

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему «Інтелектуальна система вибору архітектури
ІТ-системи на основі методу аналізу ієрархій»

на здобуття освітнього ступеня бакалавра
зі спеціальності 124 Системний аналіз
(код, найменування спеціальності)
освітньо-професійної програми Системний аналіз
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис) Владислав ДОВГАЛЬ
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіти гр. САД-41
Владислав ДОВГАЛЬ
(Ім'я, ПРІЗВИЩЕ)

Керівник: Ігор ПАТРАКЕСЬ
канд.тех.наук, (Ім'я, ПРІЗВИЩЕ)
доцент

Рецензент: _____
(Ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій**

Кафедра Інформаційних систем і технологій
Ступінь вищої освіти Бакалавр
Спеціальність Системний аналіз
Освітньо-професійна програма Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри

інформаційних систем та технологій

_____ Каміла СТОРЧАК

«_____» _____ 2026 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**
_____ Довгаль Владислав Вадимович

(прізвище, ім'я, по-батькові)

1. Тема кваліфікаційної роботи: Ітелектуальна система вибору архітектури ІТ-системи на основі методу аналізу ієрархій

керівник кваліфікаційної роботи Ігор ПАТРАКЕСЬВ к.т.н., доцент

(ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджено наказом Державного університету інформаційно-комунікаційних технологій від 20 лютого 2026 р. №51

2. Строк подання студентом роботи «01» червня 2026 р.

3. Вихідні дані до роботи:

а) принципи побудови та функціонування інформаційних систем і їх архітектур.

б) метод аналізу ієрархій (АНР) як інструмент багатокритеріального прийняття рішень.

в) науково-технічна література, стандарти (ISO/IEC 25010), сучасні підходи до оцінювання якості ІТ-архітектур.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз основних архітектур ІТ-систем та їх атрибутів якості

2. Дослідження методів багатокритеріального вибору (АНР, TOPSIS, ELECTRE, PROMETHEE).
3. Розроблення математичної моделі вибору ІТ-архітектури на основі методу аналізу ієрархій
4. Практичне застосування методу для вибору архітектури інформаційної системи моніторингу та обробки заявок громадян
5. Перелік графічного матеріалу: презентація
6. Дата видачі завдання «20» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	24.02-03.03.26	
2.	Аналіз основних архітектур ІТ-систем та їх атрибутів якості.	04.03-17.03.26	
3.	Дослідження методів багатокритеріального вибору (АНР, TOPSIS, ELECTRE).	18.03-31.03.26	
4.	Розробка математичної моделі вибору ІТ-архітектури на основі методу аналізу ієрархій.	01.04-12.04.26	
5.	Практичне застосування методу для вибору архітектури інформаційної системи моніторингу та обробки заявок громадян.	12.04-24.04.26	
6.	Висновки по роботі	25.04-16.05.26	
7.	Розробка демонстраційних матеріалів, доповідь.	19.05-20.05.26	
8.	Оформлення бакалаврської роботи	21.05-30.05.26	

Здобувач вищої освіти _____
(підпис)

Владислав ДОВГАЛЬ _____
(ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи _____
(підпис)

Ігор ПАТРАКЕСЬ _____
(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступня бакалавр: 79 стор., 17 рис., 2 таблиці, 30 джерел.

Мета роботи – розроблення методичного підходу до обґрунтованого вибору ІТ-архітектури інформаційної системи на основі методу аналізу ієрархій.

Об'єкт дослідження – процес проектування інформаційних систем.

Предмет дослідження – методи та засоби багатокритеріального вибору ІТ-архітектури.

У роботі проведено системний аналіз сучасних архітектур інформаційних систем. Визначено їх особливості, переваги та недоліки, а також досліджено атрибути якості відповідно стандарту ISO/IEC 25010. Проаналізовано методи багатокритеріального прийняття рішень, зокрема метод зваженої суми, АНР, TOPSIS, ELECTRE та PROMETHEE. Обґрунтовано доцільність застосування методу аналізу ієрархій як інструменту формалізації задачі вибору архітектури. Розроблено математичну модель ієрархічного багатокритеріального вибору ІТ-архітектури, що включає побудову ієрархії критеріїв та альтернатив, формування матриць попарних порівнянь і обчислення вагових коефіцієнтів. У практичній частині роботи застосовано запропонований підхід для вибору оптимальної архітектури інформаційної системи моніторингу та обробки заявок громадян. Отримані результати дозволили обґрунтувати вибір архітектурного рішення з урахуванням множини критеріїв якості.

ІТ-АРХІТЕКТУРА, МЕТОД АНАЛІЗУ ІЄРАРХІЙ, АНР,
БАГАТОКРИТЕРІАЛЬНИЙ ВИБІР, АТРИБУТИ ЯКОСТІ, МІКРОСЕРВІСНА
АРХІТЕКТУРА

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ АРХІТЕКТУР ІТ-СИСТЕМ ТА МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ.....	10
1.1 Основні поняття та класифікація ІТ-систем.....	10
1.2 Призначення та основні види архітектур ІТ-систем	14
архітектури інформаційних систем.....	16
1.3 Поняття атрибуту якості архітектури ІТ-системи	21
1.4 Прийняття рішень в завданнях вибору архітектури.....	24
РОЗДІЛ 2. МОДЕЛЮВАННЯ ПРОЦЕСУ ВИБОРУ АРХІТЕКТУРИ ІТ-СИСТЕМИ НА ОСНОВІ МАІ	28
2.1 Постановка задачі вибору архітектури ІТ-системи	28
2.2 Аналіз методів багатокритеріального вибору альтернатив	29
2.2.1 Метод зваженої суми (МЗС)	30
2.2.2 Метод аналізу ієрархій (АНР/МАІ).....	31
2.2.3 Метод упорядкування переваг за подібністю до ідеального рішення (TOPSIS).....	32
2.2.4 Сукупність методів вибору та виключення альтернатив в завданнях багатокритеріального прийняття рішень. (ELECTRE).....	32
2.2.5 Метод ранжування альтернатив на основі функцій переваг (PROMETHEE)	33
2.2.6 Нечіткі методи багатокритеріального вибору (методи НБВ).....	34
2.3 Порівняльна характеристика методів	34
2.4 Побудова ієрархічної моделі завдання багатокритеріального вибору типу ІТ-архітектури.....	37
2.4.1 Загальна постановка завдання вибору ІТ-архітектури методом МАІ .	38
2.4.2 Сутність ієрархічного подання завдання вибору ІТ-архітектури	39
2.4.3 Рівні ієрархічної моделі багатокритеріального вибору типу ІТ-архітектури.....	41

2.4.4. Побудова ієрархічної моделі вибору ІТ-архітектури на основі багатокритеріального підходу	41
2.5 Математична модель ієрархічного багатокритеріального вибору ІТ-архітектури.....	43
2.5.1 Формування локальних пріоритетів на рівні критеріїв.....	44
2.5.2 Формування локальних пріоритетів на рівні альтернатив [14].....	44
2.5.3 Формування глобальних пріоритетів на рівні альтернатив.....	45
2.6 Узагальнений алгоритм побудови моделі вибору ІТ-архітектури.....	46
РОЗДІЛ 3. ВИБІР АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ МЕТОДУ АНАЛІЗУ ІЄРАРХІЙ	49
3.1. Опис та характеристика інформаційної системи.....	49
3.2. Формалізація завдання вибору ІТ-архітектури.....	55
3.3. Побудова ієрархічної моделі завдання вибору ІТ-архітектури.....	59
3.4. Формування матриць попарних порівнянь.....	61
3.4.1. Матриця попарних порівнянь критеріїв	61
3.4.2. Матриці попарних порівнянь альтернатив.....	62
3.5. Розрахунок ваг критеріїв та альтернатив вибору ІТ-архітектури.....	66
3.5.1. Ваги критеріїв.....	66
3.5.2. Локальні ваги альтернатив.....	67
3.6. Перевірка узгодженості матриць.....	68
3.7. Обчислення інтегральних оцінок альтернатив	70
3.8. Вибір оптимальної ІТ-архітектури інформаційної системи	71
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	76

ВСТУП

Актуальність теми. Сучасні організації — від комерційних підприємств до органів державного управління — функціонують в умовах постійного зростання обсягів інформації, кількості користувачів та складності бізнес-процесів. Інформаційні системи стали ключовим інструментом забезпечення їхньої діяльності, а вимоги до таких систем неухильно зростають: масштабованість, відмовостійкість, швидкість реагування, можливість тривалого супроводження та інтеграції з іншими сервісами стають критично важливими. Водночас ринок пропонує велику кількість архітектурних стилів — від традиційних монолітних до сучасних мікросервісних, подієво-орієнтованих та орієнтованих на простір даних, — кожен із яких має власні сильні та слабкі сторони [6, 7]. Помилковий вибір архітектури на ранніх етапах проєктування інформаційної системи призводить до значних додаткових витрат на її подальшу модифікацію, обмежує можливості розширення та може стати причиною неуспіху проєкту в цілому. Тому розроблення науково обґрунтованих підходів до вибору ІТ-архітектури з урахуванням множини суперечливих критеріїв є актуальною задачею, яка має як теоретичне, так і практичне значення.

Складність задачі вибору архітектури зумовлена її багатокритеріальним характером, наявністю якісних та кількісних критеріїв оцінювання, відсутністю об'єктивних числових значень для більшості характеристик архітектурних стилів та необхідністю врахування експертних суджень. Серед існуючих методів багатокритеріального аналізу метод аналізу ієрархій (MAI, Analytic Hierarchy Process), запропонований Т. Сааті, є одним із найбільш універсальних [13, 14] та широко застосовуваних, оскільки дозволяє формалізувати експертні судження, перевіряти їх внутрішню узгодженість та отримувати обґрунтовані числові оцінки альтернатив навіть за відсутності прямих кількісних показників.

Мета бакалаврської роботи — розроблення методичного підходу до обґрунтованого вибору ІТ-архітектури інформаційної системи на основі методу аналізу ієрархій з урахуванням множини якісних та кількісних критеріїв оцінювання, та його практичне застосування до задачі побудови інформаційної системи моніторингу й обробки заявок громадян.

Завдання дослідження. Для досягнення поставленої мети у роботі виконано такі завдання:

- проведено системний аналіз [24, 25] сучасних архітектур ІТ-систем, наведено їх класифікацію, визначено основні поняття та атрибути якості архітектурних рішень;

- досліджено існуючі методи багатокритеріального прийняття рішень — метод зваженої суми, метод аналізу ієрархій, TOPSIS, ELECTRE, PROMETHEE та нечіткі методи багатокритеріального вибору, — проведено їх порівняльний аналіз та обґрунтовано доцільність застосування методу аналізу ієрархій для розглядуваної задачі;

- розроблено математичну модель ієрархічного багатокритеріального вибору ІТ-архітектури, що включає формування локальних пріоритетів на рівні критеріїв та альтернатив, а також обчислення глобальних пріоритетів;

- сформульовано узагальнений алгоритм побудови ієрархічної моделі та визначено послідовність етапів розв'язання задачі вибору архітектури;

- виконано формальну постановку задачі вибору ІТ-архітектури для конкретної інформаційної системи моніторингу та обробки заявок громадян, побудовано її структурно-функціональну модель у нотації C4 та трирівневу ієрархічну модель;

- сформовано матриці попарних порівнянь критеріїв та альтернатив на основі експертних суджень, обчислено локальні та глобальні ваги, перевірено узгодженість матриць;

- на основі отриманих результатів обрано оптимальне архітектурне рішення для розглядуваної інформаційної системи, проведено аналіз чутливості та обґрунтовано прийняте рішення.

Об'єкт дослідження — проєктування інформаційних систем в умовах багатокритеріального вибору архітектурних рішень.

Предмет дослідження — методи та засоби обґрунтованого вибору ІТ-архітектури інформаційної системи на основі методу аналізу ієрархій.

Методи дослідження. У роботі застосовано методи системного аналізу — для дослідження структури та властивостей ІТ-архітектур; методи багатокритеріального прийняття рішень, зокрема метод аналізу ієрархій Т. Сааті, — для формалізації задачі та обчислення пріоритетів; методи лінійної алгебри (метод головного власного вектора) — для обчислення ваг елементів ієрархії; методи об'єктно-орієнтованого моделювання та нотація С4 — для побудови структурно-функціональних моделей інформаційної системи.

Практичне значення отриманих результатів. Розроблений методичний підхід може бути застосований проєктувальниками інформаційних систем, ІТ-архітекторами та керівниками проєктів для обґрунтованого вибору архітектурного рішення на ранніх етапах розроблення системи. Результати, отримані для конкретної інформаційної системи моніторингу та обробки заявок громадян, можуть бути використані під час проєктування реальних державних інформаційних сервісів, а також як методичний матеріал у навчальному процесі при викладанні дисциплін, пов'язаних з програмною інженерією та проєктуванням ІТ-систем.

Структура та обсяг роботи. Бакалаврська робота складається зі вступу, трьох розділів, загальних висновків та списку використаних джерел.

У *першому розділі* проведено системний аналіз архітектур ІТ-систем та методів прийняття рішень: розглянуто основні поняття та класифікацію

інформаційних систем, призначення та основні види архітектур, поняття атрибутів якості архітектури.

У *другому розділі* виконано моделювання процесу вибору архітектури ІТ-системи на основі методу аналізу ієрархій: проведено постановку задачі, аналіз методів багатокритеріального вибору [16, 18], побудовано ієрархічну та математичну моделі задачі, сформульовано узагальнений алгоритм її розв'язання.

У *третьому розділі* здійснено практичне застосування розробленого підходу до вибору архітектури конкретної інформаційної системи моніторингу та обробки заявок громадян: побудовано її структурно-функціональну модель, сформовано та розв'язано задачу вибору архітектури методом аналізу ієрархій, обґрунтовано отримане рішення.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ АРХІТЕКТУР ІТ-СИСТЕМ ТА МЕТОДІВ ПРИЙНЯТТЯ РІШЕНЬ

1.1 Основні поняття та класифікація ІТ-систем

В сучасних умовах розвитку інформаційного суспільства інформаційні технології відіграють ключову роль у забезпеченні ефективного функціонування організацій, підприємств та державних установ. Основою таких технологій виступають інформаційні системи, які забезпечують процеси обробки, зберігання та передачі даних [1, 3].

Під інформаційною системою (ІТ-системою) розуміють організовану сукупність взаємопов'язаних компонентів [2, 4], які забезпечують збір, опрацювання, зберігання та передачу інформації з метою підтримки прийняття рішень.

До складу ІТ-системи входять наступні компоненти (рис 1.1):

- апаратне забезпечення (сервери, комп'ютери, мережеве обладнання);
- програмне забезпечення (операційні системи, прикладні програми);
- інформаційні ресурси (бази даних, сховища даних);
- користувачі (оператори, аналітики, менеджери);

З точки зору системного аналізу ІТ-система розглядається як складна відкрита система, що взаємодіє із зовнішнім середовищем та має вхідні і вихідні потоки.

Класифікація ІТ-систем дозволяє впорядкувати їх різноманіття [3, 4] та визначити сферу застосування. Класифікація ІТ-систем відповідно за джерелами літератури застосовується за такими критеріями:

1. За рівнем управління:

- операційні системи (TPS) — обробка транзакцій у реальному часі;

- управлінські системи (MIS) — підтримка прийняття управлінських рішень;
- системи підтримки прийняття рішень (DSS) — аналіз складних ситуацій;
- експертні системи (ES) [5] — моделювання експертних знань.

2. За функціональним призначенням:

- ERP — системи управління ресурсами підприємства;
- CRM — системи управління взаємовідносинами з клієнтами;
- SCM — системи управління постачанням;
- BI — системи бізнес-аналітики [2, 4].

3. За архітектурною побудовою:

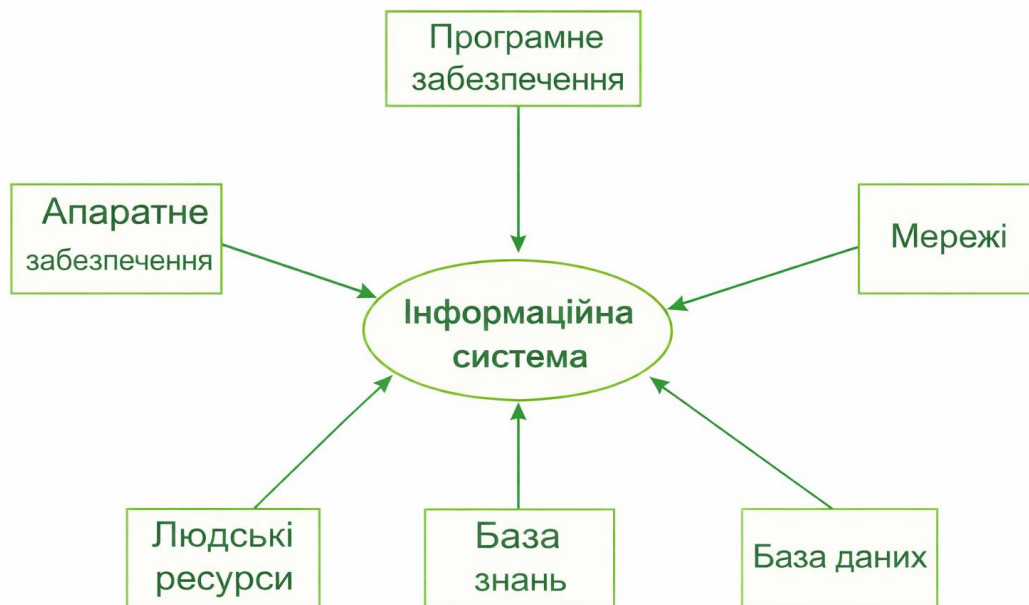


Рис. 1.1 Узагальнена структура інформаційної системи

- централізовані системи;
- клієнт-серверні системи;
- розподілені системи;
- хмарні системи.

