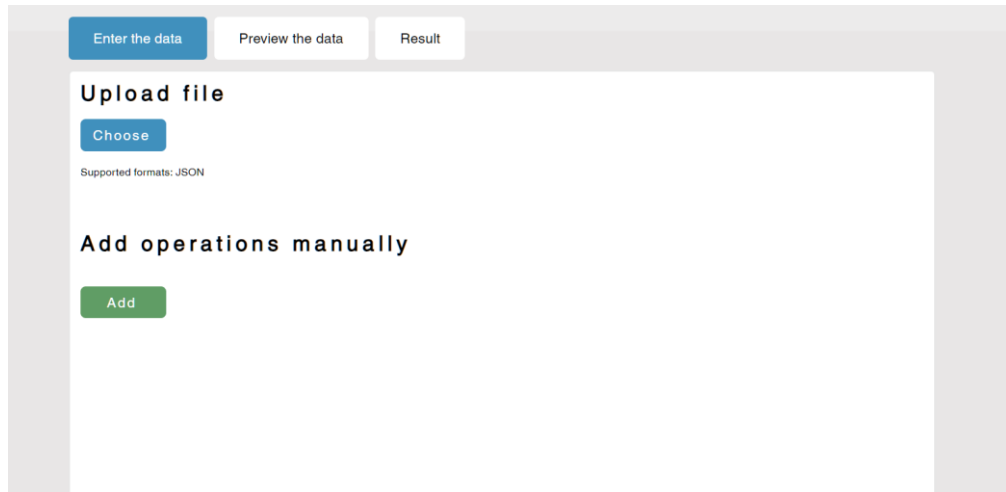


Рисунок 3.2 – Головна сторінка компоненту користувацького інтерфейсу

Натискання кнопки «Begin» користувача буде переадресовано до меню з 3 вкладками: введення даних, перевірки введених даних та розрахунку.

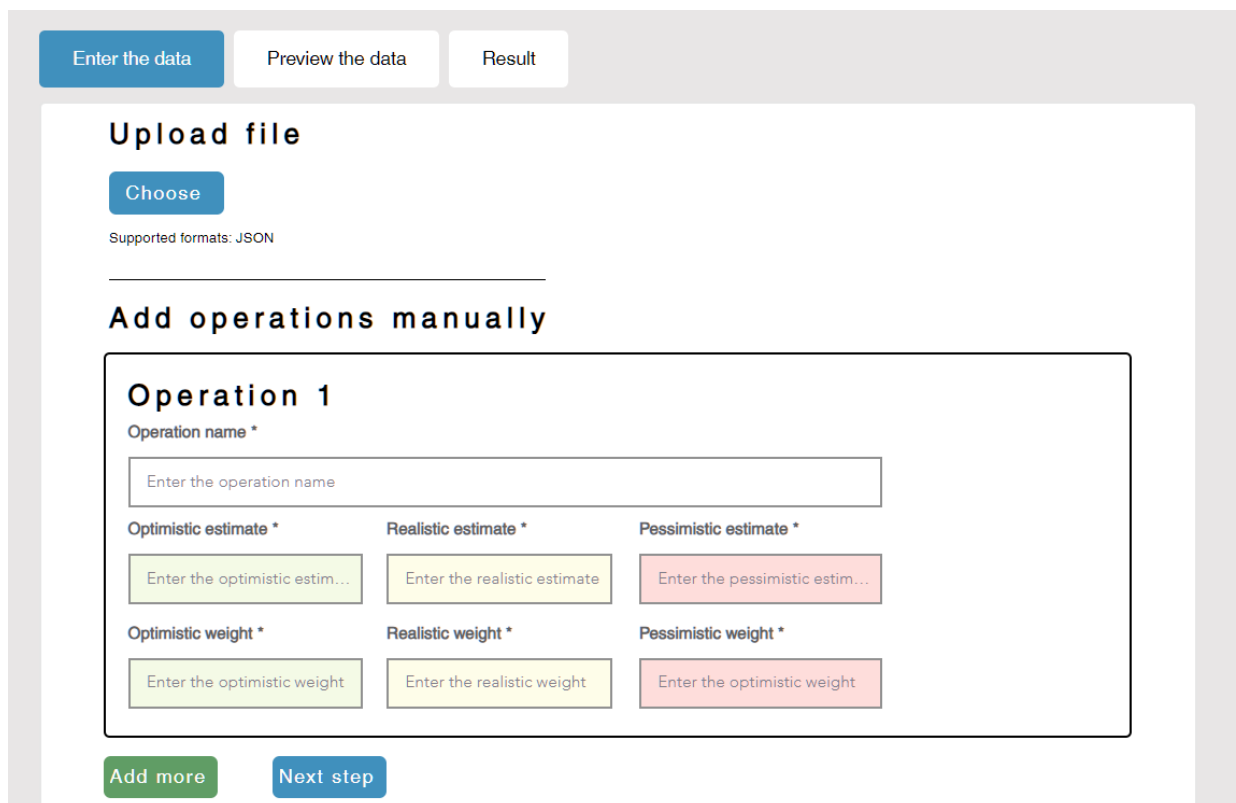
Введення даних вручну зображено на рисунках 3.3-3.5. Спочатку користувач натискає кнопку Add (рисунок 3.3) і отримує можливість ввести інформацію щодо операції



The screenshot shows a web interface with three buttons at the top: 'Enter the data' (blue), 'Preview the data' (white), and 'Result' (white). Below these is a section titled 'Upload file' with a 'Choose' button and the text 'Supported formats: JSON'. Underneath is a section titled 'Add operations manually' with a green 'Add' button.

Рисунок 3.3 – Введення даних

Після цього користувач може ввести інформацію про ще одну операцію (рисунок 3.4, кнопка «Add more», рисунок 3.5) або перейти до скрінінгу введених даних (рисунок 3.6).



The screenshot shows the same interface as Figure 3.3, but with the 'Add operations manually' section expanded. It is titled 'Operation 1' and contains several input fields: 'Operation name *', 'Optimistic estimate *', 'Realistic estimate *', 'Pessimistic estimate *', 'Optimistic weight *', 'Realistic weight *', and 'Pessimistic weight *'. Below the form are two buttons: 'Add more' (green) and 'Next step' (blue).

Рисунок 3.4 – Поле введення інформації про першу операцію

Operation 1

Operation name *

Optimistic estimate * Realistic estimate * Pessimistic estimate *

2

6

15

Optimistic weight * Realistic weight * Pessimistic weight *

2

4

10

Operation 2

Operation name *

Optimistic estimate * Realistic estimate * Pessimistic estimate *

3

7

13

Optimistic weight * Realistic weight * Pessimistic weight *

3

4

8

Add more
Next step

Рисунок 3.5 – Дві введені операції

Після введення даних користувач натискає кнопку «Next step», що відправляє його до вкладки скрінінгу введених даних (рис. 3.6).

Enter the data
Preview the data
Result

Data preview

No	Operation name	Optimistic estimate	Optimistic weight	Realistic estimate	Realistic weight	Pessimistic estimate	Pessimistic weight
1	Operation 1	2	2	6	4	15	10
2	Operation 2	3	3	7	4	13	8

Back
Calculate

Рисунок 3.6 – Скрінінг даних

Для проведення розрахунків користувач натискає кнопку «Calculate». Після проведення розрахунків користувач бачить вкладку з результатами розрахунків (рисунок 3.7).

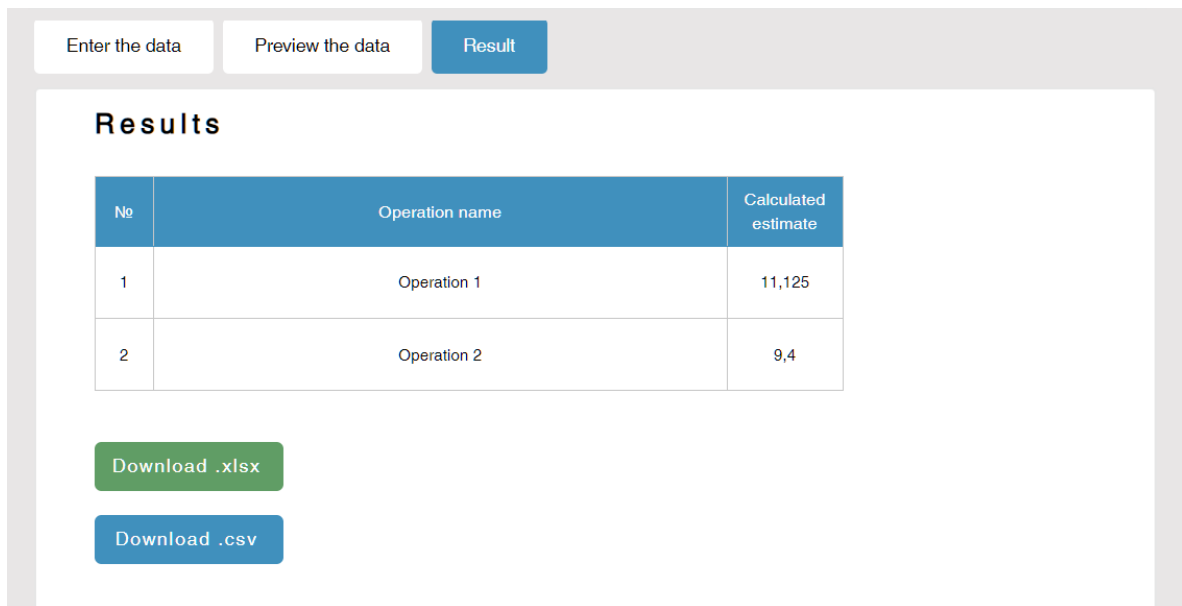


Рисунок 3.7– Результат розрахунків

Користувач має змогу завантажити результати розрахунків у форматі .csv та .xlsx, за потреби.

