

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



**СИСТЕМНИЙ АНАЛІЗ**

Навчальний посібник

Дніпро  
НТУ «ДП»  
2026

УДК 303.732.4 : 004.4

С 40

*Рекомендовано вченою радою НТУ «Дніпровська політехніка»  
як навчальний посібник для здобувачів ступеня бакалавра  
зі спеціальності 124 Системний аналіз  
(протокол № 14 від 23 04. 2026)*

Рецензенти:

В.Є. Бахрушин – д-р фіз.-мат. наук, проф. (Національний університет України «Запорізька політехніка»);

О.В. Заріцький – д-р техн. наук, доц. (Київський національний університет імені Тараса Шевченка).

Автори: А.В. Малієнко, О.Б. Владико, Т.В. Хом'як, С.В. Козир.

**Системний** аналіз [Електронний ресурс]: навч. посіб. /  
С 40 А.В. Малієнко, О.Б. Владико, Т.В. Хом'як, С.В. Козир; М-во освіти і  
науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро :  
НТУ «ДП», 2026. – 250 с.

Зміст посібника відповідає програмі підготовки здобувачів спеціальності 124 Системний аналіз (F4 Системний аналіз та наука про дані). У цьому виданні розглянуто методологію та основні інструменти управління розвитком складних систем, висвітлено окремі аспекти і тематики управління організаціями, зокрема системний підхід до його реалізації. Уміщено питання для самоконтролю засвоєння знань, перелік використаної та рекомендованої літератури для поглибленого вивчення матеріалу дисципліни «Системний аналіз».

**УДК 303.732.4 : 004.4**

© А.В. Малієнко, О.Б. Владико,  
Т.В. Хом'як, С.В. Козир 2026  
© НТУ «Дніпровська політехніка», 2026

## ЗМІСТ

ПЕРЕДМОВА.....	4
<b>1. ОСНОВНІ ПОНЯТТЯ Й ПРИНЦИПИ СИСТЕМНОГО ПІДХОДУ.....</b>	<b>6</b>
Тема 1. Базові поняття теорії систем і системний аналіз .....	6
Тема 2. Структурний системний аналіз.....	22
Тема 3. Опис систем на етапі аналізу.....	33
Тема 4. Побудова ієрархічної структури систем.....	57
Тема 5. Оцінка якості систем .....	65
<b>2. ІНФОРМАЦІЙНИЙ СИСТЕМНИЙ АНАЛІЗ .....</b>	<b>83</b>
Тема 6. Марківські процеси в системах. Інформаційний критерій ефективності системи .....	83
Тема 7. Моделювання систем. Моделювання роботи програмного забезпечення .....	92
Тема 8. Системний аналіз об'єктів і процесів методом імітаційного моделювання .....	113
Тема 9. Оцінка якості програмних систем .....	139
Тема 10. Експертний аналіз проблем. Обробка і використання знань...	169
Тема 11. Евристичні методи аналізу й оптимізації систем.....	183
Тема 12. Корпоративна інформаційна система .....	203
Тема 13. Аналіз трендів і часових рядів.....	227
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	243

## ПЕРЕДМОВА

Сьогодні системний аналіз виступає не лише як методологія або набір інструментів – він є центральною парадигмою дослідження й управління складними технічними системами в умовах великої невизначеності, цифровізації та глобальних змін. Історичне зародження системних ідей у середині ХХ століття – «системна революція», розвиток загальної теорії систем і методів операційного аналізу – усе це заклало підвалини системного аналізу. Втім сучасний стан дисципліни істотно розширив її горизонти: поряд із класичними підходами з'явилися методи моделювання даних, створення цифрових двійників, машинне навчання та інструменти для аналізу інтегрованого життєвого циклу систем.

Предметом системного аналізу залишається сама система незалежно від її природи, масштабу чи способу опису. Проте нині аналіз систем орієнтується на багаторівневу методологію, що йде від формалізації структури й поведінки об'єктів до оцінки можливих ризиків, стійкості й впливу на соціальне середовище. Мета застосування системного підходу до такого аналізу – це не тільки оптимізація технічних рішень, а й узгодженість бізнес-стратегій, інформаційних потоків з людським фактором, з метою створення керованих, безпечних і адаптивних систем.

Практична реалізація окреслених завдань потребує від виконавців мультигалузевої співпраці інженерів, аналітиків даних, економістів, фахівців з кібербезпеки, соціологів та управлінців. Системний аналіз об'єднує зусилля фахівців, пропонуючи методичні підходи до виділення системи з середовища, її формального опису, побудови концептуальних і математичних моделей, вибору стратегій управління, імітаційного та аналітичного моделювання, а також процедур верифікації та валідації рішень.