На рис. 1.2 подано ієрархічну структуру ІТ-системи відповідно до рівнів управління в організації. Такий підхід дозволяє систематизувати ІТ-системи залежно від їх функціонального призначення та ролі у процесі прийняття управлінських рішень.

Ієрархія включає чотири основні рівні: операційний, менеджерський, тактичний та стратегічний.

1. Операційний рівень (нижній рівень управління)

На нижньому рівні знаходяться TPS (Transaction Processing Systems) — системи опрацювання транзакцій. Вони забезпечують опрацювання щоденних операцій, фіксацію первинних даних, виконання повсякденних бізнес-процесів. Такі ІТ-системи є основою функціонування організації, оскільки формують первинну інформаційну базу для всіх наступних рівнів.

2. Менеджерський рівень (середній рівень управління)

На цьому рівні використовуються MIS (Management Information Systems) — менеджерські інформаційні системи. Їх призначення складається в тому що виконується агрегування даних, формування звітності, підтримка управлінського контролю.

Менеджерський рівень забезпечує керівників середньої ланки інформацією для аналізу поточного стану системи.

3. Тактичний рівень (тактичне управління)

На тактичному рівні функціонують SPPR (системи підтримки прийняття рішень). Основними функціями є аналіз альтернатив, моделювання ситуацій, підтримка прийняття рішень в складних умовах. Таки системи використовуються для вирішення напівструктурованих задач, де необхідний аналіз великої кількості факторів.



Рис. 1.2. Ієрархія інформаційних систем за рівнем управління

Ієрархічна структура ІТ-системи відображає розподіл функцій управління в організації та забезпечує узгоджену взаємодію між різними рівнями прийняття рішень. Це створює основу для побудови ефективних ІТ-архітектур та інтегрованих інформаційних систем.

У сучасних умовах більшість ІТ-систем належать до класу складних систем, що обумовлює необхідність застосування методів системного аналізу.

Складна інформаційна система характеризується:

- великою кількістю елементів;
- багаторівневою структурою;
- наявністю нелінійних зв'язків;

- невизначеністю та стохастичністю;
- динамічністю розвитку.

З точки зору системного аналізу, така система може бути подано як:

$$S = \langle E, R, P \rangle$$

де: E — множина елементів;

R — множина зв'язків;

P — множина властивостей.

Основним завданням системного аналізу в дослідженні ІТ-систем є:

- дослідження структури системи;
- виявлення ключових елементів;
- оптимізація функціонування;
- підтримка прийняття рішень.

Таким чином, ІТ-системи виступають як складні об'єкти дослідження, для яких необхідно застосовувати формалізовані методи аналізу. ІТ-системи доцільно розглядати як об'єкти системного аналізу, що створює передумови для застосування формалізованих методів прийняття рішень в задачах вибору побудови їх архітектури.

1.2 Призначення та основні види архітектур ІТ-систем

Архітектура ІТ-системи визначає загальну структуру системи, взаємодію її компонентів та принципи організації програмного й апаратного забезпечення. Вона є основою для проєктування, розробки та подальшого розвитку інформаційної системи.

Під архітектурою ІТ-системи розуміють концептуальну модель, яка описує:

- склад компонентів системи;
- структуру взаємозв'язків між ними;

- принципи обробки даних;
- способи інтеграції та масштабування.

Основне призначення архітектури полягає у [6, 7]:

- забезпеченні ефективного функціонування системи;
- досягненні необхідних показників якості (продуктивності, надійності);
- забезпеченні можливості масштабування;
- спрощенні супроводження та модернізації.

Архітектура виступає як інструмент системного аналізу, що дозволяє формалізувати складну ІТ-систему та обґрунтувати вибір її структури.

Сучасні ІТ-системи можуть будуватися на різних архітектурних підходах, які відрізняються структурою, рівнем складності та сферою застосування.

Основні види архітектур:

- монолітна;
- клієнт-серверна;
- багатошарова(n-tier);
- мікросервісна; [28]
- розподілена;
- хмарна. [29]

На рис. 1.3 наведено порівняльну схему монолітної та мікросервісної архітектур. Монолітна архітектура є традиційним підходом до побудови інформаційних систем, при якому всі функціональні модулі — рівень представлення, бізнес-логіка та шар доступу до даних — об'єднані в єдиний нерозривний застосунок. Така побудова спрощує розробку та початкове розгортання системи, проте суттєво ускладнює її масштабування та супровід у міру зростання функціональності. Будь-яке оновлення потребує повного

перескладання та повторного розгортання всього застосунку, що збільшує ризик збоїв та час простою.

Мікросервісна архітектура [7, 8] кардинально відрізняється від монолітної: система розбивається на незалежні, слабо зв'язані сервіси, кожен з яких

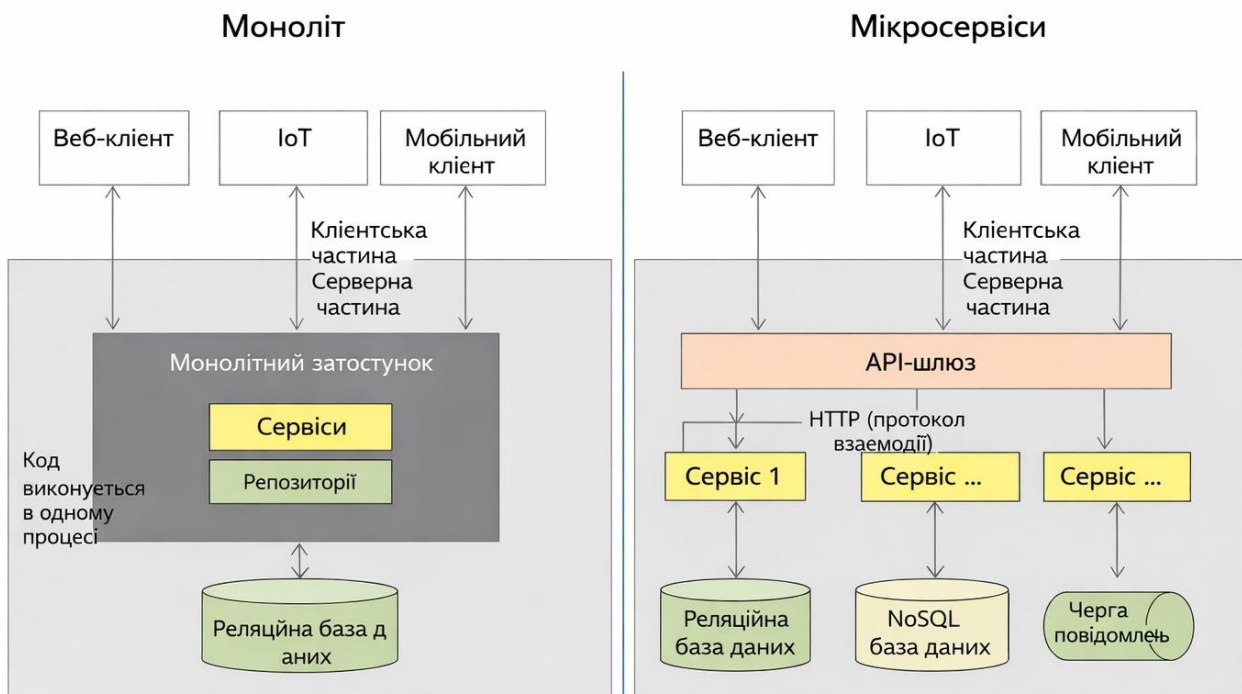


Рис. 1.3 Порівняння монолітної та мікрорсервісної архітектури інформаційних систем

виконує чітко визначену бізнес-функцію, розгортається автономно та може масштабуватися незалежно від інших. Взаємодія між сервісами здійснюється через легковагові протоколи — як правило, через REST API або асинхронний обмін повідомленнями. Мікросервісний підхід забезпечує гнучкість розробки, стійкість до відмов окремих компонентів та можливість застосування різних технологічних стеків для різних сервісів. Водночас він вимагає більш зрілої інфраструктури, ретельного управління міжсервісною взаємодією та моніторингом.

На рис. 1.4 зображено дворівневу клієнт-серверну архітектуру — одну з перших і найбільш поширених моделей розподілених систем. У цій моделі виділяють два основні рівні: клієнтський, на якому розміщується інтерфейс користувача та частина логіки застосунку, і серверний, що відповідає за зберігання даних та їх обробку. Клієнт звертається до сервера із запитом,

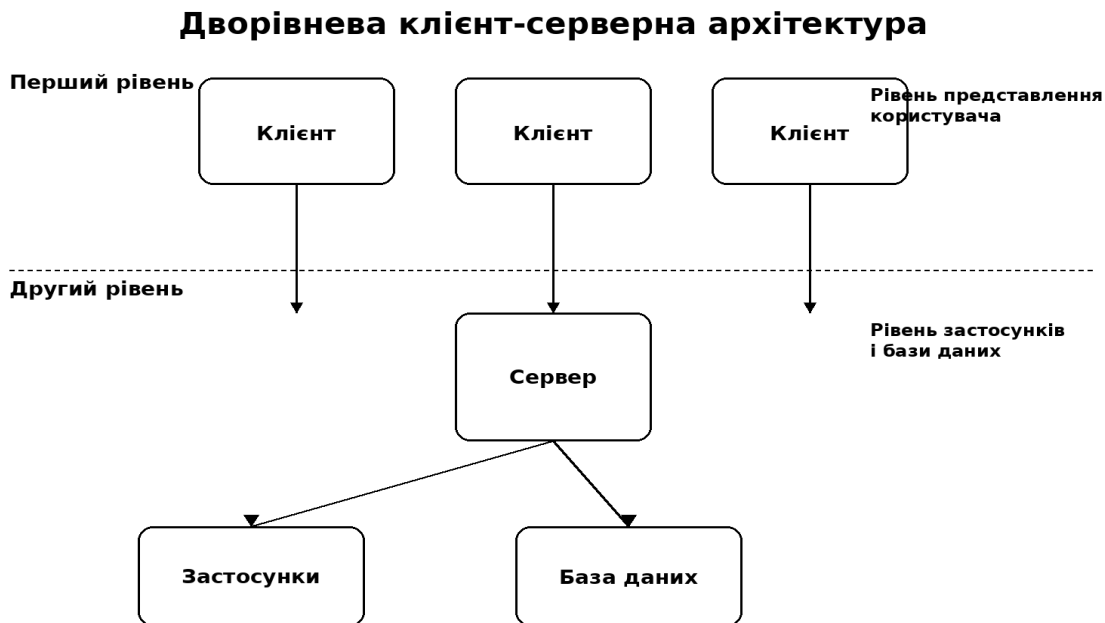


Рис. 1.4 Дворівнева клієнт-серверна архітектура

сервер опрацьовує їх та повертає результат. Дворівнева архітектура ефективна для відносно невеликих систем з обмеженою кількістю одночасних підключень. Її основний недолік — пряма залежність клієнта від сервера баз даних, що ускладнює масштабування та знижує безпеку при збільшенні навантаження.

Рис. 1.5 ілюструє багатошарову (n-tier) архітектуру, яка є логічним розвитком клієнт-серверної моделі і усуває її основні обмеження. У класичній тришаровій реалізації виокремлюють: рівень представлення (Presentation Tier), де відображається інтерфейс користувача; рівень логіки (Business Logic Tier), де

зосереджена обробка даних та реалізовані бізнес-правила; рівень даних (Data Tier), де здійснюється зберігання та управління даними у СУБД. Поділ на шари дозволяє змінювати або замінювати окремі рівні без втручання в інші, полегшує тестування компонентів, покращує безпеку та спрощує масштабування. Саме

Багатошарова архітектура (3-tier, web-система)

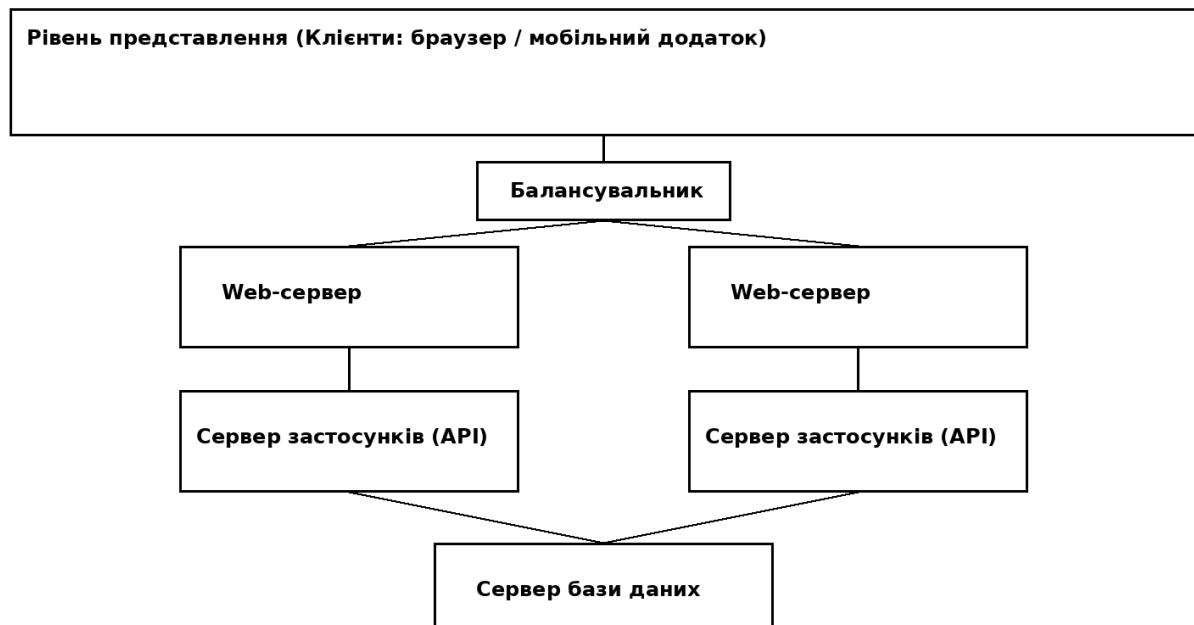


Рис. 1.5 Багатошарова ІТ-архітектура інформаційної системи

цей підхід лежить в основі переважної більшості корпоративних інформаційних систем.

На рис. 1.6 представлено розподілену архітектуру ІТ-системи, побудовану на основі мікросервісного підходу з виокремленням таких ключових компонентів, як API-шлюз, брокер повідомлень, балансувальник навантаження, реєстр сервісів та система координації (ZooKeeper). API-шлюз виступає єдиною точкою входу для клієнтських запитів і відповідає за їх маршрутизацію до відповідних мікросервісів, автентифікацію, ліміти запитів та агрегацію

відповідей. Брокер повідомлень забезпечує асинхронну комунікацію між сервісами, що підвищує стійкість системи до тимчасових збоїв окремих компонентів. Балансувальник навантаження рівномірно розподіляє вхідний трафік між екземплярами сервісів, не допускаючи перевантаження окремих

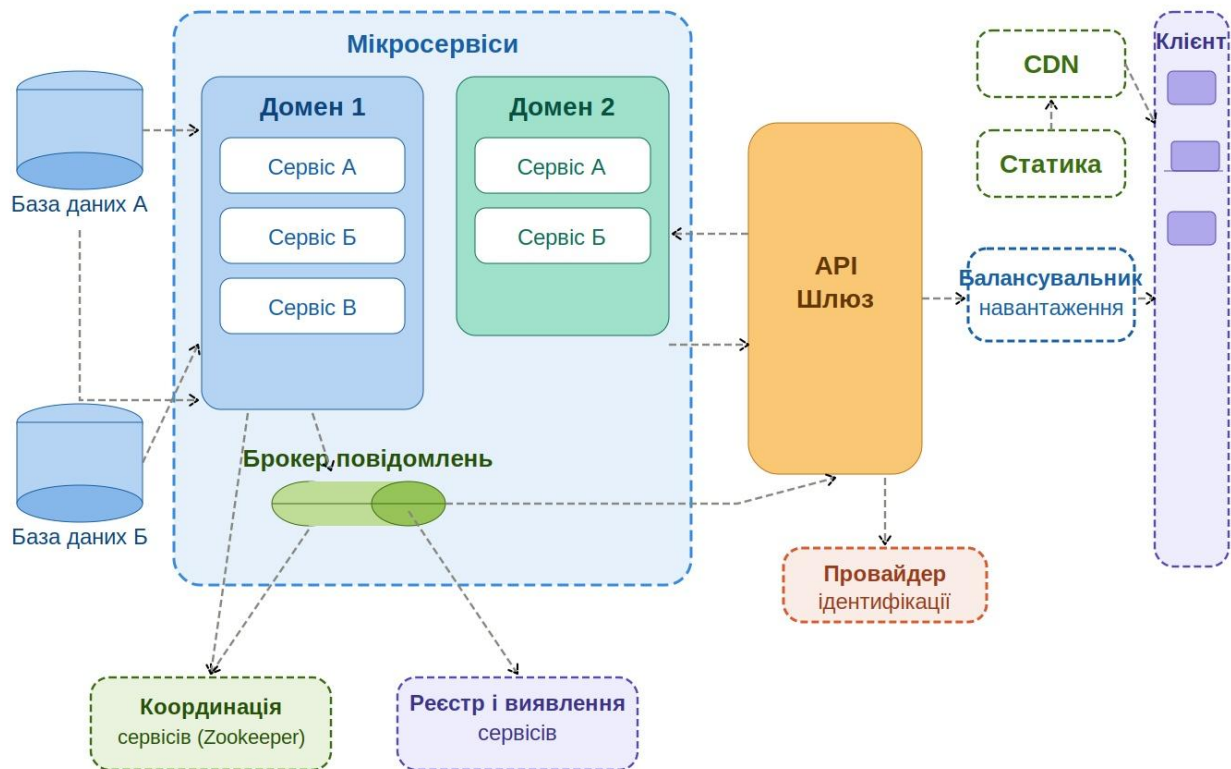


Рис. 1.6 Розподілена архітектура ІТ-системи

вузлів. Реєстр і виявлення сервісів дозволяють компонентам системи динамічно знаходити один одного в мережі без жорсткого кодування адрес. ZooKeeper забезпечує розподілену координацію та синхронізацію стану конфігурації між сервісами.