Обробка веб-сервісом вхідних даних описується наступною діаграмою прецедентів (рисунок 3.8).

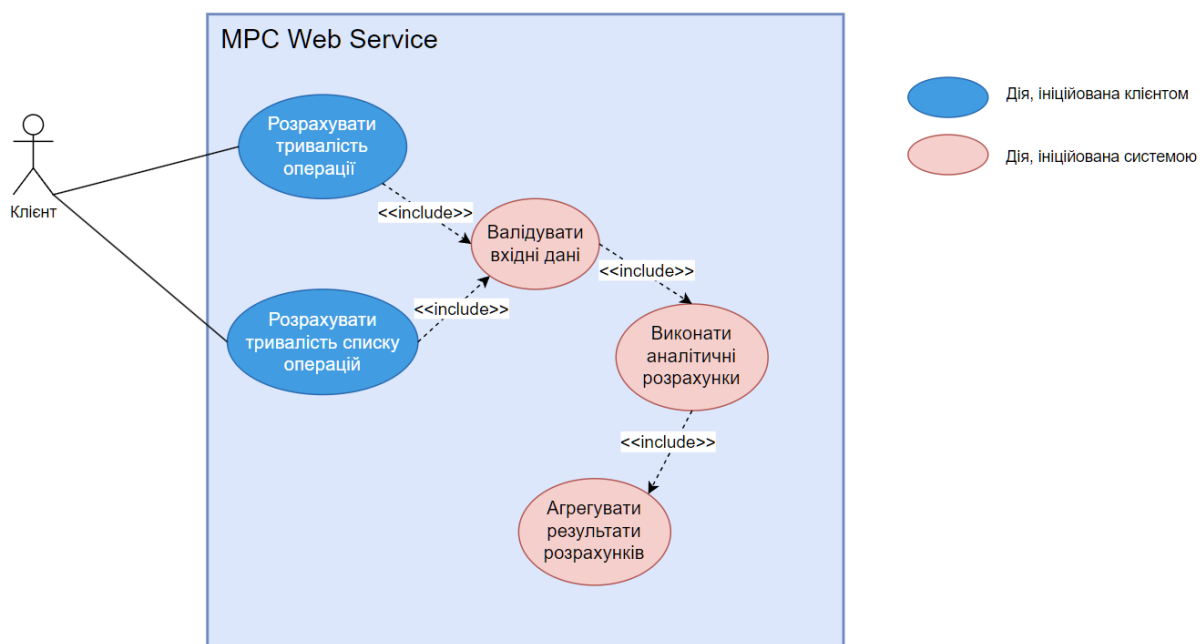


Рисунок 3.8 – Обробка вхідних даних веб-сервісом

Детальну послідовність успішної обробки вхідних даних наведено на діаграмі послідовностей (рисунок 3.9).

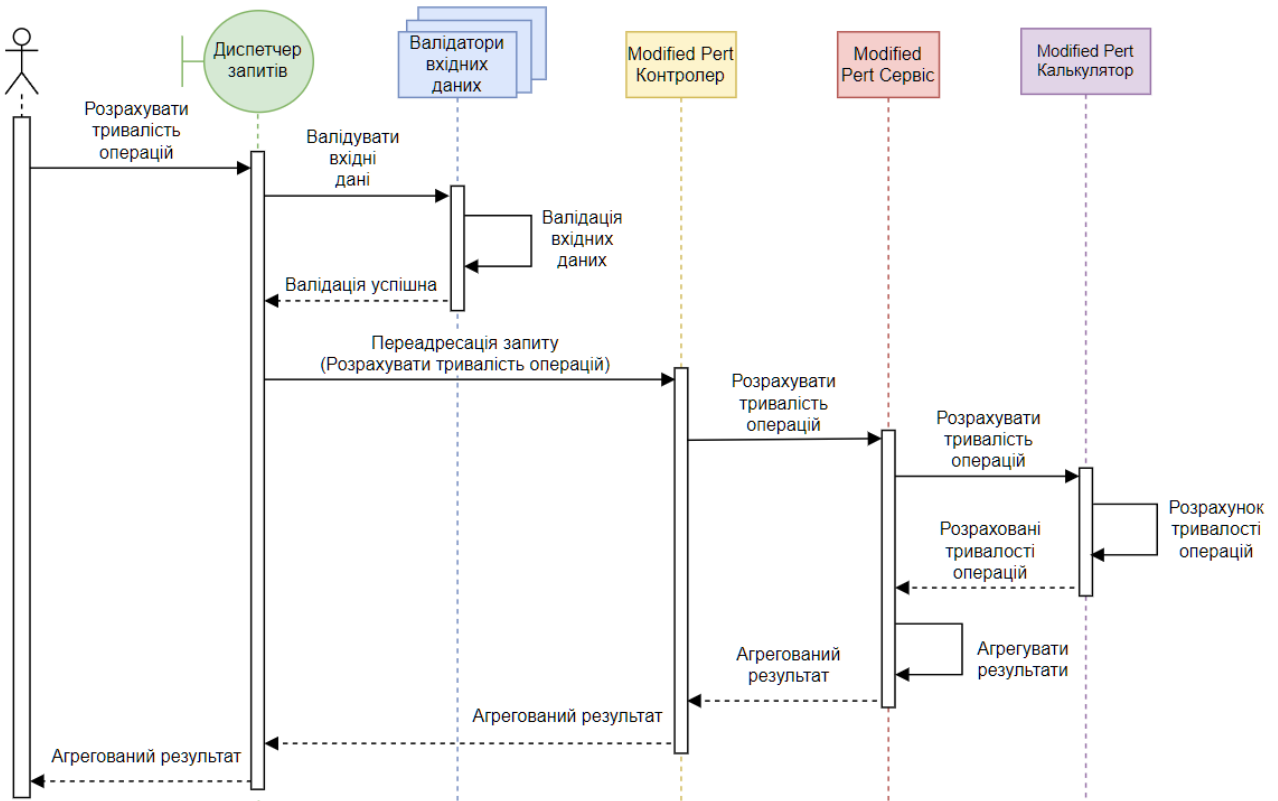


Рисунок 3.9 – Послідовність обробки вхідних даних компонентами веб-сервісу

На діаграмі послідовностей наведено усі ключові компоненти веб-сервісу PERT-based parametric calculator та їх місце у обробці вхідних даних від клієнту. З особливостей можна відмітити відсутність будь-якого сховища даних, оскільки будь-яке зберігання даних від клієнта не планувалося і не буде плануватися у майбутньому.

3.2 Впровадження інформаційної технології

Існує два способи впровадження та використання веб-сервісу PERT-based parametric calculator:

- використання веб-сервісу як окремого back-end компоненту через HTTP-клієнти на кшталт Postman;
- використання окремого компоненту – користувацького інтерфейсу для виконання запитів та отримання результатів;
- безпосередня інтеграція веб-сервісу з будь-яким застосунком для менеджменту проєктів або окремим застосунком. Єдина умова – можливість надсилання HTTP-запитів та дотримання чіткого API (Application Programming Interface)-контракту.

PERT-based parametric calculator поточної версії не надає власний користувацький інтерфейс, проте надає відкритий та простий API для інтеграції з іншими застосунками для менеджменту проєктів.

3.3 Висновки

За результатами дослідження було розроблено та описано пілотну версію веб-сервісу PERT-based parametric calculator. Описано формат комунікації клієнтів з веб-сервісом. Розглянуто функціональні та структурні особливості веб-сервісу з використанням діаграм прецедентів та послідовностей. Визначені способи впровадження методу параметричного оцінювання на базі методу PERT для розрахунку тривалості операцій IT-проєкту з міграції IC до GCP.

4 ЕКСПЕРИМЕНТАЛЬНЕ ПІДТВЕРДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

4.1 Набір даних для порівняння точності оцінювання оригінального та синтезованого методів

Для визначення придатності розробленого методу параметричного оцінювання на базі методу PERT до виконання оцінки тривалості операцій для IT-проєкту з міграції IC до GCP, а також порівняння точності оцінки з оригінальним методом PERT, було виконано розрахунок оцінок тривалості операцій зазначеними вище методами на прикладі 90 задач, розподілених між 3 IT-проєктами з кодovими назвами DEPV (таблиця 4.1), НТЕВ (таблиця 4.2) та HSDS (таблиця 4.3). Колонка «Assignee» містить інформацію про виконавця задачі, а саме його посаду. Можливі значення: Junior Back-End Engineer (JBE), Middle Back-End Engineer (MBE), Senior Back-End Engineer (SBE), Team Lead (TL), Solutions Architect (SA).