Цей посібник покликаний дати читачеві інструментальну базу й концептуальну рамку для виконання широкого кола завдань: від постановки проблеми до впровадження інтегрованих інфраструктурних рішень. У ньому поєднано класичні положення системного аналізу та сучасні методи, що робить його корисним для студентів, науковців і практиків, які працюють у сфері проєктування, аналізу складних систем та управління ними.

Основні завдання вивчення дисципліни здобувачам :

- засвоїти основні методи постановки та розв'язування задач системного аналізу в умовах невизначеності, складності, а також у ситуаціях, коли потрібно провести дослідження системи;
- набути знань про основи систем, що мають спільну мету, функції, структуру, а також про методи дослідження системи як єдиного цілого, враховуючи взаємозв'язки між її елементами та із зовнішнім середовищем;
- розуміти сутність базових методів системного аналізу, оптимального керування, теорії прийняття рішень з огляду на практику розв'язування прикладних задач управління;
- оволодіти засобами прогнозування на основі методів імітаційного

моделювання, експертного аналізу, а також евристичних методів;

– опанувати методики створення ефективних алгоритмів для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.

Після вивчення дисципліни здобувачі ступеня бакалавра спеціальності 124 Системний аналіз (F4 Системний аналіз та наука про дані) мають досягти таких результатів:

– Знати і вміти застосовувати принципи, алгоритми й основні методи постановки і розв’язування задач системного аналізу їхнього опису.

– Знати і вміти застосовувати основні методи постановки та розв’язування задач за допомогою структурного аналізу, побудови ієрархічної структури систем та оцінювання їхньої якості.

– Успішно застосовувати методи і засоби роботи з даними і знаннями, оцінювання якості програмних систем, марківських процесів у системах та моделювання роботи програмного забезпечення.

– Користуватись методами і засобами роботи з даними і знаннями, управління корпоративними інформаційними системами.

– Уміти за допомогою методів імітаційного моделювання проводити експертний аналіз, а також користуватись евристичними методами аналізу

– Уміти моделювати, прогнозувати розвиток систем за допомогою часових рядів.

# 1. ОСНОВНІ ПОНЯТТЯ Й ПРИНЦИПИ СИСТЕМНОГО ПІДХОДУ

## Тема 1. Базові поняття теорії систем і системний аналіз

### 1. Огляд основних понять системного аналізу: система, елемент системи, середовище.

Системний аналіз - це підхід до розв'язання складних проблем, що ґрунтується на вивченні систем як цілісних об'єктів, визначенні їхніх елементів і взаємодії між ними. Основні поняття системного аналізу є фундаментом для розуміння складних об'єктів і процесів, що виникають у різних галузях.

#### Ключові поняття системного аналізу:

**Система:** Сукупність взаємопов'язаних елементів, які функціонують разом для досягнення певної мети. Система має межі, структуру, функції, а також взаємодії з середовищем.

*Визначення:* Система є сукупність елементів (підсистем). За певних умов елементи самі можуть розглядатися як системи, а система, що досліджується, – як елемент більш складної системи:

– зв'язки між елементами в системі перевершують по силі зв'язку цих елементів з елементами, що не входять до системи. Ця властивість дозволяє виділити систему із середовища;

– для будь-якої системи характерно існування інтегративних якостей (властивість ємерджентності), які притаманні системі загалом, але не властиві жодному її елементу окремо: систему не можна зводити до простої сукупності елементів;

– система завжди має цілі, для яких вона функціонує та існує

**Складна та велика система:** Однією з характерних тенденцій розвитку суспільства нині є поява великих надзвичайно складних систем (великі автоматизовані, технологічні, енергетичні, гідротехнічні, інформаційні та інші комплекси). З іншого боку прагнення пізнати світ проживання людства як складну багатofункціональну систему стало реальністю сьогодення. Усе це призвело до необхідності визначити поняття складної системи, розробити методичні засади її дослідження, управління та проектування.

Нині однозначного, чіткого визначення складної системи немає. Відомі різні підходи та запропоновані різні формальні ознаки її визначення. Так, радянський вчений Г.М. Поворов пропонує відносити до складних систем, такі що мають 104 -107 елементів; до ультраскладних - системи, що складаються з 107-1030 елементів; і до суперсистем – системи із 1030-10200 елементів.