На рис. 1.7 зображено хмарну архітектуру ІТ-системи, яка описує взаємодію хмарних ресурсів з кінцевими користувачами через мережевий стек. У верхній частині схеми розташовані компоненти хмарного середовища: застосунки,

сховище даних, сервери, платформи програмного забезпечення та інфраструктура віртуальних робочих столів. Всі ці ресурси надаються через Інтернет. Нижня частина схеми відображає мережеве підключення кінцевих користувачів: маршрутизатор забезпечує Wi-Fi-доступ до Інтернету для

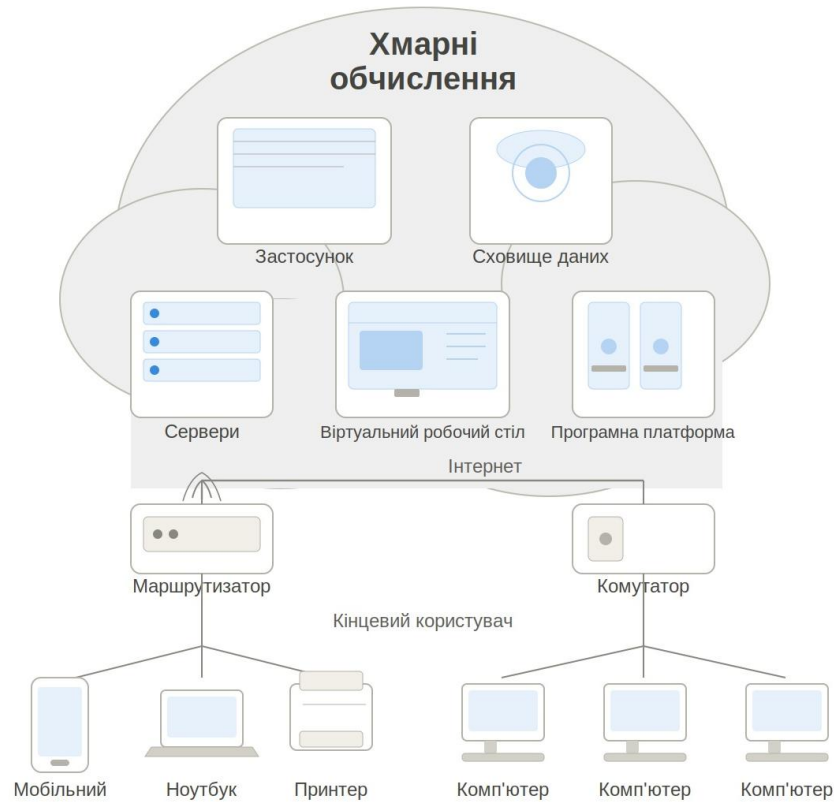


Рис. 1.7 Хмарна архітектура ІТ-системи

мобільних пристроїв, ноутбуків та принтерів; комутатор здійснює дротове підключення стаціонарних комп'ютерів. Хмарна архітектура забезпечує еластичність (можливість оперативно масштабувати ресурси залежно від навантаження), доступність з будь-якого пристрою та місця, а також переносить відповідальність за інфраструктуру на хмарного провайдера, знижуючи операційні витрати організації.

1.3 Поняття атрибуту якості архітектури ІТ-системи

Атрибут якості [6, 11] архітектури (Quality Attribute) — це вимірювана або перевіряема властивість системи, яка характеризує ступінь її відповідності вимогам зацікавлених сторін та визначає придатність обраного архітектурного рішення для конкретної предметної області. Атрибути якості не описують функціональність системи — вони визначають те, наскільки добре ця функціональність реалізована.

У сучасній інженерії програмного забезпечення атрибути якості систематизовані в низці стандартів [11, 30] та моделей. Найбільш поширеним міжнародним стандартом є ISO/IEC 25010:2011 (SQuaRE), який визначає дві основні категорії характеристик: характеристики якості продукту (Product Quality) та якості в процесі використання (Quality in Use).

З архітектурної точки зору найважливіше значення мають такі атрибути якості (рис. 1.8):

Продуктивність (Performance) — здатність системи виконувати необхідні функції у визначені часові інтервали при заданому навантаженні. Вимірюється часом відповіді (response time), пропускнуою здатністю (throughput) та використанням ресурсів. Для web-систем типовою метрикою є кількість запитів на секунду (RPS) та 95-й перцентиль часу відповіді. Архітектурні рішення, що впливають на продуктивність: кешування, балансування навантаження, асинхронна обробка, шардування баз даних.

Надійність (Reliability) — здатність системи виконувати вимагані функції в заданих умовах протягом визначеного часу. Включає підхарактеристики: зрілість (maturity), доступність (availability), відмовостійкість (fault tolerance) та відновлюваність (recoverability). Доступність зазвичай виражається у відсотках за рік (наприклад, 99,9% — не більше 8,7 годин простою на рік). Архітектурні

рішення: резервування (redundancy), автоматичне відновлення після відмов, реплікація даних, механізми Circuit Breaker.

Захищеність (Security) — здатність системи захищати інформацію та дані таким чином, щоб сторонні особи або системи мали лише той рівень доступу до даних, який відповідає їхнім повноваженням. Включає конфіденційність (confidentiality), цілісність (integrity), доступність (availability) — тріаду CIA. Архітектурні рішення: ешелонований захист (defense in depth), шифрування даних у спокої та в русі, автентифікація та авторизація, аудит дій.

Зручність супроводу (Maintainability) — характеризує ефективність, з якою система може бути модифікована для усунення дефектів, підвищення продуктивності або адаптації до змін середовища. Охоплює аналізованість, змінюваність, тестованість та модульність. Системи з низькою зв'язністю (low coupling) та високою зчепленістю (high cohesion) компонентів мають суттєво кращу супроводжуваність.

Масштабованість (Scalability) — здатність системи збільшувати або зменшувати продуктивність та ресурси відповідно до зростаючого або спадного попиту. Розрізняють вертикальне масштабування (scale up — збільшення потужності наявних вузлів) та горизонтальне масштабування (scale out — збільшення кількості вузлів). Хмарні та мікросервісні архітектури, як правило, значно краще підтримують горизонтальне масштабування, ніж монолітні.

Переносимість (Portability) — ступінь ефективності та результативності, з якою система може бути перенесена з одного апаратного, програмного або операційного середовища в інше. Для сучасних систем актуальна у контексті контейнеризації (Docker) та оркестрації (Kubernetes), які забезпечують незалежність розгортання від конкретного хмарного провайдера.

Сумісність (Compatibility) — здатність системи обмінюватися інформацією та виконувати необхідні функції при спільному використанні з іншими продуктами, системами або компонентами. Реалізується через

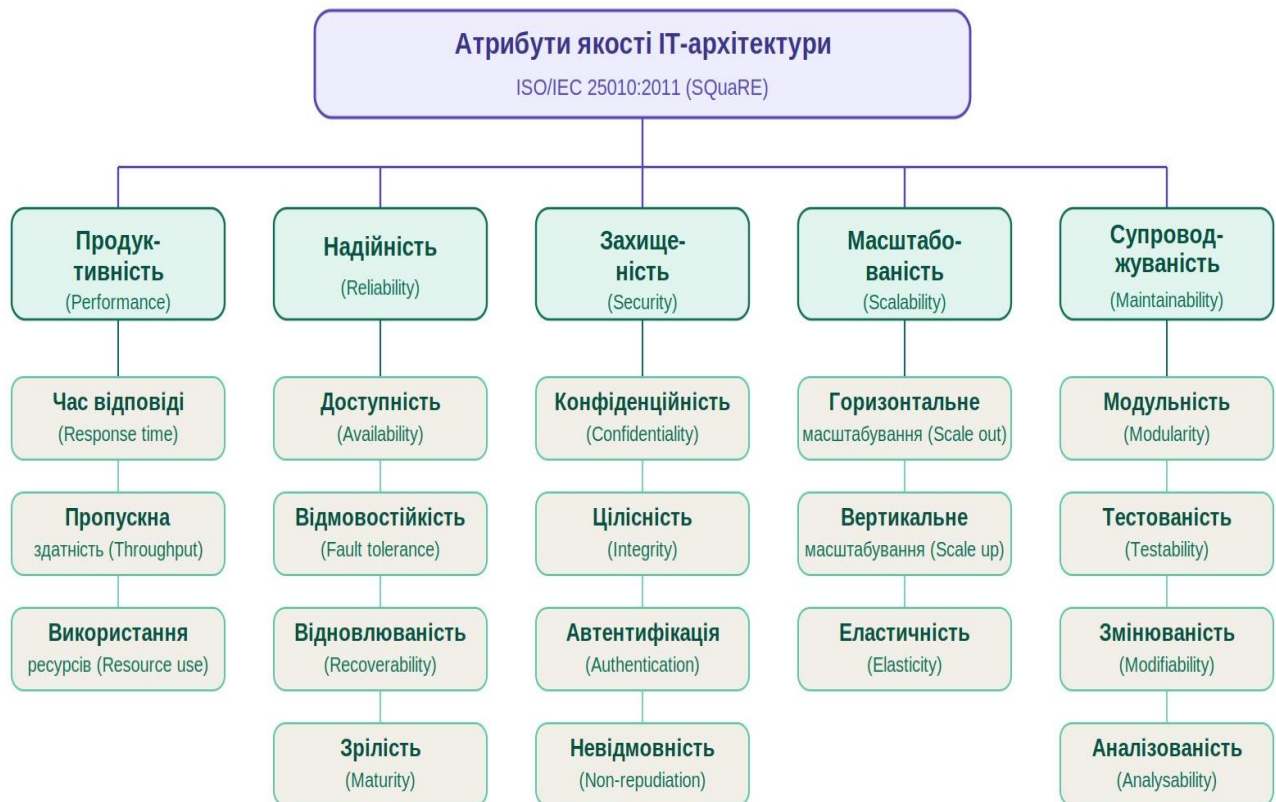


Рис. 1.8. Структурна схема атрибутів якості ІТ-архітектури відповідно до стандарту ISO/IEC 25010 :2011

стандартизовані API, протоколи обміну даними (REST, gRPC, SOAP), формати даних (JSON, XML, Protobuf) та дотримання відкритих стандартів.

З точки зору архітектурного проєктування атрибути якості нерідко перебувають у суперечності між собою. Наприклад, підвищення безпеки через шифрування може знизити продуктивність; поліпшення зручності супроводу через декомпозицію на мікросервіси ускладнює управління транзакціями та збільшує

затримки мережевої взаємодії. Тому вибір архітектури є задачею багатокритеріального прийняття рішень, де необхідно знаходити прийнятний компроміс між конкуруючими атрибутами якості відповідно до пріоритетів конкретної системи та організації.

Для формалізації процесу оцінювання архітектурних рішень з урахуванням атрибутів якості широко застосовуються спеціалізовані методи, зокрема Architecture Tradeoff Analysis Method (ATAM [12]) та Quality Attribute Workshop (QAW). Ці методи передбачають ідентифікацію ключових атрибутів якості, сценаріїв їх прояву, архітектурних рішень, що впливають на їх досягнення, а також аналіз компромісів між конкуруючими вимогами. Результатом є обґрунтована модель відповідності архітектурних рішень вимогам якості, яка може бути використана як базис для прийняття рішень щодо вибору або вдосконалення архітектури.

1.4 Прийняття рішень в завданнях вибору архітектури

Завдання вибору архітектури ІТ-системи є типовою задачею прийняття рішень в умовах багатьох критеріїв та невизначеності. Особа, що приймає рішення (ОПР), стикається з необхідністю обрати одну або кілька архітектурних альтернатив з множини допустимих, керуючись системою критеріїв, що відображають атрибути якості, обмеження проекту та стратегічні пріоритети організації.

У загальному вигляді задача вибору архітектури може бути формалізована як задача багатокритеріальної оптимізації.

Нехай $A = \{a_1, a_2, \dots, a_n\}$ — множина допустимих архітектурних альтернатив, $C = \{c_1, c_2, \dots, c_m\}$ — множина критеріїв оцінювання (атрибутів якості), $f(a_i, c_j)$ — оцінка альтернативи a_i за критерієм c_j . Тоді задача полягає у

знаходженні такої альтернативи a^* , яка є найкращою за сукупністю всіх критеріїв або забезпечує оптимальний компроміс між ними.

Для вирішення завдання вибору архітектури ІТ-систем застосовуються різні методи прийняття рішень. Основні групи методів (рис.1.9):

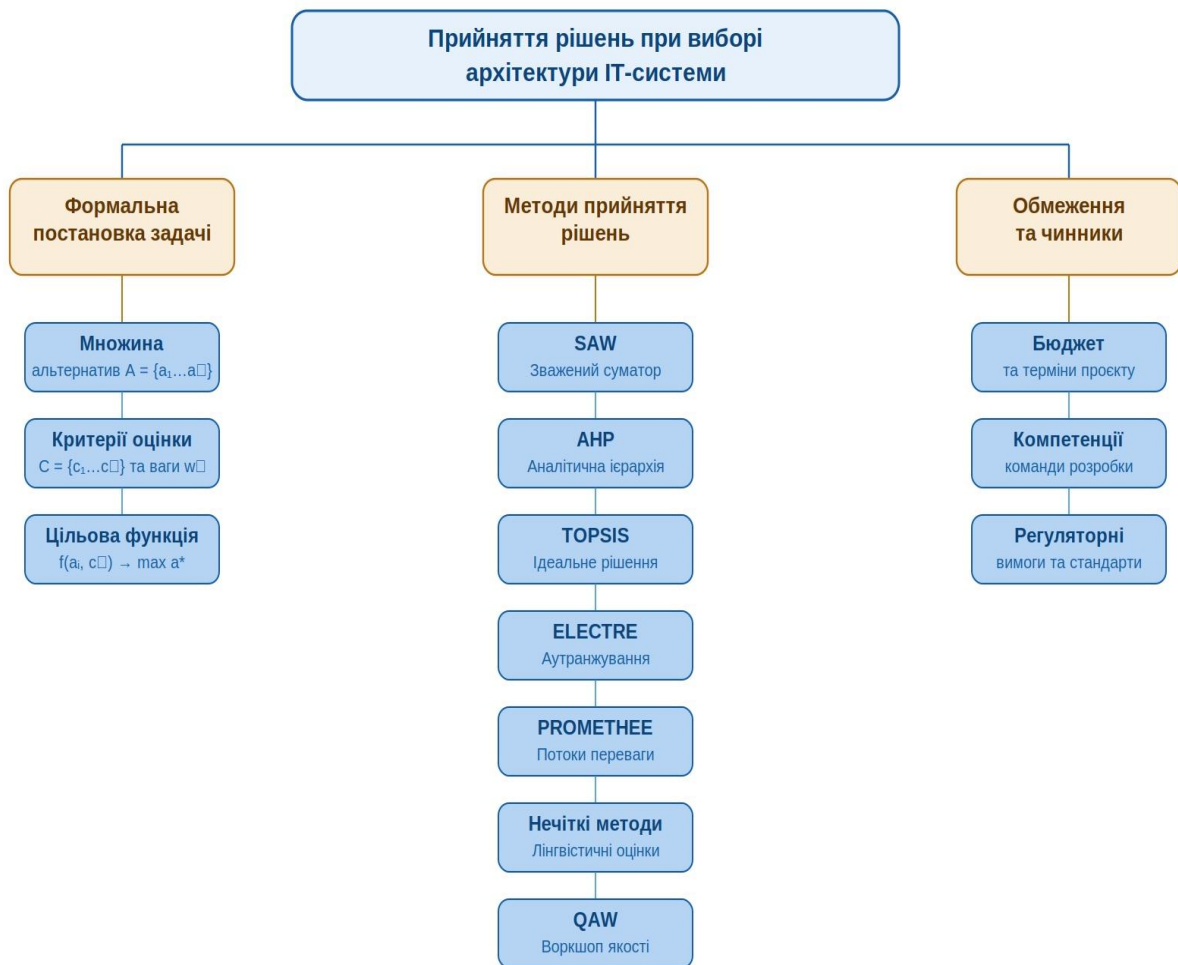


Рис. 1.9 Структурна схема прийняття рішень при виборі архітектури ІТ-системи

1. Методи зважування та агрегування (Scoring Methods). До цієї групи належать метод простого зваженого суматора (Simple Additive Weighting, SAW) та його різновиди. Кожному критерію призначається вага, що відображає його відносну важливість. Загальна оцінка альтернативи обчислюється як зважена

сума нормалізованих оцінок за окремими критеріями: $S(a_i) = \sum w_j \times f(a_i, c_j)$. Метод простий у реалізації та інтерпретації, проте передбачає компенсаторність критеріїв, тобто низька оцінка за одним критерієм може бути компенсована високою оцінкою за іншим.

2. Методи аналітичної ієрархії (Analytic Hierarchy Process, АНР). Метод АНР, розроблений Томасом Сааті, ґрунтується на попарному порівнянні як альтернатив, так і критеріїв. За результатами порівнянь будується матриця попарних порівнянь, з якої обчислюються вектори пріоритетів — відносні ваги критеріїв та відносні переваги альтернатив. Метод дозволяє перевіряти узгодженість суджень ОНР через індекс узгодженості (Consistency Index). АНР широко застосовується при виборі архітектурних рішень завдяки здатності структурувати складні ієрархічні задачі.