Таблиця 4.1 – Операції проєкту DEPV

Ticket №	Ticket name	Assignee	Duration	Assignee experience, years
1	2	3	4	5
DEPV-78809	[Employee API] Implement CRUD basic operations	MBE	3	2
DEPV-63611	[Attachment API] Create attachment signed URL	SBE	5	3
DEPV-92216	[Messaging Service] Configure APP + BigQuery	MBE	3	2
DEPV-75102	[Data Seeding Util] Add event listener	MBE	2	3
DEPV-83675	[Data Seeding Util] Create repository + gitlab CI (subtask for DEVOPS)	JBE	4	0
DEPV-71403	[Attachment API] Create repository + gitlab CI (subtask for DEVOPS)	JBE	2	1

Продовження таблиці 4.1

1	2	3	4	5
DEPV-32892	[Messaging Service] API: Retrieve all messages	SBE	2	3
DEPV-17601	[Data Seeding Util] Smoke testing on DEV	MBE	4	3
DEPV-60910	[Data Seeding Util] Fix data duplication problem	MBE	2	1
DEPV-78941	[Employee API] Design solution	SA	6	6
DEPV-1028	[Messaging Service] Firestore vs BigQuery POC	TL	2	5
DEPV-45551	[Employee API] Implement microservice testing POC	MBE	3	3
DEPV-33775	[Attachment API] Design solution	SA	7	6
DEPV-29683	[Data Seeding Util] Fix flaky integration tests	MBE	3	2
DEPV-24223	[Attachment API] Create cloud storage test POC	MBE	3	3
DEPV-67618	[Messaging Service] Design solution	SA	6	6
DEPV-59080	[Messaging Service] Catch messages on API	MBE	3	5
DEPV-33389	[Employee API] Investigate slow performance issue + Document	JBE	5	5
DEPV-50438	[Messaging Service] Change Pub/Sub topic subscription from PUSH to PULL	SBE	2	4
DEPV-66639	[Attachment API] Get attachment metadata	MBE	3	2
DEPV-40306	[Data Seeding Util] Performance testing on big datasets	SBE	3	3
DEPV-93290	[Employee API] Replace Hibernate with MyBatis	MBE	6	3
DEPV-74792	[Messaging Service] Create repository + gitlab CI (subtask for DEVOPS)	JBE	2	1
DEPV-3642	[Employee API] Cover endpoints IT	MBE	4	2

Кінець таблиці 4.1

1	2	3	4	5
DEPV-12386	[Attachment API] Add bucket listener to load attachment metadata into system	MBE	5	2
DEPV-11489	[Messaging Service] Verify Pub/Sub messages quotas	JBE	3	1
DEPV-88993	[Employee API] Get employees per department with permissions	MBE	3	2
DEPV-51728	[Attachment API] Store attachment by byte array	TL	3	5
DEPV-44858	[Data Seeding Util] Design solution	SA	9	6
DEPV-79093	[Attachment API] Store attachment by URL	MBE	2	3

Таблиця 4.2 – Операції проєкту НТЕВ

Ticket №	Ticket name	Assignee	Duration	Assignee experience, years
1	2	3	4	5
HTEB-84669	[Person Data Loader] Implement data import/export operations	MBE	4	3
HTEB-87841	[Person Data Loader] Generate signed URL for person data uploads	SBE	3	5
HTEB-11330	[Person Data Loader] Set up application integration with BigQuery for person data analytics	MBE	2	3
HTEB-73640	[Person Data Loader] Develop event listener for real-time person data updates	MBE	2	2
HTEB-17153	[Person Data Loader] Establish repository and CI/CD pipeline for person data management (DEVOPS)	JBE	3	1
HTEB-99557	[Person Data Loader] API: Retrieve all person records from the database	JBE	3	1

Продовження таблиці 4.2

1	2	3	4	5
HTEB-92034	[Person Data Loader] Conduct smoke testing on development environment for data loading functionality	SBE	4	3
HTEB-39446	[Person Data Loader] Resolve issue of duplicate person data entries in the system	MBE	3	3
HTEB-85216	[Person Data Loader] Architect solution for efficient person data processing and storage	MBE	3	2
HTEB-14443	[Person Data Loader] Conduct Proof of Concept (POC) comparing Firestore and BigQuery for person data storage	SA	3	6
HTEB-56508	[Person Data Loader] Implement POC for microservice testing framework tailored for person data	TL	5	5
HTEB-99413	[Person Data Loader] Design scalable solution for managing person data across distributed systems	SA	8	6
HTEB-67726	[Person Data Loader] Address flakiness in integration tests for reliable person data validation	MBE	3	3
HTEB-93637	[Person Data Loader] Create POC for testing person data storage in cloud environments	MBE	2	3
HTEB-32011	[Person Data Loader] Design schema for efficient storage and retrieval of person data	SA	5	6
HTEB-39097	[Person Data Loader] Develop mechanism to capture and process incoming person data streams efficiently	SA	7	7

Продовження таблиці 4.2

1	2	3	4	5
HTEB-91738	[Request API] Generate signed URLs for file attachments in requests	JBE	3	0
HTEB-14283	[Request API] Configure application to integrate with BigQuery for request analytics	SBE	2	4
HTEB-76353	[Request API] Develop event listener for real-time updates on request status	MBE	4	3
HTEB-49270	[Request API] Set up repository and CI/CD pipeline for request management (DEVOPS)	SBE	5	5
HTEB-64656	[Request API] API: Retrieve all requests from the database	MBE	3	2
HTEB-11915	[Request API] Conduct smoke testing on development environment for request processing	JBE	3	1
HTEB-70630	[Request API] Resolve issue of duplicate requests causing inconsistencies	MBE	4	2
HTEB-13515	[Request API] Design solution for efficient request routing and handling	MBE	4	3
HTEB-39969	[Request API] Conduct Proof of Concept (POC) comparing Firestore and BigQuery for request data storage	JBE	5	1
HTEB-63120	[Request API] Implement POC for microservice testing framework tailored for request API	MBE	6	3
HTEB-90128	[Request API] Design scalable solution for managing large volumes of requests	TL	3	5
HTEB-76268	[Request API] Address flakiness in integration tests for reliable request processing	MBE	4	2

Кінець таблиці 4.2

1	2	3	4	5
НТЕВ-58962	[Request API] Create POC for testing request data storage in cloud environments	MBE	5	2

Таблиця 4.3 – Операції проєкту HSDS

Ticket №	Ticket name	Assignee	Duration	Assignee experience, years
1	2	3	4	5
HSDS-57691	[Health Summary API] Develop CRUD operations for managing health summaries	MBE	4	3
HSDS-92761	[Health Summary API] Generate secure URLs for accessing health summary documents	SBE	2	5
HSDS-19766	[Health Summary API] Configure application to integrate with HIPAA-compliant storage for health data	MBE	3	3
HSDS-38859	[Health Summary API] Implement event listener for real-time updates on health summary records	MBE	3	3
HSDS-42221	[Health Summary API] Set up repository and CI/CD pipeline for health summary management (DEVOPS)	JBE	2	1
HSDS-65419	[Health Summary API] API: Retrieve all health summaries from the database	JBE	5	0
HSDS-92093	[Health Summary API] Conduct smoke testing on development environment for health summary processing	SBE	4	4
HSDS-18861	[Health Summary API] Resolve duplicated HS data affecting accuracy	MBE	6	3

Продовження таблиці 4.3

1	2	3	4	5
HSDS-82484	[Health Summary API] Design solution for secure and efficient retrieval of health summaries	MBE	5	2
HSDS-80347	[Health Summary API] Conduct Proof of Concept (POC) comparing different storage solutions for health data	SA	3	6
HSDS-10160	[Health Summary API] Implement POC for testing microservice architecture for health summary management	TL	2	7
HSDS-16821	[Health Summary API] Design scalable solution for handling large volumes of health data	SA	3	9
HSDS-54239	[Health Summary API] Address flakiness in integration tests for reliable health summary processing	MBE	4	7
HSDS-51648	[Health Summary API] Create POC for testing health data storage in cloud environments	MBE	3	5
HSDS-24432	[Health Summary API] Design schema for storing and querying health summary data efficiently	SA	6	7
HSDS-29891	[PHI Service] Develop secure API endpoints for accessing protected health information (PHI)	SA	5	8
HSDS-51219	[PHI Service] Generate signed URLs for sharing PHI documents with authorized entities	MBE	3	2
HSDS-47283	[PHI Service] Configure application to comply with HIPAA regulations for PHI storage and transmission	JBE	4	0