Такий підхід має той недолік, що це визначення складності є відносним, а не абсолютним.

Англійський кібернетик С. Бір пропонує до складних відносити системи, що описуються мовою теоретико-імовірнісних методів (мозок, економіка, форма тощо).

Найбільш чітким з погляду, визначенням складних систем є визначення, дане, наприклад, в [1].

*Визначення:*

Складною системою називається система, моделі якої недостатньо інформації для ефективного управління цією системою.

Отже, ознакою простоти системи є достатність інформації її управління. Якщо результат управління, отриманий за допомогою моделі, буде несподіваним, то таку систему відносять до складної.

Для переведення системи в розряд простий необхідно отримання інформації про неї і включення її в модель.

Від складних систем потрібно відрізнити великі системи.

*Визначення:*

Система, для актуалізації моделі якої з метою управління бракує матеріальних ресурсів (машинного часу, ємності пам'яті, інших матеріальних засобів моделювання) називається великою.

До таких систем належать економічні, організаційно-управлінські, нейрофізіологічні, біологічні тощо. системи. Способом переведення великих систем у прості є створення нових потужніших засобів обчислювальної техніки.

Як видно з визначень, поняття великої та складної системи є різними. Однак у літературі ці поняття визначено неоднозначно. Деякі автори взагалі не використовують ці поняття, інші використовують їх як синоніми, а деякі вважають різницю між ними чисто кількісною.

Щоб ще раз підкреслити суттєву різницю між поняттями "велика" та "складна" системи наведемо приклади з роботи. При цьому зведемо їх до наступної таблиці:

Таблиця 1.1 – Зовнішні фактори, які взаємодіють із системою і можуть впливати на її функціонування, але не є її частиною.

№	Система	Мала	Велика	Проста	Складна
1	Справний побутовий прилад для користувача	+		+	
2	Несправний побутовий прилад для майстра	+			+
3	Шифрозамок для викрадача		+	+	
4	Мозок, живий організм		+		+

У табл. 1.1. знаком "+" відмінено класифікаційні ознаки систем. Пояснимо, наприклад, чому шифрозамок віднесено до класу великих та простих систем. Ця система - велика, тому що у викрадача може не

вистачити ресурсу часу для розтину замка; а проста – тому що розтин зводиться до простого багатоваріантного перебору шифрів. На рис. 1.1. показані всілякі поєднання ознак систем "проста - складна", "мала - велика".

На рис. 1.1. показані всілякі поєднання ознак систем "проста - складна", "мала - велика".

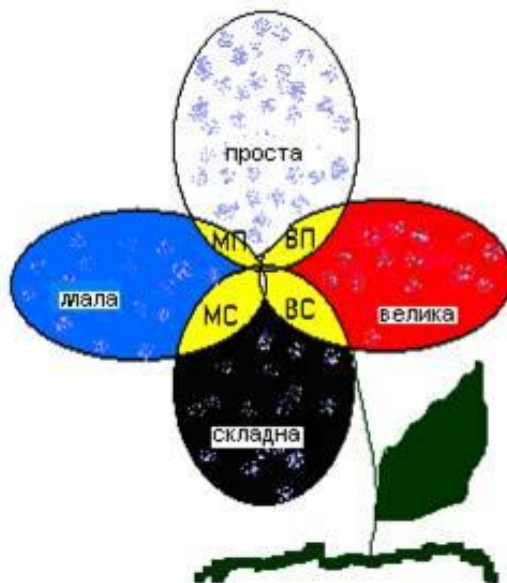


Рис. 1.1. Можливості поєднання ознак систем

**Елемент системи:** Окремі складові частини системи, які взаємодіють між собою для досягнення загальної мети.

**Система є сукупністю елементів, об'єднаних між собою зв'язками.** Її основною характеристикою є інтегративні властивості, тобто такі якості, які притаманні лише системі в цілому, але відсутні у кожного окремого елемента. Завдяки цьому система не зводиться до простої сукупності елементів.

*Визначення:*

Для будь-яких систем характерна наявність інтеграційних якостей (властивостей). Інтегративними називаються якості, властиві системі загалом, але з властиві жодному з її елементів окремо. Тому система не зводиться до простої сукупності елементів, і, розчленовуючи систему деякі частини (підсистеми), вивчаючи кожну їх окремо, не можна пізнати всі властивості системи загалом.

Для будь-яких систем характерна наявність суттєвих зв'язків між елементами, що перевищують за потужністю (силою) зв'язку цих елементів з елементами, що не входять до цієї системи. Визначення властивості дозволяє виділити систему як цілісного об'єкта з довкілля.