3. Методи переваги за ідеальним рішенням (TOPSIS). Метод TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) ранжує альтернативи за принципом мінімальної відстані від позитивно-ідеального рішення (найкращі значення за всіма критеріями) та максимальної відстані від негативно-ідеального рішення (найгірші значення). Перевагою методу є врахування геометричної близькості альтернатив до ідеалу, що дозволяє отримати більш диференційоване ранжування, ніж при простому зважуванні.

4. Методи аутранжування (ELECTRE, PROMETHEE) [22]. На відміну від компенсаторних методів, методи аутранжування не передбачають повну компенсацію критеріїв. Вони будують відносини переваги між альтернативами на основі попарних порівнянь і дозволяють виявити некомпаровані альтернативи, тобто пари, де жодна не домінує над іншою. PROMETHEE використовує функції переваги для кожного критерію та будує потоки переваги, що дозволяє отримати часткове або повне ранжування альтернатив.

5. Нечіткі методи прийняття рішень (Fuzzy Decision Making). При виборі архітектури оцінки часто мають лінгвістичний або нечіткий характер: “висока продуктивність”, “прийнятна надійність”, “помірні витрати”. Теорія нечітких множин дозволяє формалізувати такі оцінки та виконувати математичні операції над ними. Нечіткі розширення методів SAW, AHP та TOPSIS забезпечують більш адекватне відображення реальної невизначеності в задачах вибору архітектури.

Для вибору архітектури ІТ-систем у практиці програмної інженерії також широко застосовується метод Architecture Tradeoff Analysis Method (ATAM). ATAM передбачає: визначення бізнес-цілей та атрибутів якості; опис архітектурних підходів та тактик; ідентифікацію ризиків, невирішених питань та компромісів; оцінювання відповідності архітектури сценаріям використання. Метод є якісним та заснованим на структурованому обговоренні з залученням усіх зацікавлених сторін, що дозволяє досягти консенсусу щодо архітектурних рішень.

У задачах вибору архітектури важливим є також урахування обмежень та вимог нефункціонального характеру: бюджетних обмежень, наявних технологічних компетенцій команди, термінів реалізації, вимог регуляторного середовища (захист персональних даних, галузеві стандарти). Ці обмеження звужують простір допустимих рішень та впливають на ваги критеріїв при оцінюванні.

Таким чином, прийняття рішень у задачах вибору архітектури ІТ-систем є складним міждисциплінарним процесом, що поєднує методи системного аналізу, теорії прийняття рішень та інженерії програмного забезпечення. Застосування формалізованих методів оцінювання дозволяє підвищити обґрунтованість та відтворюваність архітектурних рішень, знизити вплив суб’єктивних чинників та забезпечити відповідність обраної архітектури вимогам якості та стратегічним цілям організації.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ПРОЦЕСУ ВИБОРУ АРХІТЕКТУРИ ІТ-СИСТЕМИ НА ОСНОВІ МАІ

2.1 Постановка задачі вибору архітектури ІТ-системи

У процесі проєктування інформаційних систем одним із ключових етапів є вибір архітектурного рішення, яке визначає структуру системи, принципи взаємодії її компонентів та можливості подальшого розвитку. Від правильності вибору архітектури залежить ефективність функціонування системи, її продуктивність, надійність, масштабованість та зручність супроводу.

У сучасних умовах існує значна кількість архітектурних підходів, що відрізняються між собою як за структурою, так і за характеристиками якості. Це обумовлює складність прийняття рішення, оскільки жодна з архітектур не є універсальною та не забезпечує максимальні значення за всіма критеріями одночасно.

Таким чином, задача вибору архітектури ІТ-системи [16, 17] є задачею багатокритеріального прийняття рішень, у якій необхідно врахувати множину альтернатив та критеріїв оцінювання.

Формально задачу можна представити у вигляді:

$$A = \{a_1, a_2, \dots, a_n\}$$

де a_i — множина альтернатив;

$$C = \{c_1, c_2, \dots, c_m\}$$

де c_j — множина критеріїв.

В межах даної роботи розглядаються такі альтернативи ІТ-архітектур:

- монолітна архітектура;
- клієнт-серверна архітектура;
- багатошарова архітектура [7, 9];

- мікросервісна архітектура;
- хмарна архітектура.

В якості критеріїв оцінювання можуть використовуватися атрибути якості, визначені які розглянути у розділі 1:

- продуктивність;
- надійність;
- масштабованість;
- безпека;
- супроводжуваність.

Завдання полягає у виборі такої ІТ-архітектури, яка забезпечує оптимальний компроміс між зазначеними критеріями.

2.2 Аналіз методів багатокритеріального вибору альтернатив

В задачах вибору архітектури ІТ-системи особа, що приймає рішення, зазвичай має справу не з одним, а з кількома критеріями, які часто є суперечливими. Наприклад, архітектура, що забезпечує високу масштабованість, може вимагати більших витрат на впровадження та супровід, а архітектура з високим рівнем безпеки може мати нижчу продуктивність через додаткові механізми захисту. У зв'язку з цим задача вибору архітектури належить до класу задач багатокритеріального прийняття рішень.

Багатокритеріальний вибір передбачає оцінювання множини альтернатив за сукупністю критеріїв з подальшим визначенням найбільш прийняттого рішення. У такій постановці важливо не лише визначити набір критеріїв, а й обрати адекватний метод оцінювання, який дозволить урахувати специфіку предметної області, структуру задачі, доступність експертної інформації та характер вхідних даних.

До найбільш поширених методів багатокритеріального вибору належать метод зваженої суми, метод аналізу ієрархій, метод TOPSIS, методи ELECTRE і PROMETHEE, а також нечіткі модифікації класичних підходів. Кожен із цих методів має власну логіку, переваги, обмеження та сферу доцільного застосування.

2.2.1 Метод зваженої суми (МЗС)

Метод зваженої суми [18, 19] (МЗС) є одним із найпростіших підходів до багатокритеріального вибору. Його сутність полягає у формуванні інтегральної оцінки альтернативи як зваженої суми оцінок за окремими критеріями:

$$S(a_i) = \sum_{j=1}^m w_j \cdot r_{ij},$$

де w_j — вага j -го критерію,

r_{ij} — нормалізована оцінка i -ї альтернативи за j -м критерієм.

Перевагою цього методу є його простота, наочність та легкість програмної реалізації. Він добре підходить для задач, у яких критерії можна подати у числовому вигляді, а вплив кожного критерію на підсумковий результат є лінійним.

Разом з тим метод має і суттєві обмеження. Основний недолік полягає в компенсаторному характері агрегування: низьке значення одного критерію може бути повністю компенсоване високими значеннями інших критеріїв. У реальних задачах вибору архітектури це не завжди є припустимим, оскільки критично низька надійність або безпека не повинні «перекриватися» лише хорошою масштабованістю чи продуктивністю.

Для задачі вибору архітектури ІТ-систем МЗС може використовуватися як базовий інструмент попереднього ранжування, але його аналітичні можливості є

обмеженими, особливо якщо критерії мають ієрархічну структуру або оцінки формуються експертним шляхом.

2.2.2 Метод аналізу ієрархій (АНР/МАІ)

Метод аналізу ієрархій [13, 14], запропонований Томасом Сааті, належить до найбільш відомих і практично значущих методів багатокритеріального прийняття рішень. Його ключова перевага полягає в тому, що складна задача подається у вигляді ієрархії, яка складається з мети, критеріїв і альтернатив.

Основу методу становить попарне порівняння елементів на кожному рівні ієрархії. Для цього використовується шкала відносної важливості Сааті, що дозволяє переводити якісні експертні судження в кількісну форму. Далі на основі матриць попарних порівнянь обчислюються вектори пріоритетів, які відображають ваги критеріїв та локальні пріоритети альтернатив.

Важливою перевагою АНР є можливість перевірки узгодженості експертних суджень за допомогою індексу та відношення узгодженості. Це особливо цінно в задачах, де рішення значною мірою спирається на експертну оцінку, а не лише на суто числові вимірювання.

Для задачі вибору архітектури ІТ-систем метод аналізу ієрархій є доцільним з таких причин:

1. Метод дає змогу формалізувати ієрархічну природу задачі;
2. Метод дозволяє працювати з критеріями різної природи;
3. Метод забезпечує прозорість і логічну обґрунтованість результату.

Саме тому цей метод обрано як основний в межах даної роботи.

2.2.3 Метод упорядкування переваг за подібністю до ідеального рішення (TOPSIS)

Метод TOPSIS [19, 20] ґрунтується на припущенні, що найкраща альтернатива повинна бути максимально наближеною до ідеального рішення та максимально віддаленою від антиідеального. Ідеальне рішення містить найкращі значення за всіма критеріями, тоді як антиідеальне — найгірші.

Алгоритм методу включає нормалізацію матриці оцінок, зважування критеріїв, визначення позитивного та негативного ідеальних рішень, а також обчислення відстаней кожної альтернативи до цих точок. Остаточне ранжування здійснюється за коефіцієнтом близькості до ідеалу.

Перевагою методу TOPSIS є його математична чіткість та здатність формувати досить диференційоване ранжування альтернатив. Метод добре працює з кількісними даними та є зручним у реалізації.

Водночас метод менш зручний у випадку, якщо:

- критерії мають виражений експертний характер;
- необхідно моделювати ієрархічну структуру;
- важливо здійснювати перевірку внутрішньої узгодженості суджень.

У завданні вибору ІТ-архітектури метод TOPSIS може дати корисний результат, особливо за наявності числових метрик, проте для пояснювальної, ієрархічної і експертної задачі МАІ є більш природним.

2.2.4 Сукупність методів вибору та виключення альтернатив в завданнях багатокритеріального прийняття рішень. (ELECTRE)

Сімейство методів ELECTRE [21] належить до методів аутранжування. Їхня ідея полягає не у формуванні однієї інтегральної оцінки, а у встановленні

відношень переваги між альтернативами. Тобто визначається, чи можна вважати одну альтернативу принаймні не гіршою за іншу за сукупністю критеріїв.

Перевага таких методів полягає в тому, що вони дозволяють уникнути надмірної компенсаторності. Це означає, що критично низька оцінка за одним критерієм не буде автоматично перекрита високими оцінками за іншими критеріями. У ряді практичних задач це є дуже важливим.

Разом з тим ELECTRE має і недоліки:

- відносно складна процедура налаштування;
- необхідність вибору порогів згоди та незгоди;
- складніша інтерпретація результатів порівняно з МАІ або МЗС.

Для бакалаврської роботи ELECTRE доцільно розглядати як перспективний метод для складних рішень, але не як основний, оскільки він ускладнює як математичний апарат, так і пояснення результату.

2.2.5 Метод ранжування альтернатив на основі функцій переваг (PROMETHEE)

Метод PROMETHEE, як і метод ELECTRE, належать до методів аутранжування. Його особливістю є використання функцій переваги для кожного критерію та побудова позитивного і негативного потоків переваги. Це дозволяє ранжувати альтернативи за ступенем їх домінування над іншими.

Перевагами методу є:

- достатньо висока гнучкість;
- зручне ранжування альтернатив;
- добрі можливості для аналітичної інтерпретації.

Разом з тим метод потребує попереднього налаштування типів функцій переваги та їх параметрів, що ускладнює його застосування в умовах обмеженого часу або недостатньої статистичної інформації.

У контексті вибору архітектури ІТ-систем PROMETHEE є досить потужним інструментом, однак для дипломної роботи бакалаврського рівня він поступається MAI за простотою застосування.

2.2.6 Нечіткі методи багатокритеріального вибору (методи НБВ)

У багатьох практичних ситуаціях оцінювання альтернатив виконується не у формі точних числових значень, а за допомогою лінгвістичних оцінок, таких як «висока надійність», «помірна вартість», «добра масштабованість». Для роботи з такими оцінками застосовуються нечіткі методи, зокрема Fuzzy MAI [23], Fuzzy TOPSIS та інші.

Перевага нечітких підходів полягає в можливості урахування невизначеності, нечіткості та суб'єктивності експертних оцінок. Це особливо актуально в архітектурних задачах, де не всі критерії можна виміряти строго формально.

Однак використання нечітких методів призводить до ускладнення математичного апарату та програмної реалізації.

2.3 Порівняльна характеристика методів

Для систематизації результатів аналізу доцільно провести порівняння основних методів за рядом характеристик, що мають значення для завдання вибору ІТ-архітектури [18, 27].

Таблиця 2.1

Порівняння методів багатокритеріального вибору

Метод	Простота реалізації	Робота з ієрархією	Перевірка узгодженості [13, 26]	Робота з експертним и оцінками	Подання результату	Доцільність для вибору архітектури
1	2	3	4	5	6	7
МЗС	висока	низька	ні	середня	висока	середня
MAI	висока	висока	так	висока	висока	дуже висока
TOPSIS	середня	низька	ні	середня	середня	висока
ELECTRE	низька	низька	ні	середня	низька	середня
PROMET HEE	середня	низька	ні	середня	середня	середня
Fuzzy ANP	середня	висока	частково	дуже висока	середня	дуже висока

Таблиця 2.2

Переваги та недоліки основних методів

Метод	Основні переваги	Основні недоліки
1	2	3
МЗС	Простота, швидкість обчислень, наочність	Компенсаторність критеріїв, слабка робота зі складною структурою
MAI	Ієрархічність, робота з експертними оцінками, перевірка узгодженості	Значна кількість попарних порівнянь при великій кількості елементів
TOPSIS	Орієнтація на ідеальне рішення, чітке ранжування	Менша наочність для експертів, відсутність перевірки узгодженості

ELECTRE	Можливість урахування некомпенсаторності	Складність налаштування та інтерпретації
PROMETHEE	Гнучкість, побудова потоків переваги	Потрібне задання функцій переваги та параметрів
Fuzzy AHP	Урахування нечіткості оцінок, адаптивність	Ускладнений математичний апарат

Для наочного відображення результатів аналізу побудовано графік узагальненої експертної оцінки придатності методів до задачі вибору архітектури ІТ-системи.

Подані значення на рис. 2.1. є узагальненою експертною оцінкою за п'ятибальною шкалою, сформованою з урахуванням таких ознак: зручність використання, можливість роботи з ієрархією критеріїв, придатність до експертних оцінок, пояснюваність результату та складність практичної реалізації.

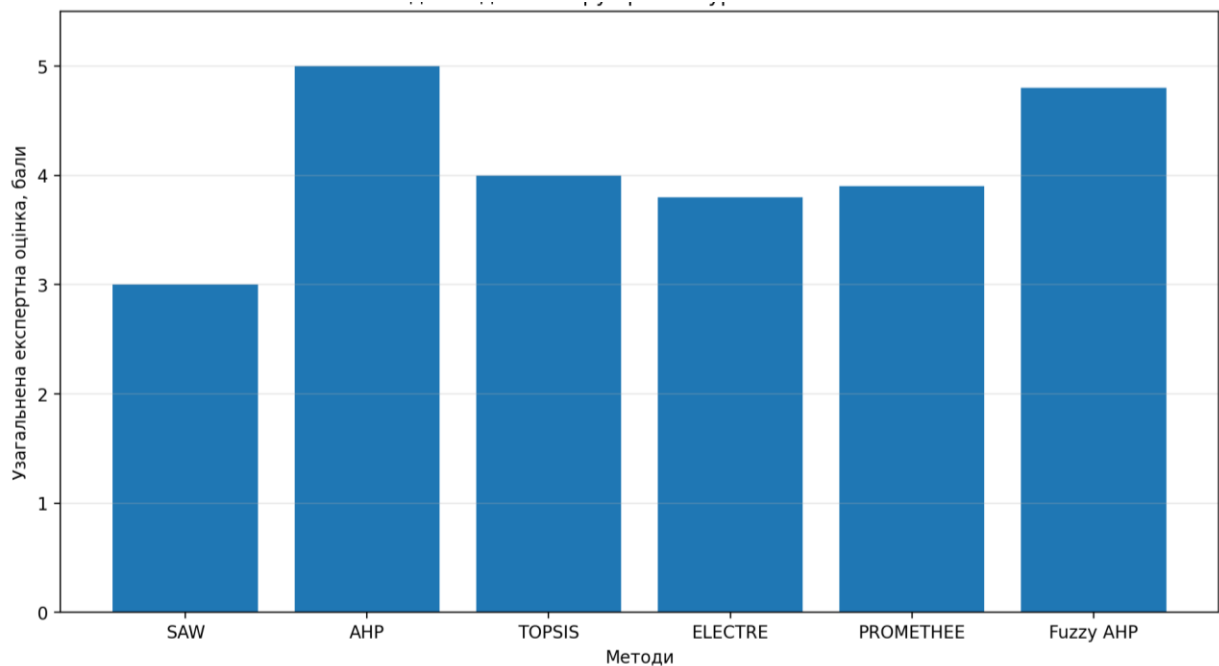


Рис. 2.1 Порівняльна придатність методів багатокритеріального вибору для задачі вибору архітектури ІТ- архітектури

За результатами узагальненої оцінки найбільш придатними для задачі вибору ІТ- архітектури є МАІ та його нечітка модифікація Fuzzy МАІ. Це пояснюється тим, що зазначені методи найкраще відповідають природі досліджуваної задачі, яка має ієрархічну структуру, значну частку експертних оцінок та потребує прозорого пояснення отриманого результату.

На основі проведеного аналізу можна дійти висновку, що для задачі вибору ІТ- архітектури найбільш доцільним є застосування методу аналізу ієрархій.