Продовження таблиці 4.3

1	2	3	4	5
HSDS-39310	[PHI Service] Implement event-driven architecture to handle PHI updates in real-time	SBE	2	4
HSDS-77726	[PHI Service] Set up repository and CI/CD pipeline for PHI management (DEVOPS)	MBE	3	3
HSDS-60745	[PHI Service] API: Retrieve PHI records securely from the database	SBE	6	5
HSDS-34677	[PHI Service] Conduct smoke testing on development environment to ensure PHI security measures	MBE	4	3
HSDS-80151	[PHI Service] Resolve issues related to PHI data duplication to maintain data integrity	JBE	4	3
HSDS-29200	[PHI Service] Design solution for secure and auditable access to PHI data	MBE	2	2
HSDS-25637	[PHI Service] Conduct Proof of Concept (POC) to evaluate different encryption methods for PHI protection	MBE	6	3
HSDS-46605	[PHI Service] Implement POC for testing microservice architecture for PHI management	JBE	5	2
HSDS-77987	[PHI Service] Design scalable solution for handling PHI requests while maintaining privacy	MBE	3	2
HSDS-57048	[PHI Service] Address integration test failures to ensure PHI data security	TL		6
HSDS-96102	[PHI Service] Create POC for testing PHI data storage in encrypted cloud environments	MBE	3	2

Кінець таблиці 4.3

1	2	3	4	5
HSDS-50550	[PHI Service] Design schema for storing and querying PHI data while adhering to regulatory requirements	MBE	3	2

4.2 Оцінка тривалості операцій за оригінальним методом PERT

Було виконано оцінку тривалості операцій, наведених у таблицях 4.1-4.3, використовуючи оригінальний метод PERT. Результати розрахунків наведено у таблицях 4.4-4.6. В таблицях наведені оцінки: оптимістична (О), найімовірніша (Н) та песимістична (П). Також, кінцева оцінка показує розраховану тривалість виконання операції, зазначену в робочих днях.

Таблиця 4.4 – Оцінка тривалості операцій проєкту DEPV оригінальним методом PERT

№	Операція	О, дні	Н, дні	П, дні	Кінцева оцінка
1	2	3	4	5	6
DEPV-78809	[Employee API] Implement CRUD basic operations	2	3	6	3,33
DEPV-63611	[Attachment API] Create attachment signed URL	4	5	8	5,33
DEPV-92216	[Messaging Service] Configure APP + BigQuery	2	3	4	3
DEPV-75102	[Data Seeding Util] Add event listener	1	2	5	2,33
DEPV-83675	[Data Seeding Util] Create repository + gitlab CI (subtask for DEVOPS)	3	4	4	3,83
DEPV-71403	[Attachment API] Create repository + gitlab CI (subtask for DEVOPS)	1	2	3	2

Продовження таблиці 4.4

1	2	3	4	5	6
DEPV-32892	[Messaging Service] API: Retrieve all messages	1	2	4	2,16
DEPV-17601	[Data Seeding Util] Smoke testing on DEV	3	4	5	4
DEPV-60910	[Data Seeding Util] Fix data duplication problem	1	2	4	2,16
DEPV-78941	[Employee API] Design solution	5	6	8	6,16
DEPV-1028	[Messaging Service] Firestore vs BigQuery POC	1	2	4	2,16
DEPV-45551	[Employee API] Implement microservice testing POC	3	3	5	3,33
DEPV-33775	[Attachment API] Design solution	4	7	9	6,83
DEPV-29683	[Data Seeding Util] Fix flaky integration tests	2	3	4	3
DEPV-24223	[Attachment API] Create cloud storage test POC	2	3	5	3,16
DEPV-67618	[Messaging Service] Design solution	4	6	9	6,16
DEPV-59080	[Messaging Service] Catch messages on API	2	3	5	3,16
DEPV-33389	[Employee API] Investigate slow performance issue + Document	2	5	7	4,83
DEPV-50438	[Messaging Service] Change Pub/Sub topic subscription from PUSH to PULL	1	2	5	2,3
DEPV-66639	[Attachment API] Get attachment metadata	2	3	7	3,5
DEPV-40306	[Data Seeding Util] Performance testing on big datasets	1	3	4	2,83
DEPV-93290	[Employee API] Replace Hibernate with MyBatis	3	6	10	6,16

Кінець таблиці 4.4

1	2	3	4	5	6
DEPV-3642	[Employee API] Cover endpoints with IT tests	3	4	5	4
DEPV-12386	[Attachment API] Add bucket listener to load attachment metadato into system	2	5	6	4,6
DEPV-11489	[Messaging Service] Verify Pub/Sub messages quotas	2	3	6	3,3
DEPV-88993	[Employee API] Get employees per department with permissions	2	3	4	3
DEPV-51728	[Attachment API] Store attachment by byte array	1	3	6	3,16
DEPV-44858	[Data Seeding Util] Design solution	5	9	15	9,3
DEPV-79093	[Attachment API] Store attachment by URL	1	2	6	2,5

Таблиця 4.5 – Оцінка тривалості операцій проекту НТЕВ оригінальним методом PERT

№	Операція	О, дні	Н, дні	П, дні	Кінцева оцінка
1	2	3	4	5	6
НТЕВ-84669	[Person Data Loader] Implement data import/export operations	3	4	7	4,33
НТЕВ-87841	[Person Data Loader] Generate signed URL for person data uploads	2	3	7	3,5
НТЕВ-11330	[Person Data Loader] Set up application integration with BigQuery for person data analytics	1	2	6	2,5
НТЕВ-73640	[Person Data Loader] Develop event listener for real-time person data updates	1	2	5	2,33
НТЕВ-17153	[Person Data Loader] Establish repository and CI/CD pipeline for person data management (DEVOPS)	1	3	4	2,83

Продовження таблиці 4.5

1	2	3	4	5	6
HTEB-99557	[Person Data Loader] API: Retrieve all person records from the database	1	3	5	3
HTEB-92034	[Person Data Loader] Conduct smoke testing on development environment for data loading functionality	2	4	6	4
HTEB-39446	[Person Data Loader] Resolve issue of duplicate person data entries in the system	2	3	6	3,33
HTEB-85216	[Person Data Loader] Architect solution for efficient person data processing and storage	1	3	5	3
HTEB-14443	[Person Data Loader] Conduct Proof of Concept (POC) comparing Firestore and BigQuery for person data storage	2	3	7	3,5
HTEB-56508	[Person Data Loader] Implement POC for microservice testing framework tailored for person data	2	5	8	5
HTEB-99413	[Person Data Loader] Design scalable solution for managing person data across distributed systems	5	8	12	8,17
HTEB-67726	[Person Data Loader] Address flakiness in integration tests for reliable person data validation	2	3	5	3,17
HTEB-93637	[Person Data Loader] Create POC for testing person data storage in cloud environments	1	2	6	2,5
HTEB-32011	[Person Data Loader] Design schema for efficient storage and retrieval of person data	3	5	8	5,17
HTEB-39097	[Person Data Loader] Develop mechanism to capture and process incoming person data streams efficiently	4	7	9	6,83

Продовження таблиці 4.5

1	2	3	4	5	6
HTEB-32246	[Request API] Implement CRUD operations for managing requests	2	5	9	5,17
HTEB-91738	[Request API] Generate signed URLs for file attachments in requests	3	3	6	3,5
HTEB-14283	[Request API] Configure application to integrate with BigQuery for request analytics	2	2	6	2,67
HTEB-76353	[Request API] Develop event listener for real-time updates on request status	2	4	5	3,83
HTEB-49270	[Request API] Set up repository and CI/CD pipeline for request management (DEVOPS)	2	5	7	4,83
HTEB-64656	[Request API] API: Retrieve all requests from the database	1	3	6	3,17
HTEB-11915	[Request API] Conduct smoke testing on development environment for request processing	2	3	7	3,5
HTEB-70630	[Request API] Resolve issue of duplicate requests causing inconsistencies	1	4	9	4,33
HTEB-13515	[Request API] Design solution for efficient request routing and handling	2	4	7	4,17
HTEB-39969	[Request API] Conduct Proof of Concept (POC) comparing Firestore and BigQuery for request data storage	2	5	6	4,67
HTEB-63120	[Request API] Implement POC for microservice testing framework tailored for request API	3	6	8	5,83
HTEB-90128	[Request API] Design scalable solution for managing large volumes of requests	2	3	5	3,17
HTEB-76268	[Request API] Address flakiness in integration tests for reliable request processing	2	4	6	4

Кінець таблиці 4.5

1	2	3	4	5	6
НТЕВ-58962	[Request API] Create POC for testing request data storage in cloud environments	3	5	9	5,33