Такий вибір пояснюється сукупністю його властивостей. По-перше, метод дозволяє представити задачу у вигляді логічно зрозумілої ієрархії, що відповідає структурі досліджуваної проблеми. По-друге, він дає змогу використовувати експертні судження там, де відсутні точні кількісні оцінки. По-третє, метод містить механізм перевірки узгодженості суджень, що підвищує обґрунтованість результату.

Крім того, АНР є достатньо простим для програмної реалізації, що є важливим з огляду на подальшу розробку прототипу системи підтримки вибору архітектури у третьому розділі роботи.

Таким чином, саме метод аналізу ієрархій приймається в даній роботі як основний інструмент багатокритеріального оцінювання альтернатив архітектур ІТ-систем.

2.4 Побудова ієрархічної моделі завдання багатокритеріального вибору типу ІТ-архітектури

Одним із ключових етапів застосування методу аналізу ієрархій є побудова ієрархічної моделі задачі. Саме на цьому етапі складна задача вибору ІТ- архітектури перетворюється на впорядковану багаторівневу структуру, яка

дозволяє послідовно оцінити відносну важливість критеріїв і переваги альтернатив. Ієрархічне подання є зручним, оскільки воно забезпечує декомпозицію загальної задачі на простіші складові, кожна з яких може бути проаналізована окремо.

В межах даної роботи задача вибору типу ІТ-архітектури розглядається як задача багатокритеріального прийняття рішень, у якій існує множина можливих архітектурних альтернатив та множина критеріїв, за якими ці альтернативи порівнюються. Метою є визначення такої архітектури, яка найкраще відповідає встановленим вимогам [13, 15].

2.4.1 Загальна постановка завдання вибору ІТ-архітектури методом МАІ

Нехай задано множину альтернатив:

$$A = \{a_1, a_2, \dots, a_n\},$$

де a_i — i -та архітектурна альтернатива.

У завдання вибору типу ІТ-архітектури до множини альтернатив можуть входити:

$$A = \{a_1, a_2, a_3, a_4, a_5\},$$

де:

a_1 — монолітна архітектура;

a_2 — клієнт-серверна архітектура;

a_3 — багатошарова архітектура;

a_4 — мікросервісна архітектура;

a_5 — хмарна архітектура.

Також задано множину критеріїв оцінювання:

$$C = \{c_1, c_2, \dots, c_m\},$$

де c_j — j -й критерій.

В межах даної роботи доцільно використовувати такі критерії:

$$C = \{c_1, c_2, c_3, c_4, c_5\},$$

де:

c_1 — продуктивність;

c_2 — надійність;

c_3 — масштабованість;

c_4 — безпека;

c_5 — супроводжуваність.

Тоді загальна задача полягає у знаходженні альтернативи

$$a^* \in A,$$

яка є найкращою за сукупністю критеріїв C .

Формально це можна подати як задачу максимізації інтегральної оцінки:

$$a^* = \arg \max_{a_i \in A} S(a_i),$$

де $S(a_i)$ — узагальнена оцінка альтернативи a_i .

Оскільки значення $S(a_i)$ не задається безпосередньо, воно визначається в результаті ієрархічного аналізу.

2.4.2 Сутність ієрархічного подання завдання вибору ІТ-архітектури

Метод аналізу ієрархій [15, 26] ґрунтується на припущенні, що складну проблему можна декомпонувати у вигляді ієрархії. Ієрархія — це багаторівнева структура, елементи якої пов'язані відношенням підпорядкованості: верхній рівень відображає загальну мету, середні рівні — критерії та, за потреби, підкритерії, а нижній рівень — альтернативи.

У загальному вигляді ієрархічна модель має вигляд (рис.2.2):

$$H = \langle G, C, A \rangle,$$

де:

G — ціль прийняття рішення;

C — множина критеріїв;

A — множина альтернатив.

Для задачі вибору ІТ-архітектури:

$$H = \langle G, \{c_1, c_2, c_3, c_4, c_5\}, \{a_1, a_2, a_3, a_4, a_5\} \rangle,$$

де G — вибір оптимального типу ІТ-архітектури.

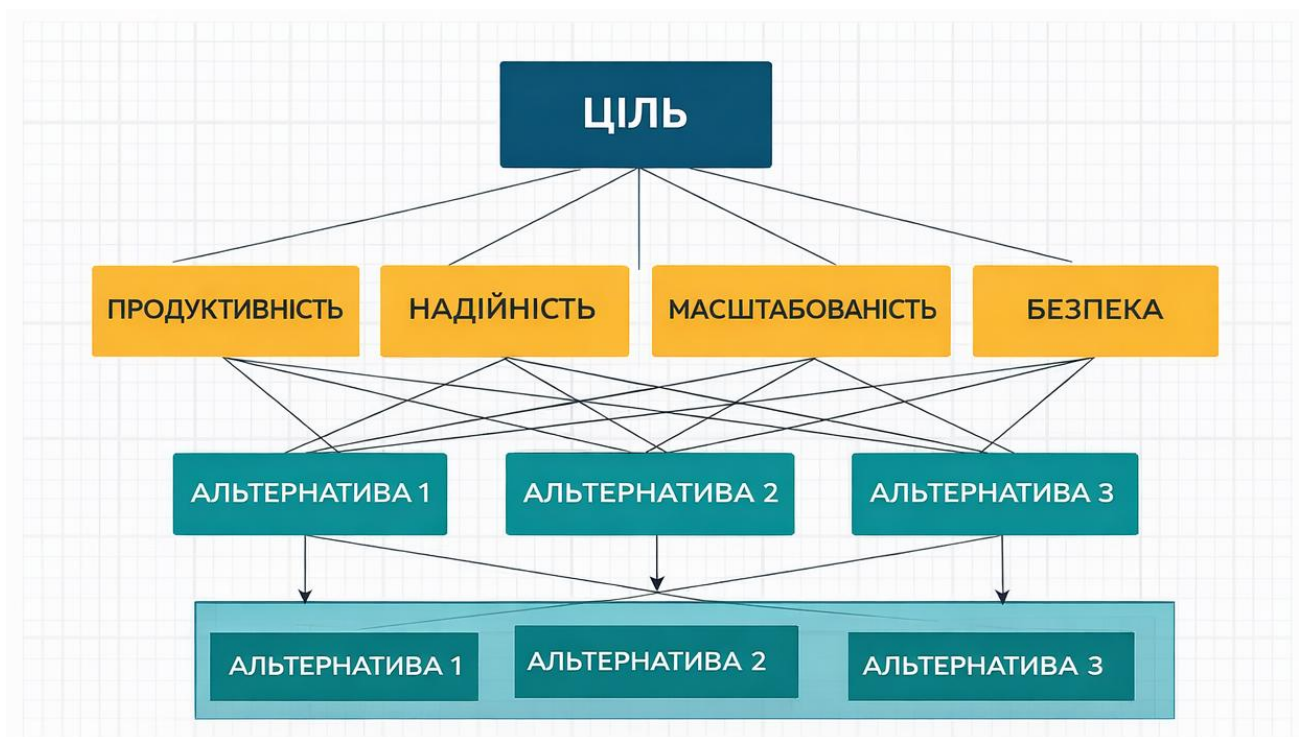


Рис. 2.2 Структура методу ієрархічного аналізу

Такий підхід дозволяє:

- розділити загальну задачу на локальні задачі порівняння;
- структуровано подати взаємозв'язок між метою, критеріями та альтернативами;
- забезпечити прозорість процедури оцінювання.

2.4.3 Рівні ієрархічної моделі багатокритеріального вибору типу ІТ-архітектури

Ієрархічна модель задачі вибору типу ІТ-архітектури доцільно будується у вигляді трирівневої структури.

Перший рівень — мета. На верхньому рівні формується основна мета дослідження:

$G = \text{«Вибір оптимального типу ІТ-архітектури»}.$

Цей рівень є єдиним і визначає загальну спрямованість задачі.

Другий рівень — критерії. Другий рівень утворюють критерії оцінювання, які відображають найбільш суттєві властивості архітектурних рішень. Формально цей рівень задається множиною:

$$C = \{c_1, c_2, \dots, c_m\}.$$

Кожний критерій є підлеглим елементом відносно мети G , оскільки характеризує один із аспектів досягнення цілі.

Третій рівень — альтернативи. Нижній рівень складають альтернативні типи архітектур:

$$A = \{a_1, a_2, \dots, a_n\}.$$

Кожна альтернатива порівнюється відносно кожного критерію другого рівня.

Таким чином, структура моделі може бути подана як:

$$G \rightarrow C \rightarrow A.$$

2.4.4. Побудова ієрархічної моделі вибору ІТ-архітектури на основі багатокритеріального підходу

Побудова ієрархічної моделі здійснюється відповідно до послідовності кроків.

Крок 1. Формулювання мети. Спочатку визначається центральна проблема, яку необхідно розв’язати. У даному випадку це вибір архітектури ІТ-системи, що найбільшою мірою відповідає вимогам проєкту. Формалізовано визначення мети побудови ієрархічної моделі багатокритеріального вибору типу ІТ-архітектури:

$$G = \arg \text{ best architecture}$$

або у змістовному вигляді:

$$G = \text{ вибір найкращої архітектурної альтернативи.}$$

Крок 2. Визначення множини альтернатив. На цьому етапі формується перелік архітектур, між якими здійснюватиметься вибір. Альтернативи мають бути:

- взаємовиключними;
- порівнюваними;
- релевантними поставленій задачі.

Для сформульованої задачі вибору ІТ-архітектури:

$$A = \{a_1, a_2, a_3, a_4, a_5\}.$$

Крок 3. Визначення критеріїв вибору ІТ-архітектури. Критерії мають відображати ті властивості архітектури, які є значущими для прийняття рішення. Важливо, щоб критерії були: суттєвими; недубльованими; зрозумілими для експертного оцінювання.

Нехай:

$$C = \{c_1, c_2, c_3, c_4, c_5\}.$$

Крок 4. Уточнення структури залежностей. На цьому етапі встановлюється, що критерії безпосередньо пов’язані з метою, а альтернативи

— з кожним критерієм. Тобто для кожного критерію c_j оцінюється внесок кожної альтернативи a_i .

Формально це означає наявність відображень:

$$\phi_j: A \rightarrow R,$$

де $\phi_j(a_i)$ — оцінка альтернативи a_i за критерієм c_j .

Крок 5. Формування ієрархічної схеми. Результатом є повна ієрархія:

- верхній вузол — мета G ;
- середній рівень — вузли критеріїв c_j ;
- нижній рівень — вузли альтернатив a_i .

2.5 Математична модель ієрархічного багатокритеріального вибору ІТ-архітектури

Ієрархічну модель можна розглядати як орієнтований граф:

$$H = (V, E),$$

де V — множина вершин;

E — множина дуг.

Множина вершин визначається як:

$$V = \{G\} \cup C \cup A.$$

Множина дуг:

$$E = \{(G, c_j)\} \cup \{(c_j, a_i)\},$$

де $j = 1, \dots, m, i = 1, \dots, n$.

Тобто між метою та критеріями існують дуги:

$$(G, c_1), (G, c_2), \dots, (G, c_m),$$

а між кожним критерієм та альтернативами:

$$(c_j, a_1), (c_j, a_2), \dots, (c_j, a_n).$$

Це означає, що оцінювання здійснюється на двох рівнях:

1. Визначення ваг критеріїв відносно мети;
2. Визначення ваг альтернатив відносно кожного критерію.

2.5.1 Формування локальних пріоритетів на рівні критеріїв

Після побудови ієрархії виконується попарне порівняння критеріїв за їх внеском у досягнення мети. Для цього будується матриця попарних порівнянь критеріїв [13, 15]:

$$A^{(C)} = (a_{ij}), i, j = 1, \dots, m,$$

де a_{ij} показує, наскільки критерій c_i за критерій c_j .

Властивості такої матриці:

$$a_{ii} = 1, a_{ij} = 1/a_{ji}.$$

На основі матриці $A^{(C)}$ визначається вектор ваг критеріїв:

$$W = (w_1, w_2, \dots, w_m)^T,$$

$$\text{де } \sum_{j=1}^m w_j = 1$$

У класичному варіанті МАІ вектор w визначається як нормований власний вектор матриці $A^{(C)}$, що відповідає найбільшому власному значенню:

$$A^{(C)}w = \lambda_{\max}w.$$

2.5.2 Формування локальних пріоритетів на рівні альтернатив [14]

Для кожного критерію c_j будується окрема матриця попарних порівнянь альтернатив:

$$A^{(j)} = (a_{pq}^{(j)}), p, q = 1, \dots, n,$$

де $a_{pq}^{(j)}$ показує, наскільки альтернатива a_p є кращою за альтернативу a_q з точки зору критерію c_j .

Для кожної такої матриці визначається вектор локальних пріоритетів:

$$\mathbf{p}^{(j)} = (p_1^{(j)}, p_2^{(j)}, \dots, p_n^{(j)})^T,$$

$$\text{де } \sum_{i=1}^n p_i^{(j)} = 1$$

Тоді елемент $p_i^{(j)}$ відображає локальний пріоритет альтернативи a_i за критерієм c_j .

2.5.3 Формування глобальних пріоритетів на рівні альтернатив

Після визначення ваг критеріїв [14, 15] та локальних пріоритетів альтернатив виконується агрегування результатів. Глобальний пріоритет альтернативи a_i визначається як зважена сума її локальних пріоритетів:

$$P_i = \sum_{j=1}^m w_j \cdot p_i^{(j)}.$$

У векторній формі:

$$\mathbf{P} = \mathbf{P}_{loc} \mathbf{w},$$

де $\mathbf{P}_{loc}$ — матриця локальних пріоритетів розмірності $n \times m$:

$$\mathbf{P}_{loc} = \begin{bmatrix} p_1^{(1)} & p_1^{(2)} & \dots & p_1^{(m)} \\ p_2^{(1)} & p_2^{(2)} & \dots & p_2^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ p_n^{(1)} & p_n^{(2)} & \dots & p_n^{(m)} \end{bmatrix}.$$

Оптимальною вважається альтернатива, для якої глобальний пріоритет максимальний:

$$a^* = \arg \max_{a_i \in A} P_i.$$

2.6 Узагальнений алгоритм побудови моделі вибору ІТ-архітектури

Узагальнений алгоритм [13, 27] побудови ієрархічної моделі ієрархічного багатокритеріального вибору ІТ-архітектури показано на рис. 2.3. Згідно з представленою блок-схемою, алгоритм реалізації методу аналізу ієрархій включає низку взаємопов'язаних етапів.

На початковому етапі здійснюється формування мети прийняття рішення, яка визначає загальну спрямованість дослідження та виступає верхнім рівнем ієрархічної моделі. В межах даної роботи метою є вибір оптимального типу ІТ-архітектури.

Далі виконується визначення множини критеріїв оцінювання та альтернатив. Критерії характеризують основні властивості архітектурних рішень, тоді як альтернативи представляють можливі варіанти реалізації ІТ-системи.

Наступним кроком є побудова ієрархічної структури, що встановлює зв'язки між метою, критеріями та альтернативами у вигляді багаторівневої моделі. Така структура дозволяє розкласти складну задачу на сукупність локальних задач порівняння.

Після формування ієрархії здійснюється побудова матриці попарних порівнянь критеріїв, на основі якої визначаються їх вагові коефіцієнти. Отримані ваги відображають відносну важливість кожного критерію у досягненні поставленої мети.

Важливим етапом є перевірка узгодженості експертних оцінок. Для цього обчислюється відношення узгодженості CR. У випадку, якщо значення CR

перевищує допустимий поріг $(0,1)$, необхідно виконати коригування матриці попарних порівнянь та повторити розрахунки. Після визначення ваг критеріїв формується система матриць попарних порівнянь альтернатив за кожним критерієм. Це дозволяє оцінити відносну перевагу альтернатив у межах окремих критеріїв.

Аналогічно до попереднього етапу виконується перевірка узгодженості для кожної матриці альтернатив. У разі невідповідності умовам узгодженості здійснюється коригування експертних оцінок.

Далі обчислюються локальні пріоритети альтернатив, які характеризують їх ефективність за кожним окремим критерієм. На основі цих значень формується матриця локальних пріоритетів.

На завершальному етапі виконується обчислення глобальних пріоритетів альтернатив шляхом агрегування локальних оцінок із урахуванням ваг критеріїв. Отримані значення використовуються для ранжування альтернатив та визначення оптимального архітектурного рішення.

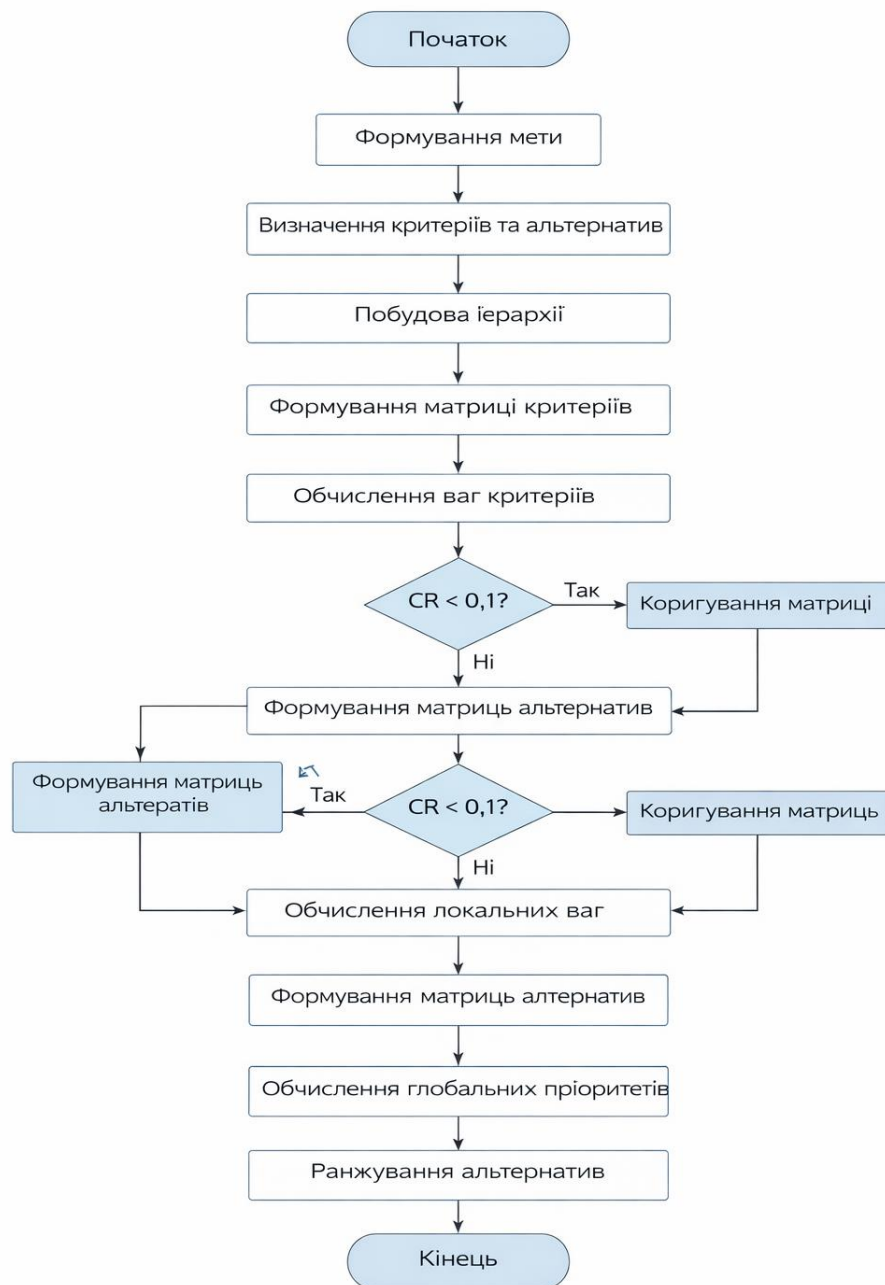


Рис. 2.3 Блок-схема алгоритму МАІ для вибору ІТ-архітектури

Таким чином, запропонований алгоритм забезпечує послідовне та обґрунтоване вирішення задачі вибору ІТ- архітектури, поєднуючи експертні оцінки з формальними математичними процедурами методу аналізу ієрархій.

РОЗДІЛ 3. ВИБІР АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ МЕТОДУ АНАЛІЗУ ІЄРАРХІЙ

3.1. Опис та характеристика інформаційної системи

Розглянемо гіпотетичну інформаційну систему [6, 9] моніторингу та обробки заявок громадян, яка призначена для автоматизації процесів приймання, реєстрації, оброблення та контролю виконання звернень громадян до органів державної влади або місцевого самоврядування. Основною метою системи є забезпечення своєчасного реагування на звернення, підвищення прозорості роботи відповідних служб та покращення якості обслуговування населення.

Система орієнтована на використання різними категоріями користувачів, зокрема громадянами, які подають заявки; операторами, що здійснюють первинну обробку звернень; виконавцями, відповідальними за їх виконання; а також керівниками, які здійснюють контроль та аналіз ефективності роботи. Доступ до функціональних можливостей системи визначається ролями користувачів та їх повноваженнями.

Функціональні можливості системи включають: реєстрацію звернень громадян через веб-інтерфейс або мобільні додатки; класифікацію заявок за типами проблем; маршрутизацію заявок до відповідальних підрозділів; контроль строків виконання; зміну статусів заявок; формування повідомлень користувачам; а також генерацію аналітичних звітів щодо навантаження та ефективності роботи служб.

З точки зору структури, інформаційна система може бути представлена у вигляді сукупності взаємопов'язаних підсистем (модулів), які реалізують окремі функціональні задачі. До основних структурних компонентів системи належать:

- модуль взаємодії з користувачами, що забезпечує введення та перегляд заявок через веб- або мобільний інтерфейс;

- модуль оброблення заявок, який відповідає за реєстрацію, класифікацію, маршрутизацію та зміну статусів звернень;
- модуль управління виконавцями, що забезпечує призначення відповідальних осіб та контроль їх діяльності;
- модуль повідомлень, який реалізує механізми інформування користувачів про стан оброблення заявок;
- аналітичний модуль, що забезпечує формування звітів, агрегування даних та оцінювання ефективності роботи;
- модуль інтеграції, який забезпечує взаємодію з зовнішніми інформаційними системами та сервісами.

Взаємодія між зазначеними модулями здійснюється шляхом обміну даними та повідомленнями, що формують єдиний інформаційний простір системи. Така структура дозволяє розглядати систему як сукупність функціонально відокремлених компонентів, що є важливим при виборі архітектурного стилю.

До основних вимог, що висуваються до системи, належать: забезпечення високої продуктивності при обробленні великої кількості одночасних запитів, масштабованість у разі зростання кількості користувачів, надійність та відмовостійкість, а також зручність супроводження та модернізації. Крім того, система повинна забезпечувати швидку реакцію на події (звернення громадян) та підтримувати інтеграцію з іншими інформаційними ресурсами.

Наведені в таблиці 3.1 характеристики та можливі структури інформаційної системи визначають множину критеріїв, за якими доцільно здійснювати вибір архітектурного рішення, що обґрунтовує використання методу аналізу ієрархій у подальшому дослідженні.

Таблиця 3.1.

Характеристика існуючих ІТ-архітектур інформаційних систем

	 багатошарова (layered)	 модульний моноліт (modular monolith)	 мікроядро (microkernel)	 мікросервіси (microservices)	 сервісно- орієнтована на сервіси (service-based)	 сервісно- орієнтована (service-oriented)	 подієво-орієнтована (event-driven)	 орієнтована на простір даних (space-based)
Характеристика	технічна	технічна	технічна	технічна	технічна	технічна	технічна	технічна
розподіл (партиціювання)								
вартість	\$	\$	\$	\$\$\$\$\$	\$\$	\$\$\$\$	\$\$\$	\$\$\$\$
супроводжуваність	★	★★	★★★	★★★★★	★★★★★	★	★★★	★★★
тестованість	★★	★★	★★★	★★★★★	★★★★★	★	★★	★
розгортання	★	★★	★★★	★★★★★	★★★★★	★	★★★	★★★
простота	★★★★★	★★★★★	★★★	★	★★★	★	★	★
масштабованість	★	★	★	★★★★★	★★★	★★★	★★★★★	★★★★★
еластичність	★	★	★	★★★★★	★★	★★★	★★★	★★★★★
швидкодія (реактивність)	★★★	★★★	★★★	★★	★★★	★★	★★★★★	★★★★★
відмовостійкість	★	★	★	★★★★★	★★★★★	★★★	★★★★★	★★★
розвиток (еволюційність)	★	★	★★★	★★★★★	★★★★★	★	★★★★★	★★★
абстрагування	★	★	★★★	★	★	★★★★★	★★★★★	★
інтеграційність (інтероперабельність)	★	★	★★★	★★★	★★	★★★★★	★★★	★★

Основні характеристики існуючих ІТ-архітектур та їх важливість для побудови інформаційних систем подано в таблиці 3.2.

Таблиця 3.2

Характеристика	Чому важлива
1	2
Maintainability (супроводжуваність)	Систему необхідно постійно доопрацьовувати та підтримувати
Scalability (масштабованість)	Кількість заявок може зростати
Responsiveness (швидкодія / реактивність)	Користувач повинен швидко отримувати відповідь
Fault-tolerance (відмовостійкість)	Користувач повинен швидко отримувати відповідь
Deployability (розгортання)	Необхідно забезпечити зручне оновлення модулів

Характеристика	Чому важлива
1	2
Cost (вартість)	Вартість системи не повинна бути надто високою

Основними альтернативними архітектурними рішеннями для інформаційної системи є:

- багат шарова архітектура (Layered Architecture);
- модульний моноліт (Modular Monolith);
- архітектура мікроядра (Microkernel Architecture);
- мікросервісна архітектура (Microservices Architecture);
- сервісно-орієнтована на сервіси (Service-Based Architecture);
- сервісно-орієнтована архітектура (Service-Oriented Architecture, SOA);
- подієво-орієнтована архітектура (Event-Driven Architecture);
- архітектура, орієнтована на простір даних (Space-Based Architecture) [7].

Діаграма інформаційної системи опрацювання запитів користувача подана на рис. 3.1. Діаграму розроблено у нотації C4 [10] що дозволяє відобразити основні функціональні можливості, які реалізуються окремими контейнерами (модулями) в межах єдиної програмної системи, а також показати їх взаємодію з користувачами та зовнішніми сервісами без надмірної деталізації внутрішньої реалізації окремих компонентів.

На діаграмі виділено три категорії користувачів, які взаємодіють із системою:

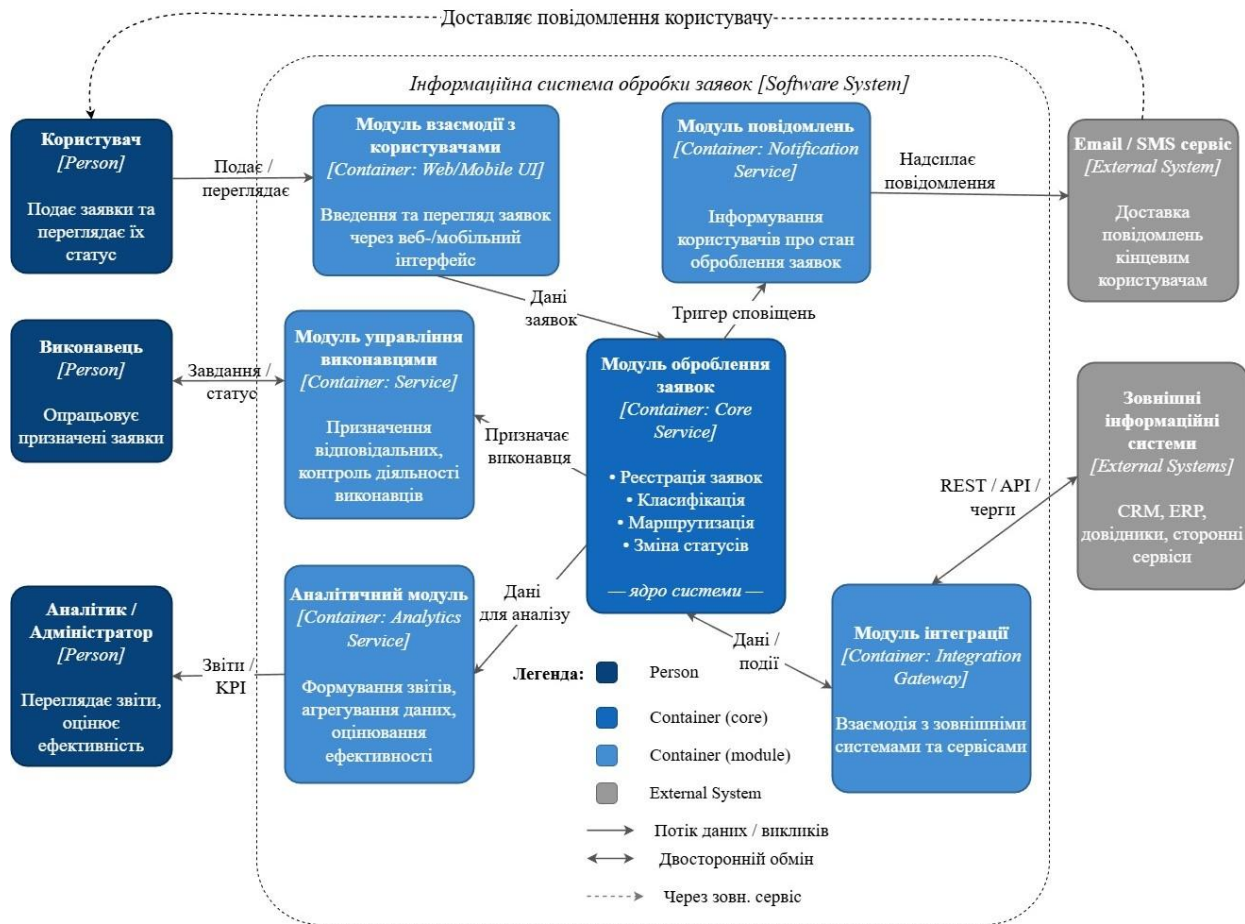


Рис. 3.1. Діаграма інформаційної системи опрацювання запитів користувача

- користувачі (заявники) — подають заявки та відстежують їхній стан;
- виконавці — приймають заявки до виконання та підтверджують результати їх виконання;
- аналітик/адміністратор — здійснює моніторинг активності системи та оцінює ефективність функціонування служб.

Розмежування прав доступу користувачів здійснюється відповідно до ролей, визначених у моделі доступу.

В межах системи виділено набір контейнерів, які формують її структурно-функціональні модулі. Центральним елементом архітектури є модуль оброблення заявок, який реалізує ключові бізнес-функції — реєстрацію, класифікацію,

маршрутизацію та зміну статусів заявок, а також забезпечує інтеграцію між основними компонентами системи, через які проходять інформаційні потоки.

Модуль взаємодії з користувачами реалізує веб- та мобільні інтерфейси для подання та перегляду заявок.

Модуль управління виконанням забезпечує оброблення заявок, підтвердження їх виконання та контроль відповідності встановленим вимогам.

Модуль сповіщень відповідає за інформування користувачів щодо стану заявок.

Аналітичний модуль здійснює агрегацію даних для формування звітності та оцінювання ефективності функціонування системи.

Модуль інтеграції виконує роль шлюзу для взаємодії з іншими інформаційними системами.

Для забезпечення міжсистемної взаємодії передбачено сервіс доставки повідомлень, через який реалізується передавання електронних сповіщень користувачам, а також інтеграція з зовнішніми інформаційними системами (зокрема, CRM, ERP та іншими сервісами) за допомогою прикладних програмних інтерфейсів (API) або черг повідомлень.

Зв'язки між контейнерами на діаграмі відображено спрямованими стрілками з описом, що визначає характер взаємодії та тип передаваних даних. Двосторонні зв'язки використовуються для компонентів, між якими здійснюється взаємний обмін даними або запитами, що характерно для таких пар, як «інтерфейс користувача — модуль оброблення заявок» або «модуль оброблення — модуль інтеграції».

У результаті представлена архітектура дозволяє чітко визначити функціональне призначення окремих модулів, встановити характер і інтенсивність взаємодії між ними, а також ідентифікувати потенційні точки інтеграції та вузькі місця системи.

Запропонована структуризація створює основу для подальшої формалізації задачі вибору архітектурного стилю, оскільки виділені характеристики модулів і їх взаємодій можуть бути використані як критерії оцінювання альтернативних архітектурних рішень у наступних розділах.

3.2. Формалізація завдання вибору ІТ-архітектури

Задача вибору архітектури інформаційної системи належить до класу багатокритеріальних задач прийняття рішень [16, 17] в умовах, коли альтернативи оцінюються одночасно за декількома, часто суперечливими, критеріями. Її формальне подання є необхідною передумовою для застосування методу аналізу ієрархій та забезпечує однозначну інтерпретацію вхідних даних, проміжних обчислень і кінцевого результату.

Постановка задачі. Нехай задано скінченну множину альтернативних архітектурних рішень

$$A = \{A_1, A_2, \dots, A_n\},$$

де n — кількість розглянутих архітектурних стилів. В межах бакалаврської роботи $n = 8$, а елементи множини відповідають перелікові, наведеному в підрозділі 3.1: багатошарова архітектура, модульний моноліт, архітектура мікроядра, мікросервісна архітектура, архітектура, орієнтована на сервіси (Service-Based), сервісно-орієнтована архітектура (SOA), подієво-орієнтована архітектура [7, 8] та архітектура, орієнтована на простір даних.

Нехай також задано скінченну множину критеріїв оцінювання

$$K = \{K_1, K_2, \dots, K_m\},$$

де m — кількість критеріїв. Згідно з таблицею 3.2, у роботі обрано шість критеріїв ($m = 6$): супроводжуваність (K_1), масштабованість (K_2), швидкодія /

реактивність (K_3), відмовостійкість (K_4), зручність розгортання (K_5), вартість (K_6).

Мета завдання. Необхідно обрати таку альтернативу $A^* \in A$, яка є найкращою з точки зору сукупного значення всіх критеріїв з урахуванням їх відносної важливості для розглянутої предметної області. Формально це записується як

$$A^* = \operatorname{argmax}_{A_i \in A} S(A_i),$$

де $S(A_i)$ — інтегральна (глобальна) оцінка i -ї альтернативи, що обчислюється як зважена сума її локальних оцінок за всіма критеріями:

$$S(A_i) = \sum_{j=1}^m w_j \cdot v_{ij}, \quad i = \overline{1, n}.$$

У наведеному виразі w_j — вага (коефіцієнт відносної важливості) -го критерію, v_{ij} — локальна оцінка i -ї альтернативи за j -м критерієм. Ваги критеріїв та локальні оцінки альтернатив задовольняють умовам нормування:

$$\sum_{j=1}^m w_j = 1, \quad w_j \geq 0; \quad \sum_{i=1}^n v_{ij} = 1, \quad v_{ij} \geq 0.$$

Особливості задачі. Складність розв'язання полягає в тому, що:

- критерії мають різну природу (якісні — супроводжуваність, зручність розгортання; кількісні — вартість) і виражаються в різних шкалах, що унеможливорює пряме їх порівняння;

- критерії є взаємно суперечливими: підвищення масштабованості та відмовостійкості зазвичай супроводжується зростанням вартості та ускладненням розгортання, тому жодна альтернатива не може бути найкращою за всіма критеріями одночасно;

- відсутні об'єктивні числові значення оцінок альтернатив за більшістю критеріїв, оскільки йдеться про якісні архітектурні характеристики, які можна оцінювати лише на основі експертних суджень;

- ваги критеріїв заздалегідь невідомі і також мають визначатися експертно з урахуванням специфіки конкретної інформаційної системи та вимог замовника.