Таблиця 4.6 – Оцінка тривалості операцій проєкту HSDS оригінальним методом PERT

№	Операція	О, дні	Н, дні	П, дні	Кінцева оцінка
1	2	3	4	5	6
HSDS-57691	[Health Summary API] Develop CRUD operations for managing health summaries	3	4	5	4
HSDS-92761	[Health Summary API] Generate secure URLs for accessing health summary documents	1	2	6	2,5
HSDS-19766	[Health Summary API] Configure application to integrate with HIPAA-compliant storage for health data	2	3	5	3,17
HSDS-38859	[Health Summary API] Implement event listener for real-time updates on health summary records	1	3	6	3,17
HSDS-42221	[Health Summary API] Set up repository and CI/CD pipeline for health summary management (DEVOPS)	1	2	7	2,67
HSDS-65419	[Health Summary API] API: Retrieve all health summaries from the database	3	5	6	4,83
HSDS-92093	[Health Summary API] Conduct smoke testing on development environment for health summary processing	2	4	5	3,83
HSDS-18861	[Health Summary API] Resolve issue of duplicated health summary entries affecting accuracy	2	6	7	5,5
HSDS-82484	[Health Summary API] Design solution for secure and efficient retrieval of HS	3	5	8	5,17

Продовження таблиці 4.6

1	2	3	4	5	6
HSDS-80347	[Health Summary API] Conduct Proof of Concept (POC) comparing different storage solutions for health data	2	3	8	3,67
HSDS-10160	[Health Summary API] Implement POC for testing microservice architecture for health summary management	1	2	6	2,5
HSDS-16821	[Health Summary API] Design scalable solution for handling large volumes of health data	2	3	8	3,67
HSDS-54239	[Health Summary API] Address flakiness in integration tests for reliable health summary processing	2	4	6	4
HSDS-51648	[Health Summary API] Create POC for testing health data storage in cloud environments	1	3	4	2,83
HSDS-24432	[Health Summary API] Design schema for storing and querying health summary data efficiently	3	6	7	5,67
HSDS-29891	[PHI Service] Develop secure API endpoints for accessing protected health information (PHI)	2	5	7	4,83
HSDS-51219	[PHI Service] Generate signed URLs for sharing PHI documents with authorized entities	2	3	8	3,67
HSDS-47283	[PHI Service] Configure application to comply with HIPAA regulations for PHI storage and transmission	3	4	5	4
HSDS-39310	[PHI Service] Implement event-driven architecture to handle PHI updates in real-time	1	2	5	2,33

Кінець таблиці 4.6

1	2	3	4	5	6
HSDS-34677	[PHI Service] Conduct smoke testing on development environment to ensure PHI security measures	1	4	6	3,83
HSDS-80151	[PHI Service] Resolve issues related to PHI data duplication to maintain data integrity	3	4	6	4,17
HSDS-29200	[PHI Service] Design solution for secure and auditable access to PHI data	1	2	5	2,33
HSDS-25637	[PHI Service] Conduct Proof of Concept (POC) to evaluate different encryption methods for PHI protection	3	6	9	6
HSDS-46605	[PHI Service] Implement POC for testing microservice architecture for PHI management	2	5	9	5,17
HSDS-77987	[PHI Service] Design scalable solution for handling PHI requests while maintaining privacy	2	3	6	3,33
HSDS-57048	[PHI Service] Address integration test failures to ensure PHI data security	3	5	7	5
HSDS-96102	[PHI Service] Create POC for testing PHI data storage in encrypted cloud environments	2	3	6	3,33
HSDS-50550	[PHI Service] Design schema for storing and querying PHI data while adhering to regulatory requirements	1	3	4	2,83

4.3 Оцінка тривалості операцій за методом параметричного оцінювання на базі методу PERT

Для виконання розрахунків за синтезованим методом було визначено шкалу вагових коефіцієнтів сценаріїв шляхом анонімного опитування 20 незалежних експертів. Результати анкетування наведено у таблиці 4.7.

Таблиця 4.7 – Результати анкетування вибору шкали вагових коефіцієнтів

Варіант шкали	Кількість голосів
1-5	4
1-10	11
Числа Фібоначчі	4
Натуральні числа	1

Результати розрахунків наведено у таблицях 4.8-4.10. В таблицях наведені оцінки: оптимістична (О), найімовірніша (Н) та песимістична (П). Кожній оцінці зіставлено ваговий коефіцієнт, а саме: ваговий коефіцієнт оптимізму (ВКО), ваговий коефіцієнт реалізму (ВКР) та ваговий коефіцієнт песимізму (ВКП), що визначає ступінь впливу сценарію на кінцеву оцінку, вказану в робочих днях.

Таблиця 4.8 – Оцінка тривалості операцій проєкту DEPV синтезованим методом

№	Операція	О, дні	ВКО	Н, дні	ВКР	П, дні	ВКП	Кінцева оцінка
1	2	3	4	5	6	7	8	9
DEPV-78809	[Employee API] Implement CRUD basic operations	2	2	3	2	6	6	4,6
DEPV-63611	[Attachment API] Create attachment signed URL	4	1	5	4	8	5	6,4
DEPV-92216	[Messaging Service] Configure APP + BigQuery	2	1	3	5	4	4	3,3
DEPV-75102	[Data Seeding Util] Add event listener	1	2	2	2	5	6	3,6
DEPV-83675	[Data Seeding Util] Create repository + gitlab CI (subtask for DEVOPS)	3	1	4	4	4	5	3,9
DEPV-71403	[Attachment API] Create repository + gitlab CI (subtask for DEVOPS)	1	2	2	4	3	4	2,2
DEPV-32892	[Messaging Service] API: Retrieve all messages	1	2	2	4	4	4	2,6
DEPV-17601	[Data Seeding Util] Smoke testing on DEV	3	1	4	4	5	5	4,4
DEPV-60910	[Data Seeding Util] Fix data duplication problem	1	2	2	2	4	6	3
DEPV-78941	[Employee API] Design solution	5	2	6	1	8	7	7,2
DEPV-1028	[Messaging Service] Firestore vs BigQuery POC	1	1	2	3	4	6	3,1

Продовження таблиці 4.8

1	2	3	4	5	6	7	8	9
DEPV-45551	[Employee API] Implement microservice testing POC	3	2	3	3	5	5	4
DEPV-33775	[Attachment API] Design solution	4	2	7	3	9	5	7,4
DEPV-29683	[Data Seeding Util] Fix flaky integration tests	2	2	3	1	4	7	3,5
DEPV-24223	[Attachment API] Create cloud storage test POC	2	2	3	2	5	6	4
DEPV-67618	[Messaging Service] Design solution	4	1	6	4	9	5	7,3
DEPV-59080	[Messaging Service] Catch messages on API	2	2	3	1	5	7	4,2
DEPV-33389	[Employee API] Investigate slow performance issue + Document	2	1	5	3	7	6	5,9
DEPV-50438	[Messaging Service] Change Pub/Sub topic subscription from PUSH to PULL	1	1	2	4	5	5	3,4
DEPV-66639	[Attachment API] Get attachment metadata	2	1	3	2	7	7	5,7
DEPV-40306	[Data Seeding Util] Performance testing on big datasets	1	1	3	4	4	5	3,3
DEPV-93290	[Employee API] Replace Hibernate with MyBatis	3	2	6	2	10	6	7,8
DEPV-74792	[Messaging Service] Create repository + gitlab CI (subtask for DEVOPS)	1	1	2	4	5	5	3,4

Кінець таблиці 4.8

1	2	3	4	5	6	7	8	9
DEPV-3642	[Employee API] Cover endpoints with IT tests	3	1	4	2	5	7	4,6
DEPV-12386	[Attachment API] Add bucket listener to load attachment metadata into system	2	1	5	4	6	5	5,2
DEPV-11489	[Messaging Service] Verify Pub/Sub messages quotas	2	2	3	2	6	6	4,6
DEPV-88993	[Employee API] Get employees per department with permissions	2	2	3	3	4	5	3,3
DEPV-51728	[Attachment API] Store attachment by byte array	1	1	3	3	6	6	4,6
DEPV-44858	[Data Seeding Util] Design solution	5	2	9	3	15	5	11,2
DEPV-79093	[Attachment API] Store attachment by URL	1	1	2	2	6	7	4,7

Таблиця 4.9 – Оцінка тривалості операцій проєкту НТЕВ синтезованим методом

№	Операція	О, дні	ВКО	Н, дні	ВКР	П, дні	ВКП	Кінцева оцінка
1	2	3	4	5	6	7	8	9
НТЕВ-84669	[Person Data Loader] Implement data import/export operations	3	2	4	2	7	6	5,6
НТЕВ-87841	[Person Data Loader] Generate signed URL for person data uploads	2	1	3	4	7	5	4,9

Продовження таблиці 4.9

1	2	3	4	5	6	7	8	9
HTEB-11330	[Person Data Loader] Set up application integration with BigQuery for person data analytics	1	1	2	5	6	4	3,5
HTEB-73640	[Person Data Loader] Develop event listener for real-time person data updates	1	2	2	2	5	6	3,6
HTEB-17153	[Person Data Loader] Establish repository and CI/CD pipeline for person data management (DEVOPS)	1	1	3	4	4	5	3,3
HTEB-99557	[Person Data Loader] API: Retrieve all person records from the database	1	2	3	4	5	4	3,4
HTEB-92034	[Person Data Loader] Conduct smoke testing on development environment for data loading functionality	2	2	4	4	6	4	4,4
HTEB-39446	[Person Data Loader] Resolve issue of duplicate person data entries in the system	2	1	3	4	6	5	4,4
HTEB-85216	[Person Data Loader] Architect solution for efficient person data processing and storage	1	2	3	2	5	6	3,8
HTEB-14443	[Person Data Loader] Conduct Proof of Concept (POC) comparing Firestore and BigQuery for person data storage	2	2	3	1	7	7	5,6