Зазначені особливості унеможливають застосування простих методів прямого ранжування і обумовлюють використання спеціалізованого математичного апарату, який дозволяє формалізувати експертні судження та забезпечити перевірку їх внутрішньої узгодженості.

Формалізація через попарні порівняння. Для подолання вказаних труднощів задача зводиться до сукупності задач попарного порівняння елементів. Замість прямого присвоєння числових оцінок експерту пропонується послідовно порівнювати елементи попарно, що відповідає природному способу прийняття рішень людиною та забезпечує більшу точність суджень.

Результати попарних порівнянь критеріїв формуються у вигляді матриці попарних порівнянь критеріїв розмірності $m \times m$:

$$C = [c_{jk}]_{m \times m}, \quad c_{jk} > 0, \quad c_{jj} = 1, \quad c_{jk} = \frac{1}{c_{kj}},$$

де елемент c_{jk} відображає, у скільки разів критерій K_j важливіший за критерій K_k з точки зору розв'язуваної задачі. Властивість $c_{jk} = 1/c_{kj}$ забезпечує оберненосиметричність матриці.

Аналогічно для кожного критерію K_j (де $j = \overline{1, m}$) формується власна матриця попарних порівнянь альтернатив розмірності $n \times n$:

$$B^{(j)} = [b_{ik}^{(j)}]_{n \times n}, \quad b_{ik}^{(j)} > 0, \quad b_{ii}^{(j)} = 1, \quad b_{ik}^{(j)} = \frac{1}{b_{ki}^{(j)}}.$$

Загальна кількість матриць попарних порівнянь у задачі становить $m + 1 = 7$ (одна матриця критеріїв та шість матриць альтернатив). Чисельні значення елементів матриць задаються за дев'ятибальною шкалою Сааті, опис якої наводиться в підрозділі 3.3.

Ієрархічна структура задачі. Описана задача природним чином подається у вигляді трирівневої ієрархії:

- рівень 1 (мета) — вибір оптимальної архітектури інформаційної системи;
- рівень 2 (критерії) — множина $K = \{K_1, \dots, K_m\}$;
- рівень 3 (альтернативи) — множина $A = \{A_1, \dots, A_n\}$.

Кожен елемент верхнього рівня пов'язаний з усіма елементами безпосередньо нижчого рівня, що відповідає принципу повної ієрархії. Така структура дозволяє декомпонувати складну задачу на сукупність простіших задач попарного порівняння і поетапно обчислити шукані ваги.

Узагальнений алгоритм розв'язання. Розв'язання сформульованої задачі виконується послідовним виконанням таких етапів:

- побудова ієрархічної моделі задачі ;
- формування матриць попарних порівнянь на основі експертних суджень;
- обчислення локальних ваг критеріїв w_j та локальних оцінок альтернатив v_{ij} як компонент головних власних векторів відповідних матриць;
- перевірка узгодженості матриць шляхом обчислення індексу узгодженості та відношення узгодженості з урахуванням нормативних значень;
- обчислення інтегральних оцінок $S(A_i)$ для всіх альтернатив за формулою згортки;
- вибір альтернативи A^* з максимальним значенням інтегральної оцінки та обґрунтування отриманого результату.

Така постановка задачі повністю відповідає вимогам методу аналізу ієрархій Т. Сааті та забезпечує можливість послідовного розв'язання

сформульованої задачі вибору ІТ-архітектури з урахуванням як кількісних, так і якісних критеріїв оцінювання.

3.3. Побудова ієрархічної моделі завдання вибору ІТ-архітектури

Для подальшого розв'язання завдання вибору ІТ-архітектури необхідно побудувати ієрархічну модель задачі вибору ІТ-архітектури. Така модель формалізує процес декомпозиції складної багатокритеріальної задачі на сукупність простіших задач попарного порівняння і дозволяє послідовно обчислити локальні та глобальні пріоритети елементів.

Для розглянутої задачі побудовано трирівневу ієрархію (рис. 3.2), яка містить:

- на першому рівні — глобальну мету: вибір оптимальної ІТ-архітектури інформаційної системи моніторингу та обробки заявок;

- на другому рівні — шість критеріїв оцінювання ($m = 6$): супроводжуваність (K_1), масштабованість (K_2), швидкодія / реактивність (K_3), відмовостійкість (K_4), зручність розгортання (K_5) та вартість (K_6);

- на третьому рівні — вісім альтернативних архітектурних рішень ($n = 8$): багатошарова архітектура (A_1), модульний моноліт (A_2), архітектура мікроядра (A_3), мікросервісна архітектура (A_4), Service-Based архітектура (A_5), сервісно-орієнтована архітектура SOA (A_6), подієво-орієнтована архітектура (A_7) та архітектура, орієнтована на простір даних, Space-Based (A_8).

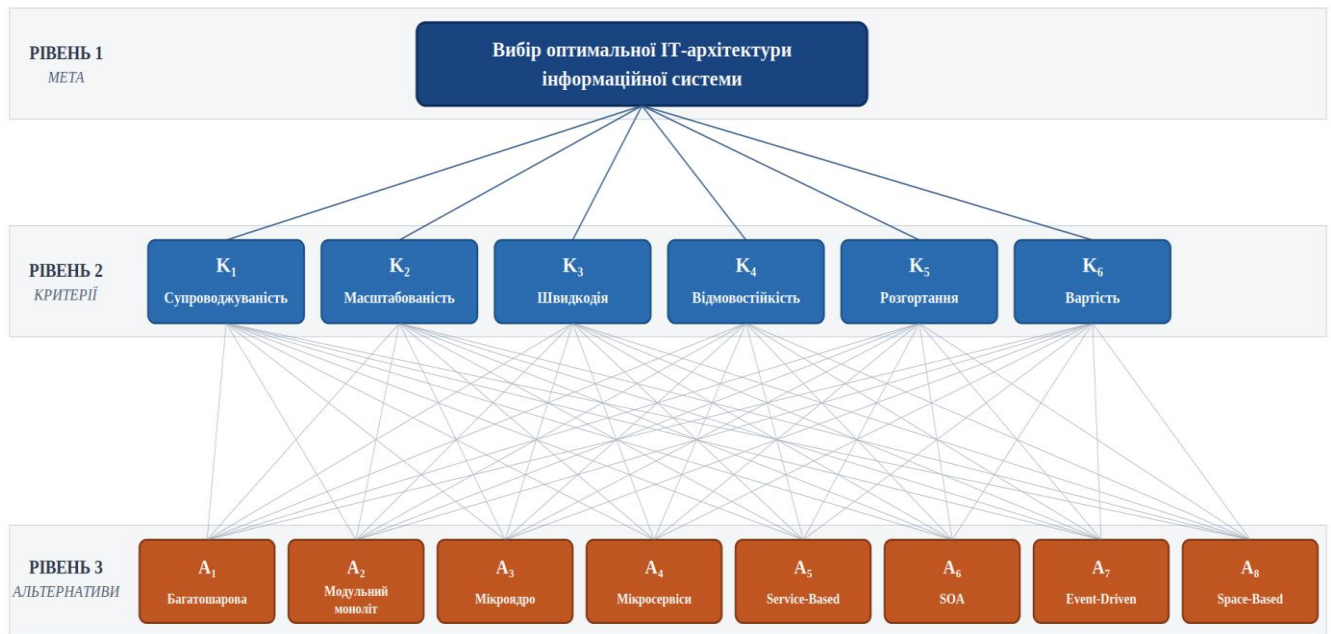


Рис. 3.2. Ієрархічна модель задачі вибору ІТ-архітектури

Побудована ієрархія є повною: кожен критерій рівня 2 пов'язаний з усіма альтернативами рівня 3. Це означає, що кожна альтернатива оцінюється за всіма критеріями без винятку, а отриманий вектор глобальних пріоритетів коректно відображає сукупну якість кожного рішення.

Кількість матриць попарних порівнянь, що мають бути сформовані для повного розв'язання задачі, складає $m + 1 = 7$:

- одна матриця попарних порівнянь критеріїв C розмірності 6×6 для визначення вектора ваг критеріїв $w = (w_1, w_2, \dots, w_6)$;

- шість матриць попарних порівнянь альтернатив $B^{(1)}, B^{(2)}, \dots, B^{(6)}$ розмірності 8×8 , кожна з яких відображає відносні переваги альтернатив за відповідним критерієм та використовується для визначення локальних векторів оцінок $v^{(j)} = (v_{1j}, v_{2j}, \dots, v_{8j})$, $j = \overline{1,6}$.

Таким чином, побудована ієрархічна модель повністю відповідає вимогам методу Сааті та забезпечує підґрунтя для подальшого формування матриць попарних порівнянь у наступному підрозділі.

3.4. Формування матриць попарних порівнянь

Числові значення елементів матриць попарних порівнянь визначаються за дев'ятибальною шкалою Сааті, наведеною в табл. 3.3. Шкала містить непарні значення, що відповідають якісним рівням переваги одного елемента над іншим, а парні значення (2, 4, 6, 8) використовуються як проміжні.

Таблиця 3.3

Шкала відносної важливості Т. Сааті

Бал	Якісна оцінка
1	Однакова важливість елементів
3	Помірна перевага одного над іншим
5	Суттєва (значна) перевага
7	Сильна перевага
9	Абсолютна (вирішальна) перевага
2, 4, 6, 8	Проміжні значення

3.4.1. Матриця попарних порівнянь критеріїв

Матриця попарних порівнянь критеріїв [13] **С** сформована на підставі експертних суджень з урахуванням специфіки розглянутої предметної області — інформаційної системи моніторингу та обробки заявок громадян. Найвищий пріоритет надано масштабованості (K_2), оскільки кількість користувачів і обсяг звернень може значно зростати, далі — відмовостійкості (K_4) як критичній характеристиці системи державного рівня, потім — швидкодії (K_3), супроводжуваності (K_1), вартості (K_6) та зручності розгортання (K_5). Сформовану матрицю наведено в табл. 3.4.

Таблиця 3.4

Матриця попарних порівнянь критеріїв К

	K_1	K_2	K_3	K_4	K_5	K_6
K_1	1	1/4	1/2	1/3	3	2
K_2	4	1	3	2	6	5
K_3	2	1/3	1	1/2	4	3
K_4	3	1/2	2	1	5	4
K_5	1/3	1/6	1/4	1/5	1	1/2
K_6	1/2	1/5	1/3	1/4	2	1

Матриця К задовольняє основним властивостям матриць попарних порівнянь: оберненосиметричність ($c_{jk} = 1/c_{kj}$), одиничність діагональних елементів ($c_{jj} = 1$) та додатність усіх елементів ($c_{jk} > 0$).

3.4.2. Матриці попарних порівнянь альтернатив

Для кожного з шести критеріїв сформовано окрему матрицю попарних порівнянь восьми альтернатив розмірності 8×8 . Експертні судження щодо переваги однієї альтернативи над іншою за конкретним критерієм базуються на типових архітектурних характеристиках відповідних стилів, описаних у фундаментальній літературі з програмної архітектури.

Таблиця 3.5

Матриця $B^{(1)}$ за критерієм K_1 (супроводжуваність)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	1	1/2	1	1/3	1/2	1/2	1/2	1/2
A_2	2	1	1	1/2	1/2	1	1	1
A_3	1	1	1	1/3	1/2	1/2	1/2	1/2
A_4	3	2	3	1	1	2	2	2
A_5	2	2	2	1	1	1	1	1
A_6	2	1	2	1/2	1	1	1	1
A_7	2	1	2	1/2	1	1	1	1
A_8	2	1	2	1/2	1	1	1	1

Таблиця 3.6

Матриця $B^{(2)}$ за критерієм K_2 (масштабованість)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	1	1/2	1/2	1/5	1/3	1/3	1/4	1/4
A_2	2	1	1	1/3	1/2	1/2	1/2	1/3
A_3	2	1	1	1/3	1/2	1/2	1/2	1/3
A_4	5	3	3	1	2	2	1	1
A_5	3	2	2	1/2	1	1	1/2	1/2
A_6	3	2	2	1/2	1	1	1/2	1/2
A_7	4	2	2	1	2	2	1	1
A_8	4	3	3	1	2	2	1	1

Таблиця 3.7

Матриця $B^{(3)}$ за критерієм K_3 (швидкодія)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	1	1/2	1/2	1/2	1/2	1/2	1/3	1/3
A_2	2	1	1	1	1	1	1/2	1/2
A_3	2	1	1	1	1	1	1/2	1/2
A_4	2	1	1	1	1	1	1/2	1/2
A_5	2	1	1	1	1	1	1/2	1/2
A_6	2	1	1	1	1	1	1/2	1/2
A_7	3	2	2	2	2	2	1	1
A_8	3	2	2	2	2	2	1	1

Таблиця 3.8

Матриця $B^{(4)}$ за критерієм K_4 (відмовостійкість)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	1	1/2	1/2	1/4	1/3	1/3	1/4	1/4
A_2	2	1	1	1/3	1/2	1/2	1/2	1/2
A_3	2	1	1	1/3	1/2	1/2	1/2	1/2
A_4	4	3	3	1	2	2	1	1
A_5	3	2	2	1/2	1	1	1/2	1/2
A_6	3	2	2	1/2	1	1	1/2	1/2
A_7	4	2	2	1	2	2	1	1
A_8	4	2	2	1	2	2	1	1

Таблиця 3.9

Матриця $B^{(5)}$ за критерієм K_5 (розгортання)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	1	1	1	2	1	2	1	2
A_2	1	1	1	2	1	2	1	2
A_3	1	1	1	1	1	1	1	2
A_4	1/2	1/2	1	1	1	1	1	1
A_5	1	1	1	1	1	1	1	2
A_6	1/2	1/2	1	1	1	1	1	1
A_7	1	1	1	1	1	1	1	2
A_8	1/2	1/2	1/2	1	1/2	1	1/2	1

Таблиця 3.10

Матриця $B^{(6)}$ за критерієм K_6 (вартість)

Для критерію вартості елемент $b_{ik}^{(6)}$ інтерпретується як перевага в економічній доступності (нижча вартість $\rightarrow$ вища оцінка).

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
A_1	1	1	1	3	2	2	2	4
A_2	1	1	1	3	2	2	2	4
A_3	1	1	1	2	1	1	1	3
A_4	1/3	1/3	1/2	1	1/2	1/2	1/2	1
A_5	1/2	1/2	1	2	1	1	1	2
A_6	1/2	1/2	1	2	1	1	1	2
A_7	1/2	1/2	1	2	1	1	1	2
A_8	1/4	1/4	1/3	1	1/2	1/2	1/2	1

Усі сім сформованих матриць задовольняють основним властивостям матриць попарних порівнянь і готові до подальшого обчислення локальних ваг елементів.

3.5. Розрахунок ваг критеріїв та альтернатив вибору ІТ-архітектури

Для обчислення вектора локальних ваг елементів матриці попарних порівнянь у методі аналізу ієрархій застосовується метод власного вектора, який ґрунтується на знаходженні головного власного вектора матриці. Шуканий вектор ваг w визначається як нормований власний вектор, що відповідає максимальному за модулем власному значенню $\lambda_{\max}$ матриці M , тобто є розв'язком рівняння:

$$M \cdot w = \lambda_{\max} \cdot w, \quad \sum_{i=1}^n w_i = 1.$$

3.5.1. Ваги критеріїв

У результаті застосування методу власного вектора до матриці C (табл. 3.4) отримано максимальне власне значення $\lambda_{\max}^C = 6,1225$ та відповідний нормований власний вектор. Розрахункові значення ваг критеріїв подано у табл. 3.11.

Таблиця 3.11

Ваги критеріїв

Критерій	Позначення	Вага w_j	Частка, %
Супроводжуваність	K_1	0,1006	10,06
Масштабованість	K_2	0,3825	38,25
Швидкодія	K_3	0,1596	15,96
Відмовостійкість	K_4	0,2504	25,04
Розгортання	K_5	0,0428	4,28
Вартість	K_6	0,0641	6,41
Сума		1,0000	100,00

Аналіз отриманих ваг показує, що найбільша важливість у задачі належить масштабованості (38,25 %) та відмовостійкості (25,04 %), сумарно — 63,3 %. Це відповідає специфіці інформаційної системи державного рівня з очікуваним зростанням навантаження. Найменшу вагу мають критерії розгортання (4,28 %) та вартості (6,41 %), що відображає пріоритет якісних характеристик функціонування над операційними витратами.

3.5.2. Локальні ваги альтернатив

Аналогічна процедура застосована до кожної з шести матриць $B^{(j)}$ (табл. 3.5–3.10) [14, 15]. Отримані локальні вектори ваг альтернатив за окремими критеріями зведено у матрицю V розмірності 8×6 (табл. 3.12), у якій рядки відповідають альтернативам, а стовпці — критеріям. В межах кожного стовпця сума значень дорівнює одиниці.

Таблиця 3.12

Матриця локальних ваг альтернатив $V = [v_{ij}]_{8 \times 6}$

Альтернатива	K_1	K_2	K_3	K_4	K_5	K_6
A_1 Багатошарова	0,0658	0,0401	0,0574	0,0417	0,1581	0,2008
A_2 Модульний моноліт	0,1086	0,0696	0,1063	0,0744	0,1581	0,2008
A_3 Мікроядро	0,0725	0,0696	0,1063	0,0744	0,1326	0,1446
A_4 Мікросервіси	0,2180	0,2064	0,1063	0,2044	0,1035	0,0607
A_5 Service-Based	0,1539	0,1143	0,1063	0,1165	0,1326	0,1131
A_6 SOA	0,1271	0,1143	0,1063	0,1165	0,1035	0,1131
A_7 Event-Driven	0,1271	0,1843	0,2055	0,1860	0,1326	0,1131
A_8 Space-Based	0,1271	0,2015	0,2055	0,1860	0,0790	0,0536
Сума	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

З таблиці 3.12 видно, що найвищі локальні оцінки за основними критеріями (масштабованість, відмовостійкість) мають мікросервісна архітектура (A_4), Space-Based (A_8) та подієво-орієнтована (A_7), тоді як за критеріями розгортання та вартості провідні позиції займають традиційні моноліти (A_1 , A_2).

3.6. Перевірка узгодженості матриць

Для оцінювання якості експертних суджень у методі аналізу ієрархій обчислюються два показники: індекс узгодженості (Consistency Index) та відношення узгодженості (Consistency Ratio).

Індекс узгодженості обчислюється за формулою:

$$CI = \frac{\lambda_{\max} - n}{n - 1},$$

де $\lambda_{\max}$ — максимальне власне значення матриці, n — її розмірність.

Відношення узгодженості визначається співвідношенням:

$$CR = \frac{CI}{RI(n)},$$

де $RI(n)$ — середнє значення індексу узгодженості для випадкових матриць розмірності n (табл. 3.13).

Таблиця 3.13

Значення випадкового індексу узгодженості $RI(n)$

n	3	4	5	6	7	8	9
$RI(n)$	0,58	0,90	1,12	1,24	1,32	1,41	1,45

Згідно з критерієм Сааті, матриця попарних порівнянь вважається узгодженою, якщо $CR \leq 0,10$. У випадку перевищення цього порогу експертні судження потребують перегляду.

Результати перевірки узгодженості для всіх семи матриць наведено в табл. 3.14.

Таблиця 3.14

Перевірка узгодженості матриць попарних порівнянь

Матриця	Розмірність	$\lambda_{\max}$	CI	RI	CR	Висновок
K — критерії	6×6	6,1225	0,0245	1,24	0,0198	узгоджено
$B^{(1)}$ — за K_1	8×8	8,1005	0,0144	1,41	0,0102	узгоджено
$B^{(2)}$ — за K_2	8×8	8,0753	0,0108	1,41	0,0076	узгоджено
$B^{(3)}$ — за K_3	8×8	8,0130	0,0019	1,41	0,0013	узгоджено
$B^{(4)}$ — за K_4	8×8	8,1017	0,0145	1,41	0,0103	узгоджено
$B^{(5)}$ — за K_5	8×8	8,1360	0,0194	1,41	0,0138	узгоджено
$B^{(6)}$ — за K_6	8×8	8,0742	0,0106	1,41	0,0075	узгоджено

Усі обчислені значення відношення узгодженості значно нижчі за нормативний поріг $CR = 0,10$ (максимальне значення складає $CR = 0,0198$ для

матриці критеріїв). Це свідчить про високу внутрішню несуперечливість сформованих експертних суджень і дозволяє вважати отримані локальні ваги достовірними.

3.7. Обчислення інтегральних оцінок альтернатив

Інтегральна (глобальна) оцінка i -ї альтернативи [14, 15] обчислюється як скалярний добуток вектора її локальних ваг (відповідного рядка матриці V) на вектор ваг критеріїв w :

$$S(A_i) = \sum_{j=1}^m w_j \cdot v_{ij}, \quad i = \overline{1, n}.$$

У матричній формі результат подається як добуток:

$$S = V \cdot w,$$

де $S = (S(A_1), S(A_2), \dots, S(A_n))^T$ — вектор інтегральних оцінок.

Підстановка числових значень із табл. 3.11 та 3.12 дає, наприклад, для альтернативи A_4 (мікросервісна архітектура):

$$S(A_4) = 0,1006 \cdot 0,2180 + 0,3825 \cdot 0,2064 + 0,1596 \cdot 0,1063 + 0,2504 \cdot 0,2044 + 0,0428 \cdot 0,1035 + 0,0641 \cdot 0,0607 = 0,1774.$$

Аналогічно обчислено інтегральні оцінки для всіх восьми альтернатив. Результати наведено в табл. 3.15 у порядку спадання.

Таблиця 3.15

Інтегральні оцінки альтернатив

Ранг	Альтернатива	$S(A_i)$	Частка, %
1	A_4 Мікросервісна архітектура	0,1774	17,74
2	A_8 Space-Based архітектура	0,1760	17,60
3	A_7 Подієво-орієнтована (Event-Driven)	0,1756	17,56

Ранг	Альтернатива	$S(A_i)$	Частка, %
4	A_5 Service-Based архітектура	0,1183	11,83
5	A_6 SOA	0,1143	11,43
6	A_2 Модульний моноліт	0,0928	9,28
7	A_3 Архітектура мікроядра	0,0844	8,44
8	A_1 Багатошарова архітектура	0,0612	6,12
Сума		1,0000	100,00

Аналіз отриманих значень дозволяє виокремити три групи альтернатив:

- група лідерів (A_4 , A_8 , A_7) з оцінками понад 0,17, що становить близько 53 % сумарного пріоритету та об'єднує сучасні розподілені архітектури;
- група середніх рішень (A_5 , A_6) з оцінками 0,11–0,12, що відповідає частково розподіленим сервіс-орієнтованим підходам;
- група відстаючих рішень (A_2 , A_3 , A_1) з оцінками нижче 0,10 — традиційні моноліти, які поступаються за ключовими критеріями.

3.8. Вибір оптимальної ІТ-архітектури інформаційної системи

Згідно з результатами обчислень, наведеними у табл. 3.15, найвищу інтегральну оцінку $S(A^*) = 0,1774$ отримала мікросервісна архітектура (A_4), яка визнається оптимальним архітектурним рішенням для розглянутої інформаційної системи моніторингу та обробки заявок громадян.

Обґрунтування вибору. Перевага мікросервісної архітектури зумовлена її високими локальними оцінками за найбільш вагомими критеріями задачі [7, 8]:

- за масштабованістю (K_2 , вага 38,25 %) — локальна оцінка $v_{42} = 0,2064$ є найвищою серед усіх альтернатив, що пояснюється можливістю незалежного

горизонтального масштабування окремих сервісів пропорційно їх навантаженню;

- за відмовостійкістю (K_4 , вага 25,04 %) — локальна оцінка $v_{44} = 0,2044$ також лідирує, оскільки відмова окремого сервісу не призводить до повної непрацездатності системи завдяки ізоляції процесів та можливості резервування;

- за супроводжуваністю (K_1 , вага 10,06 %) — локальна оцінка $v_{41} = 0,2180$ є найвищою, оскільки невеликі автономні сервіси простіше модифікувати, тестувати та розгортати без впливу на інші частини системи.

Аналіз чутливості. Слід відзначити, що різниця між трьома лідерами рейтингу (A_4 , A_8 , A_7) є відносно невеликою (від 0,1756 до 0,1774, що становить менше 1,1 %). Це означає, що при незначних змінах експертних суджень результат може змінитися. Зокрема:

- при підвищенні ваги критерію швидкодії (K_3) перевагу могла б отримати Space-Based архітектура (A_8), оптимізована для високопродуктивних обчислень з даними у пам'яті;

- при наявності виразної подієвої природи бізнес-процесів та необхідності асинхронної обробки великих потоків подій конкурентоспроможним рішенням є подієво-орієнтована архітектура (A_7).

Однак для розглянутої предметної області — системи обробки заявок громадян, де домінуючими є вимоги до масштабованості, відмовостійкості та можливості незалежної модифікації окремих модулів, — саме мікросервісна архітектура забезпечує найкращий баланс характеристик.

Архітектурна реалізація. Виходячи з обраного стилю, архітектурна структура системи, представлена на рис. 3.1, природним чином відображається на мікросервісну реалізацію: кожен з шести модулів (модуль взаємодії з користувачами, модуль оброблення заявок, модуль управління виконавцями, модуль повідомлень, аналітичний модуль, модуль інтеграції) реалізується як

автономний сервіс із власною базою даних, незалежним розгортанням та власним життєвим циклом. Взаємодія між сервісами здійснюється через API-шлюз та шину подій для асинхронних сценаріїв.

ВИСНОВКИ

У бакалаврській роботі розв'язано актуальну науково-практичну задачу розроблення методичного підходу до обґрунтованого вибору ІТ-архітектури інформаційної системи на основі методу аналізу ієрархій та його практичного застосування до задачі побудови інформаційної системи моніторингу й обробки заявок громадян. У результаті виконаного дослідження отримано такі основні результати.

1. Проведено системний аналіз сучасних архітектур ІТ-систем. Розглянуто основні поняття та запропоновано класифікацію інформаційних систем за рівнем управління (TPS, MIS, DSS, ES), функціональним призначенням (ERP, CRM, SCM, BI) та архітектурною побудовою (централізовані, клієнт-серверні, розподілені, хмарні). Визначено основні види архітектурних стилів — від багат шарової архітектури та модульного моноліту до мікросервісних, подієво-орієнтованих та орієнтованих на простір даних рішень. Сформульовано поняття атрибутів якості архітектури як системи характеристик, що використовуються для оцінювання та порівняння архітектурних рішень.

2. Досліджено основні методи багатокритеріального прийняття рішень. Проведено порівняльний аналіз методу зваженої суми, методу аналізу ієрархій, TOPSIS, ELECTRE, PROMETHEE та нечітких методів багатокритеріального вибору з урахуванням таких чинників, як здатність опрацьовувати якісні та кількісні критерії, можливість перевірки узгодженості суджень, простота реалізації та обчислювальна складність. Обґрунтовано вибір методу аналізу ієрархій як найбільш адекватного для розглядуваної задачі з огляду на його

універсальність, наявність формалізованої процедури перевірки узгодженості експертних суджень та можливість агрегування критеріїв різної природи.

3. Розроблено математичну модель ієрархічного багатокритеріального вибору ІТ-архітектури. Формалізовано процедури формування локальних пріоритетів критеріїв та альтернатив на основі методу головного власного вектора, а також обчислення глобальних пріоритетів альтернатив за формулою згортки. Сформульовано умови нормування ваг та властивості матриць попарних порівнянь, що забезпечують коректність математичної постановки задачі.

4. Розроблено узагальнений алгоритм побудови та розв'язання ієрархічної моделі багатокритеріального вибору ІТ-архітектури, який включає шість послідовних етапів: формулювання мети, формування переліку критеріїв та альтернатив, побудову ієрархічної моделі, формування матриць попарних порівнянь, обчислення локальних і глобальних пріоритетів та перевірку узгодженості. Запропонований алгоритм забезпечує системність розв'язання задачі та можливість його повторного застосування до інших класів інформаційних систем.

5. Виконано практичне застосування розробленого підходу до конкретної задачі. Розглянуто інформаційну систему моніторингу та обробки заявок громадян, виокремлено її основні функціональні модулі (модуль взаємодії з користувачами, модуль оброблення заявок, модуль управління виконавцями, модуль повідомлень, аналітичний модуль, модуль інтеграції) та побудовано структурно-функціональну модель системи в нотації C4, яка формалізує склад системи та її зв'язки із зовнішнім середовищем.

6. Виконано формальну постановку задачі вибору ІТ-архітектури як багатокритеріальної задачі прийняття рішень з вісьмома альтернативами (багатошарова архітектура, модульний моноліт, архітектура мікроядра, мікросервісна архітектура, Service-Based, SOA, подієво-орієнтована, Space-

Based) та шістьма критеріями оцінювання (супроводжуваність, масштабованість, швидкодія, відмовостійкість, зручність розгортання, вартість). Побудовано трирівневу ієрархічну модель задачі.

7. Сформовано сім матриць попарних порівнянь — одну матрицю критеріїв розмірності 6×6 та шість матриць альтернатив розмірності 8×8 — на основі дев'ятибальної шкали Сааті. Експертні судження сформовано з урахуванням специфіки інформаційної системи державного рівня та узагальнених характеристик архітектурних стилів, наведених у фундаментальній літературі з програмної архітектури.

8. Обчислено локальні та глобальні пріоритети методом головного власного вектора. Встановлено, що найбільший внесок у задачу мають критерії масштабованості (38,25 %) та відмовостійкості (25,04 %), що відповідає специфіці інформаційної системи з очікуваним зростанням навантаження. Перевірено узгодженість усіх сформованих матриць — максимальне значення відношення узгодженості склало $CR = 0,0198$, що значно нижче за нормативний поріг 0,10 і свідчить про високу несуперечливість експертних суджень.

9. На основі обчислених інтегральних оцінок встановлено, що оптимальним архітектурним рішенням для розглядуваної інформаційної системи є мікросервісна архітектура (інтегральна оцінка $S = 0,1774$). Перевагу мікросервісної архітектури обґрунтовано її найвищими локальними оцінками за критеріями з найбільшою вагою — масштабованістю та відмовостійкістю, які забезпечуються можливістю незалежного горизонтального масштабування окремих сервісів та ізоляцією процесів, що запобігає поширенню відмов.

Отримані у роботі результати підтверджують, що метод аналізу ієрархій є ефективним інструментом для обґрунтованого вибору ІТ-архітектури інформаційної системи в умовах багатокритеріальності, поєднання якісних і кількісних критеріїв та обмеженості об'єктивних числових оцінок альтернатив.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сорока К. О. Основи теорії систем і системного аналізу: навч. посіб. Харків : ХНАМГ, 2004. 291 с.
2. Морозов А. О., Куса Н. Д., Гриша С. М. Інформаційні технології в економіці : підручник. Київ : Кондор, 2018. 432 с.
3. Грицунов О. В. Інформаційні системи та технології : навч. посіб. Харків : ХНАМГ, 2010. 222 с.
4. Laudon K. C., Laudon J. P. Management Information Systems: Managing the Digital Firm. 17th ed. Pearson, 2022. 656 p.
5. Power D. J. Decision Support Systems: Concepts and Resources for Managers. Westport : Quorum Books, 2002. 251 p.
6. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Addison-Wesley Professional, 2021. 464 p.
7. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media, 2020. 432 p.
8. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd ed. O'Reilly Media, 2021. 612 p.
9. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 560 p.
10. Brown S. The C4 model for visualising software architecture. URL: <https://c4model.com> (дата звернення: 15.04.2026).
11. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva : ISO, 2011. 34 p.

12. Kazman R., Klein M., Clements P. ATAM: Method for Architecture Evaluation. Technical Report CMU/SEI-2000-TR-004. Pittsburgh : Software Engineering Institute, Carnegie Mellon University, 2000. 83 p.
13. Saaty T. L. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. New York : McGraw-Hill, 1980. 287 p.
14. Saaty T. L. Decision making with the analytic hierarchy process. International Journal of Services Sciences. 2008. Vol. 1, No. 1. P. 83–98.
15. Saaty T. L., Vargas L. G. Models, Methods, Concepts and Applications of the Analytic Hierarchy Process. 2nd ed. New York : Springer, 2012. 346 p.
16. Тоценко В. Г. Методи та системи підтримки прийняття рішень: алгоритмічний аспект. Київ : Наукова думка, 2002. 381 с.
17. Triantaphyllou E. Multi-Criteria Decision Making Methods: A Comparative Study. Boston : Kluwer Academic Publishers, 2000. 290 p.
19. Hwang C.-L., Yoon K. Multiple Attribute Decision Making: Methods and Applications. Berlin : Springer-Verlag, 1981. 269 p.
20. Behzadian M., Khanmohammadi Otaghsara S., Yazdani M., Ignatius J. A state-of-the-art survey of TOPSIS applications. Expert Systems with Applications. 2012. Vol. 39, No. 17. P. 13051–13069.
21. Roy B. Classement et choix en présence de points de vue multiples : la méthode ELECTRE. Revue française d'informatique et de recherche opérationnelle. 1968. Vol. 2, No. 8. P. 57–75.
22. Brans J. P., Vincke P. A Preference Ranking Organisation Method: The PROMETHEE Method for Multiple Criteria Decision-Making. Management Science. 1985. Vol. 31, No. 6. P. 647–656.
23. Zadeh L. A. Fuzzy sets. Information and Control. 1965. Vol. 8, No. 3. P. 338–353.

24. Згуровський М. З., Панкратова Н. Д. Системний аналіз: проблеми, методологія, застосування. 2-ге вид. Київ : Наукова думка, 2011. 727 с.
25. Катренко А. В. Системний аналіз об'єктів та процесів комп'ютеризації : навч. посіб. Львів : Новий світ-2000, 2003. 424 с.
26. Saaty T. L. How to make a decision: The Analytic Hierarchy Process. *European Journal of Operational Research*. 1990. Vol. 48, No. 1. P. 9–26.
27. Vaidya O. S., Kumar S. Analytic hierarchy process: An overview of applications. *European Journal of Operational Research*. 2006. Vol. 169, No. 1. P. 1–29.
28. Evans E. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, 2003. 560 p.
29. *Microsoft Application Architecture Guide*. 2nd ed. Microsoft Press, 2009. 528 p.
URL: [https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ee658099\(v=pandp.10\)](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ee658099(v=pandp.10)) (дата звернення: 18.04.2026).
30. ДСТУ ISO/IEC/IEEE 42010:2019. Інженерія систем і програмних засобів. Архітектурний опис. Київ : ДП «УкрНДНЦ», 2019. 36 с.