Продовження таблиці 4.9

1	2	3	4	5	6	7	8	9
НТЕБ-56508	[Person Data Loader] Implement POC for microservice testing framework tailored for person data	2	1	5	3	8	6	6,5
НТЕБ-99413	[Person Data Loader] Design scalable solution for managing person data across distributed systems	5	2	8	3	12	5	9,4
НТЕБ-67726	[Person Data Loader] Address flakiness in integration tests for reliable person data validation	2	2	3	3	5	5	3,8
НТЕБ-93637	[Person Data Loader] Create POC for testing person data storage in cloud environments	1	2	2	1	6	7	4,6
НТЕБ-32011	[Person Data Loader] Design schema for efficient storage and retrieval of person data	3	2	5	2	8	6	6,4
НТЕБ-39097	[Person Data Loader] Develop mechanism to capture and process incoming person data streams efficiently	4	1	7	4	9	5	7,7
НТЕБ-32246	[Request API] Implement CRUD operations for managing requests	2	2	5	1	9	7	7,2
НТЕБ-91738	[Request API] Generate signed URLs for file attachments in requests	3	1	3	3	6	6	4,8

Продовження таблиці 4.9

1	2	3	4	5	6	7	8	9
HTEB-14283	[Request API] Configure application to integrate with BigQuery for request analytics	2	1	2	4	6	5	4
HTEB-76353	[Request API] Develop event listener for real-time updates on request status	2	1	4	2	5	7	4,5
HTEB-49270	[Request API] Set up repository and CI/CD pipeline for request management (DEVOPS)	2	1	5	4	7	5	5,7
HTEB-64656	[Request API] API: Retrieve all requests from the database	1	2	3	2	6	6	4,4
HTEB-11915	[Request API] Conduct smoke testing on development environment for request processing	2	1	3	4	7	5	4,9
HTEB-70630	[Request API] Resolve issue of duplicate requests causing inconsistencies	1	1	4	2	9	7	7,2
HTEB-13515	[Request API] Design solution for efficient request routing and handling	2	1	4	4	7	5	5,3
HTEB-39969	[Request API] Conduct Proof of Concept (POC) comparing Firestore and BigQuery for request data storage	2	2	5	2	6	6	5

Кінець таблиці 4.9

1	2	3	4	5	6	7	8	9
НТЕВ-63120	[Request API] Implement POC for microservice testing framework tailored for request API	3	2	6	3	8	5	6,4
НТЕВ-90128	[Request API] Design scalable solution for managing large volumes of requests	2	1	3	3	5	6	4,1
НТЕВ-76268	[Request API] Address flakiness in integration tests for reliable request processing	2	2	4	3	6	5	4,6
НТЕВ-58962	[Request API] Create POC for testing request data storage in cloud environments	3	1	5	2	9	7	7,6

Таблиця 4.10 – Оцінка тривалості операцій проєкту HSDS синтезованим методом

№	Операція	О, дні	ВКО	Н, дні	ВКР	П, дні	ВКП	Кінцева оцінка
1	2	3	4	5	6	7	8	9
HSDS-57691	[Health Summary API] Develop CRUD operations for managing health summaries	3	2	4	2	5	6	4,4
HSDS-92761	[Health Summary API] Generate secure URLs for accessing health summary documents	1	1	2	4	6	5	3,9

Продовження таблиці 4.10

1	2	3	4	5	6	7	8	9
HSDS-19766	[Health Summary API] Configure application to integrate with HIPAA-compliant storage for health data	2	1	3	5	5	4	3,7
HSDS-38859	[Health Summary API] Implement event listener for real-time updates on health summary records	1	2	3	2	6	6	4,4
HSDS-42221	[Health Summary API] Set up repository and CI/CD pipeline for health summary management (DEVOPS)	1	1	2	4	7	5	4,4
HSDS-65419	[Health Summary API] API: Retrieve all health summaries from the database	3	2	5	4	6	4	5
HSDS-92093	[Health Summary API] Conduct smoke testing on development environment for health summary processing	2	2	4	4	5	4	4
HSDS-18861	[Health Summary API] Resolve issue of duplicated health summary entries affecting accuracy	2	1	6	4	7	5	6,1
HSDS-82484	[Health Summary API] Design solution for secure and efficient retrieval of health summaries	3	2	5	2	8	6	6,4

Продовження таблиці 4.10

1	2	3	4	5	6	7	8	9
HSDS-80347	[Health Summary API] Conduct Proof of Concept (POC) comparing different storage solutions for health data	2	2	3	1	8	7	6,3
HSDS-10160	[Health Summary API] Implement POC for testing microservice architecture for health summary management	1	1	2	3	6	6	4,3
HSDS-16821	[Health Summary API] Design scalable solution for handling large volumes of health data	2	2	3	3	8	5	5,3
HSDS-54239	[Health Summary API] Address flakiness in integration tests for reliable health summary processing	2	2	4	3	6	5	4,6
HSDS-51648	[Health Summary API] Create POC for testing health data storage in cloud environments	1	2	3	1	4	7	3,3
HSDS-24432	[Health Summary API] Design schema for storing and querying health summary data efficiently	3	2	6	2	7	6	6
HSDS-29891	[PHI Service] Develop secure API endpoints for accessing protected health information (PHI)	2	1	5	4	7	5	5,7

Продовження таблиці 4.10

1	2	3	4	5	6	7	8	9
HSDS-51219	[PHI Service] Generate signed URLs for sharing PHI documents with authorized entities	2	2	3	1	8	7	6,3
HSDS-47283	[PHI Service] Configure application to comply with HIPAA regulations for PHI storage and transmission	3	1	4	3	5	6	4,5
HSDS-39310	[PHI Service] Implement event-driven architecture to handle PHI updates in real-time	1	1	2	4	5	5	3,4
HSDS-77726	[PHI Service] Set up repository and CI/CD pipeline for PHI management (DEVOPS)	2	1	3	2	5	7	4,3
HSDS-60745	[PHI Service] API: Retrieve PHI records securely from the database	3	1	6	4	7	5	6,2
HSDS-34677	[PHI Service] Conduct smoke testing on development environment to ensure PHI security measures	1	2	4	2	6	6	4,6
HSDS-80151	[PHI Service] Resolve issues related to PHI data duplication to maintain data integrity	3	1	4	4	6	5	4,9
HSDS-29200	[PHI Service] Design solution for secure and auditable access to PHI data	1	1	2	2	5	7	4

Кінець таблиці 4.10

1	2	3	4	5	6	7	8	9
HSDS-25637	[PHI Service] Conduct Proof of Concept (POC) to evaluate different encryption methods for PHI protection	3	1	6	4	9	5	7,2
HSDS-46605	[PHI Service] Implement POC for testing microservice architecture for PHI management	2	2	5	2	9	6	6,8
HSDS-77987	[PHI Service] Design scalable solution for handling PHI requests while maintaining privacy	2	2	3	3	6	5	4,3
HSDS-57048	[PHI Service] Address integration test failures to ensure PHI data security	3	1	5	3	7	6	6
HSDS-96102	[PHI Service] Create POC for testing PHI data storage in encrypted cloud environments	2	2	3	3	6	5	4,3
HSDS-50550	[PHI Service] Design schema for storing and querying PHI data while adhering to regulatory requirements	1	1	3	2	4	7	3,5

4.4 Порівняльний аналіз точності оцінювання оригінального та синтезованого методів

Інформацію щодо точності оцінки тривалості операцій проєкту DEPV, НТЕВ та HSDS наведено у таблицях 4.11-4.13 відповідно. Точність оцінки тривалості операцій у відсотках розраховувано наступним чином

$$A = \left| 1 - \frac{|o - e|}{o} \cdot 100\% \right|, \quad (4.1)$$

де A – точність оцінки тривалості операції;

o – спостережуване значення тривалості операції;

e – значення тривалості операції, розраховане за відповідним методом.

Таблиця 4.11 – Точність оцінювання операцій проєкту DEPV

№	Оцінка за методом PERT, дні	Оцінка за методом параметричного оцінювання на базі методу PERT, дні	Спостережувана тривалість, дні	Точність оцінки методом PERT, %	Точність оцінки методом параметричного оцінювання на базі методу PERT, %
1	2	3	4	5	6
DEPV-78809	3,33	4,6	5	66,6	92
DEPV-63611	5,33	6,4	7	76,14	91,43
DEPV-92216	3	3,3	3,5	85,71	94,29
DEPV-75102	2,33	3,6	3	77,67	80
DEPV-83675	3,83	3,9	4	95,75	97,5

Продовження таблиці 4.11

1	2	3	4	5	6
DEPV- 71403	2	2,2	2	100	90
DEPV- 32892	2,17	2,6	2	91,5	70
DEPV- 17601	4	4,4	3	66,67	53,33
DEPV- 60910	2,17	3	3	72,33	100
DEPV- 78941	6,17	7,2	6,5	94,92	89,23
DEPV- 1028	2,17	3,1	3	72,33	96,67
DEPV- 45551	3,33	4	4	83,25	100
DEPV- 33775	6,83	7,4	7	97,57	94,29
DEPV- 29683	3	3,5	4	75	87,5
DEPV- 24223	3,17	4	3,5	90,57	85,71
DEPV- 67618	6,17	7,3	7	88,14	95,71
DEPV- 59080	3,17	4,2	4	79,25	95
DEPV- 33389	4,83	5,9	6	80,5	98,33
DEPV- 50438	2,33	3,4	3	77,67	86,67
DEPV- 66639	3,5	5,7	6	58,33	95
DEPV- 40306	2,83	3,3	3	94,33	90

Кінець таблиці 4.11

1	2	3	4	5	6
DEPV-93290	6,17	7,8	7	88,14	88,57
DEPV-74792	2,33	3,4	3,5	66,57	97,14
DEPV-3642	4	4,6	4	100	85
DEPV-12386	4,67	5,2	5	93,4	96
DEPV-11489	3,33	4,6	5	66,6	92
DEPV-88993	3	3,3	4	75	82,5
DEPV-51728	3,17	4,6	4,5	70,44	97,78
DEPV-44858	9,33	11,2	11	84,82	98,18

Таблиця 4.12 – Точність оцінювання операцій проєкту НТЕВ

№	Оцінка за методом PERT, дні	Оцінка за методом параметричного оцінювання на базі методу PERT, дні	Спостережувана тривалість, дні	Точність оцінки методом PERT, %	Точність оцінки методом параметричного оцінювання на базі методу PERT, %
1	2	3	4	5	6
НТЕВ-84669	4,33	5,6	5	86,6	88
НТЕВ-87841	3,5	4,9	5	70	98
НТЕВ-11330	2,5	3,5	4	62,5	87,5

Продовження таблиці 4.12

1	2	3	4	5	6
НТЕВ-17153	2,83	3,3	3,5	80,86	94,29
НТЕВ-99557	3	3,4	4	75	85
НТЕВ-92034	4	4,4	4	100	90
НТЕВ-39446	3,33	4,4	4	83,25	90
НТЕВ-85216	3	3,8	3,5	85,71	91,43
НТЕВ-14443	3,5	5,6	6	58,33	93,33
НТЕВ-56508	5	6,5	6	83,33	91,67
НТЕВ-99413	8,17	9,4	9	90,78	95,56
НТЕВ-67726	3,17	3,8	3,5	90,57	91,43
НТЕВ-93637	2,5	4,6	4	62,5	85
НТЕВ-32011	5,17	6,4	6	86,17	93,33
НТЕВ-39097	6,83	7,7	8	85,38	96,25
НТЕВ-32246	5,17	7,2	7	73,86	97,14
НТЕВ-91738	3,5	4,8	4	87,5	80
НТЕВ-14283	2,67	4	4	66,75	100
НТЕВ-76353	3,83	4,5	4	95,75	87,5

Кінець таблиці 4.12

1	2	3	4	5	6
НТЕВ-49270	4,83	5,7	6	80,5	95
НТЕВ-64656	3,17	4,4	5	63,4	88
НТЕВ-11915	3,5	4,9	5	70	98
НТЕВ-70630	4,33	7,2	7	61,86	97,14
НТЕВ-13515	4,17	5,3	5	83,4	94
НТЕВ-39969	4,67	5	4,5	96,22	88,89
НТЕВ-63120	5,83	6,4	6	97,17	93,33
НТЕВ-90128	3,17	4,1	4	79,25	97,5
НТЕВ-76268	4	4,6	4	100	85

Таблиця 4.13 – Точність оцінювання операцій проєкту HSDS

№	Оцінка за методом PERT, дні	Оцінка за методом параметричного оцінювання на базі методу PERT, дні	Спостережувана тривалість, дні	Точність оцінки методом PERT, %	Точність оцінки методом параметричного оцінювання на базі методу PERT, %
1	2	3	4	5	6
HSDS-57691	4	4,4	4	100	90
HSDS-92761	2,5	3,9	4	62,5	97,5

Продовження таблиці 4.13

1	2	3	4	5	6
HSDS-38859	3,17	4,4	5	63,4	88
HSDS-42221	2,67	4,4	4	66,75	90
HSDS-65419	4,83	5	5	96,6	100
HSDS-92093	3,83	4	4	95,75	100
HSDS-18861	5,5	6,1	6	91,67	98,33
HSDS-82484	5,17	6,4	6	86,17	93,33
HSDS-80347	3,67	6,3	7	52,43	90
HSDS-10160	2,5	4,3	5	50	86
HSDS-16821	3,67	5,3	3,5	95,14	48,57
HSDS-54239	4	4,6	3	66,67	46,67
HSDS-51648	2,83	3,3	3	94,33	90
HSDS-24432	5,67	6	6,5	87,23	92,31
HSDS-29891	4,83	5,7	7	69	81,43
HSDS-51219	3,67	6,3	6	61,17	95
HSDS-47283	4	4,5	4	100	87,5
HSDS-39310	2,33	3,4	3	77,67	86,67

Кінець таблиці 4.13

1	2	3	4	5	6
HSDS-77726	3,17	4,3	4	79,25	92,5
HSDS-60745	5,67	6,2	5	86,6	76
HSDS-34677	3,83	4,6	4,5	85,11	97,78
HSDS-80151	4,17	4,9	5	83,4	98
HSDS-29200	2,33	4	3	77,67	66,67
HSDS-25637	6	7,2	6	100	80
HSDS-46605	5,17	6,8	7	73,86	97,14
HSDS-77987	3,33	4,3	3	89	56,67
HSDS-57048	5	6	5	100	80
HSDS-96102	3,33	4,3	4	83,25	92,5

Середнє значення точності оцінювання тривалості операцій двома методами за кожним з проєктів наведено у таблиці 4.14.

Таблиця 4.14 – Середнє значення точності оцінки методів за проєктами

Проект	Середня точність оцінювання за методом PERT, %	Середня точність оцінювання за методом параметричного оцінювання на базі методу PERT, %
DEPV	80,83	90,51
НТЕВ	80,03	91,58
HSDS	81,61	85,81

Проаналізувавши результати розрахунків та показники точності за оригінальним методом PERT та методом параметричного оцінювання на базі методу PERT, зроблено висновок, що синтезований метод має перевагу над оригінальним методом PERT в середньому на 8,48%. Це пояснюється тим, що деякі з операцій, в виду свого контексту не спираються на найімовірніший сценарій, і саме через це оригінальний метод PERT надає менш точну оцінку. Наприклад, операцію HSDS-42221 виконував розробник рівня junior без попереднього досвіду, що й призвело до збільшення часу виконання операції до робочих 4 днів.

Обидва методи використовували 3 ідентичні сценарії, однак синтезований метод, базуючись на контексті операцій, дав змогу скоригувати вагу відповідного сценарію для формування кінцевої оцінки.

Контекстом операції в даному випадку є: кваліфікація виконавця, наявність попереднього досвіду у виконавця та наявність попереднього досвіду у команді розробників.

4.5 Висновки

В цьому розділі було виконано оцінку тривалості виконання операцій для трьох IT-проектів з міграції IC до GCP використовуючи оригінальний метод PERT та метод параметричного оцінювання на базі методу PERT. Результати оцінювання було порівняно з реальними (спостережуваними) результатами. Розраховано точність оцінювання двома методами. Аналіз точності показав, що синтезований метод в середньому на 8,48% точніший за оригінальний метод, що пояснюється відсутністю орієнтованості контексту розглянутих операцій на найімовірніший сценарій.

Є доречним провести додаткові дослідження та випробувати синтезований метод на більшій кількості IT-проектів з міграції IC до GCP, для

того, щоб остаточно визначити, чи сприяє використання синтезованого методу підвищенню точності оцінювання тривалості операцій для ІТ-проєкту з міграції ІС до GCP.

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи було проаналізовано існуючі методи оцінки тривалості операцій для ІТ-проектів з міграції ІС до хмарної платформи.

Під час аналізу було виявлено, що метод PERT є найкраще адаптованим до часової невизначеності за рахунок використання трьох можливих сценаріїв. Визначено проблему методу PERT для оцінки тривалості операцій в контексті ІТ-проекту з міграції ІС до GCP – орієнтованість на найімовірніший сценарій тривалості не є доречним для всіх випадків і знижує точність оцінювання для таких операцій. Як результат, було поставлено за мету підвищити точність оцінювання тривалості виконання операцій саме для ІТ-проектів з міграції ІС до GCP.

Для досягнення поставленої мети було синтезовано метод параметричного оцінювання на базі методу PERT. Синтезований метод використовує три можливі сценарії оцінки, як і оригінальний метод, а також ставить у відповідність кожному сценарію ваговий коефіцієнт, що визначає вплив сценарію на кінцеву оцінку тривалості операції. Використання вже раніше розроблених сценаріїв для оригінального методу сприяє полегшенню розрахунків, за умови, що для оцінки тривалості операцій вже використовувався оригінальний метод. Недоліком синтезованого методу є додаткові часові та трудові витрати на аналіз контексту операції і визначення вагових коефіцієнтів.

За результатами дослідження було розроблено та описано пілотну версію веб-сервісу PERT-based parametric calculator. Описано формат комунікації клієнтів з веб-сервісом. Розглянуто функціональні та структурні особливості веб-сервісу з використанням діаграм прецедентів та послідовностей. Визначені способи впровадження методу параметричного оцінювання на базі методу PERT для розрахунку тривалості операцій ІТ-проекту з міграції ІС до GCP.

Виконано оцінку тривалості виконання операцій для трьох ІТ-проектів з міграції ІС до GCP, використовуючи оригінальний метод PERT та метод параметричного оцінювання на базі методу PERT. Результати оцінювання було порівняно з реальними (спостережуваними) результатами. Розраховано точність оцінювання двома методами. Аналіз точності показав, що синтезований метод в середньому на 8,48% точніший за оригінальний метод, що пояснюється відсутністю орієнтованості контексту розглянутих операцій на найімовірніший сценарій.

Є доречним провести додаткові дослідження та випробувати синтезований метод на більшій кількості ІТ-проектів з міграції ІС до GCP, для того щоб остаточно визначити, чи сприяє використання синтезованого методу підвищенню точності оцінювання тривалості операцій для ІТ-проекту з міграції ІС до GCP.

Результати кваліфікаційної роботи обговорювалися на XX міжнародній конференції «Ways of distance learning development in current conditions» [37], міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» [38], опубліковані у науковому журналі «АСУ та прилади автоматики»[39].

Роботу виконано згідно вимог методичних вказівок [40]. Пояснювальну записку оформлено згідно державного стандарту України [41]. Перелік джерел оформлено згідно державного стандарту України [42].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Blog: On Premise vs. Cloud: Key Differences, Benefits and Risks. URL: <https://www.cleo.com/blog/knowledge-base-on-premise-vs-cloud> (дата звернення 18 березня 2024).
2. What is Expert Judgment in Project Management? URL: <https://www.wrike.com/project-management-guide/faq/what-is-expert-judgment-in-project-management> (дата звернення 18 березня 2024).
3. Analogous estimating. URL: <https://www.projectmanagement.com/wikis/368759/analogous-estimating> (дата звернення 18 березня 2024).
4. Project Evaluation Review Technique (PERT). URL: <https://corporatefinanceinstitute.com/resources/management/project-evaluation-review-technique-pert/> (дата звернення 18 березня 2024).
5. How To Run A Project Discussion Meeting. URL: <https://zipdo.co/guide/project-discussion-meeting/> (дата звернення 18 березня 2024).
6. Microsoft. Microsoft Project. URL: <https://www.microsoft.com/en-us/microsoft-365/project/project-management-software> (дата звернення 18 березня 2024).
7. Primavera P6 Enterprise Project Portfolio Management. URL: <https://www.oracle.com/construction-engineering/primavera-p6/> (дата звернення 18 березня 2024).
8. Flexible modern work management. URL: <https://www.smartsheet.com/platform> (дата звернення 18 березня 2024).
9. Atlassian Trello. URL: <https://trello.com/tour> (дата звернення 18 березня 2024).
10. The only work management platform built for scale. URL: <https://asana.com/product> (дата звернення 18 березня 2024).

11. Atlassian Jira. URL: <https://www.atlassian.com/software/jira/guides/getting-started/introduction> (дата звернення 18 березня 2024).
12. Basecamp: Features. URL: <https://basecamp.com/features> (дата звернення 18 березня 2024).
13. Google Cloud overview. URL: <https://cloud.google.com/docs/overview> (дата звернення 18 березня 2024).
14. Cloud computing with AWS. URL: <https://aws.amazon.com/what-is-aws/> (дата звернення 18 березня 2024).
15. What is Azure?. URL: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure> (дата звернення 18 березня 2024).
16. What is the IBM Cloud platform? URL: <https://cloud.ibm.com/docs/overview?topic=overview-what-is-platform> (дата звернення 18 березня 2024).
17. Oracle Cloud Infrastructure (OCI). URL: <https://www.oracle.com/cloud/> (дата звернення 18 березня 2024).
18. Google Cloud. Cloud data warehouse to power your data-driven innovation. URL: <https://cloud.google.com/bigquery> (дата звернення 18 березня 2024).
19. Google Cloud. Firestore. URL: <https://cloud.google.com/firestore> (дата звернення 18 березня 2024).
20. Object storage for companies of all sizes. URL: <https://cloud.google.com/storage> (дата звернення 18 березня 2024).
21. Google Cloud. App Engine. URL: <https://cloud.google.com/appengine> (дата звернення 18 березня 2024).
22. Google Cloud. Virtual machines for any workload. URL: <https://cloud.google.com/products/compute> (дата звернення 18 березня 2024).
23. Google Cloud. Build applications or websites quickly on a fully managed platform. URL: <https://cloud.google.com/run> (дата звернення 18 березня 2024).

24. Google Cloud. Cloud Functions. URL: <https://cloud.google.com/functions> (дата звернення 18 березня 2024).
25. Hypertext transfer protocol version 2 (HTTP/2) / М. Belshe et al. Internet Engineering Task Force (IETF). URL: <https://datatracker.ietf.org/doc/html/rfc7540> (дата звернення 8 травня 2024).
26. Google Inc. What is pub/sub?. Google Cloud. URL: <https://cloud.google.com/pubsub/docs/overview> (дата звернення 8 травня 2024).
27. Cloud SQL overview | Cloud SQL Documentation | Google Cloud. Google Cloud. URL: <https://cloud.google.com/sql/docs/introduction> (date of access: 8 травня 2024).
28. Dataflow documentation | Google Cloud. Google Cloud. URL: <https://cloud.google.com/dataflow/docs> (date of access: 8 травня 2024).
29. Cloud Tasks documentation | Cloud Tasks Documentation | Google Cloud. Google Cloud. URL: <https://cloud.google.com/tasks/docs> (дата звернення 8 травня 2024).
30. Git Hub | PERT-based-parametric-estimation-BE. URL: <https://github.com/that-viktor/PERT-bases-parametric-estimation-BE> (дата звернення 12 травня 2024).
31. Java Web Development – Javatpoint. www.javatpoint.com. URL: <https://www.javatpoint.com/java-web-development> (дата звернення: 12 травня 2024).
32. Spring Framework. Spring Framework 6.1.6. URL: <https://spring.io/projects/spring-framework> (дата звернення: 12 травня 2024).
33. Web MVC framework. Spring | Home. URL: <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html> (дата звернення: 12 травня 2024).
34. Spring Boot Reference Documentation. Spring | Home. URL: <https://docs.spring.io/spring-boot/docs/3.2.5/reference/htmlsingle/> (дата звернення: 12 травня 2024).

35. What is a REST API?. Red Hat.
URL: <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (дата звернення: 12 травня 2024).

36. Fowler M. Richardson Maturity Model. martinowler.com.
URL: <https://martinfowler.com/articles/richardsonMaturityModel.html> (дата звернення: 12 травня 2024).

37. Шутько В. В. ДОСЛІДЖЕННЯ МЕТОДІВ УПРАВЛІННЯ РОЗКЛАДОМ ІТ ПРОЄКТУ З МІГРАЦІЇ ІС У GOOGLE CLOUD PLATFORM. The XX International Scientific and Practical Conference «Ways of distance learning development in current conditions»: Міжнар. наук. конф., м. Мюнхен, 22 трав. 2023 р. Мюнхен, 2023. С. 422–425.

38. Шутько В. В. ПОРІВНЯННЯ ТОЧНОСТІ ОЦІНЮВАННЯ ТРИВАЛОСТІ ОПЕРАЦІЙ ІТ-ПРОЄКТУ З МІГРАЦІЇ ІС ДО ХМАРИ. *ІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ*: матеріали ХХVІІІ Міжнар. молодіж. форуму, м. Харків, 17 трав. 2024 р. Харків, 2024. С. 286–288.

39. Панфьорова І. Ю., Шутько В. В., Розробка методу параметричного оцінювання тривалості операцій ІТ-проєкту з міграції інформаційної системи до хмарної платформи // «АСУ і прилади автоматики», №180, 2024.

40. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-наукової програми «Управління проєктами в галузі інформаційних технологій») / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. –Харків: ХНУРЕ, 2024. –24 с.

41. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. –Чинний від 22.06.2015. –Київ: ДП «УкрНДНЦ», 2016. –26 с.

42. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. / Видання офіційне. – К.: ДП «УкрНДНЦ», 2016 – 20 с.