**Компоненти систем** - це складові частини, які разом формують систему, забезпечуючи її функціонування та цілісність. До компонентів

системи належать елементи та зв'язки, які визначають структуру і характер взаємодії між частинами системи.

*Визначення:*

Елемент - неподільна частина системи, що має самостійність по відношенню до даної системи. Неподільність елемента сприймається як недоцільність обліку у межах моделі даної системи його внутрішньої будови.

Зв'язок – сукупність залежності властивостей одного елемента від властивостей інших елементів системи. Встановити зв'язок між двома елементами – це означає виявити наявність залежності їх властивостей.

Взаємодія - сукупність взаємозв'язків та взаємовідносин між властивостями елементів, коли вони набувають характеру взаємодії один одному.

Структура системи - сукупність елементів системи та зв'язків між ними у вигляді множини.

Зовнішнє середовище - це набір існуючих у просторі та в часі об'єктів (систем), які, як передбачається, діють на систему рис. 2.

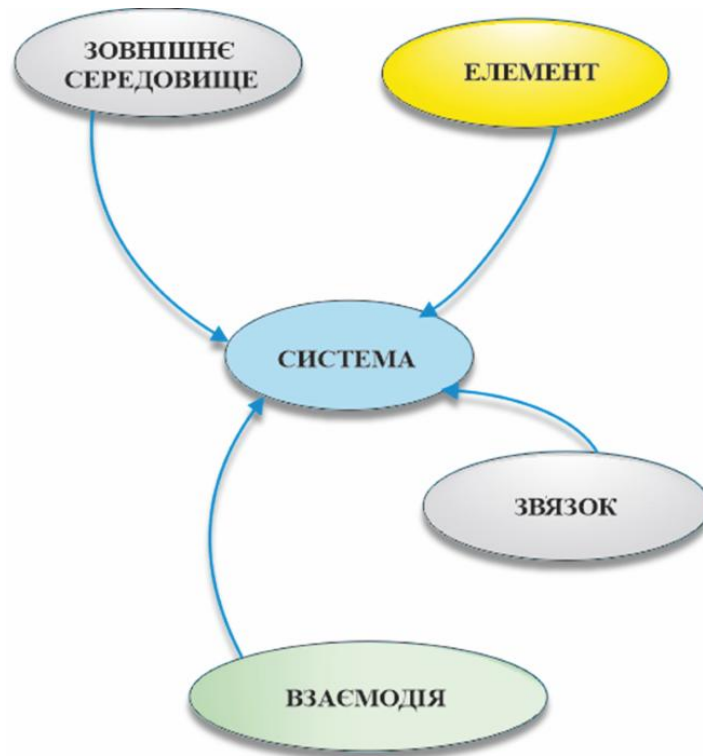


Рис. 1.2. Компоненти систем

Компоненти системи формують її структуру. Елементи виступають як основні будівельні блоки, а зв'язки забезпечують інтеграцію цих блоків у єдине ціле. Саме взаємодія між елементами через зв'язки створює умови для появи інтегративних властивостей системи, які відсутні в окремих елементах.

Таким чином, елементи і зв'язки - це базові компоненти, які визначають функціональність і цілісність будь-якої системи.

## **1.2. Основні принципи системного аналізу: цілісність, інтеграція, ієрархія.**

Системний аналіз ґрунтується на кількох основних принципах, які дозволяють ефективно аналізувати складні системи та знаходити оптимальні рішення [2]. Розглянемо принципи докладніше:

**Принцип цілісності.** Система розглядається як єдине ціле, а не просто набір окремих елементів. Важливо розуміти, що кожен елемент взаємодіє з іншими і що зміна одного елемента може вплинути на всю систему.

**Принцип ієрархічності.** Система розглядається як складна структура з різними рівнями, між якими встановлюються певні зв'язки. Системи можуть мати складну структуру, і для зручності розуміння її розбивають на підсистеми. Кожен рівень ієрархії може мати свої завдання та функції.

Ієрархічні системи зазвичай досліджуються та створюються «згори», починаючи з аналізу модулів першого ієрархічного рівня. У разі відсутності ієрархії дослідник повинен вирішити, в якому порядку він розглядатиме частини системи. Так, наприклад, конструктор під час створення нового зразка виділяє у ньому початковий елемент, якого потім подумки чи кресленні підганяє другий, третій, наступні. Наладчик починає пошук несправності в системі з тестів, які визначають найбільш типові відмови.

**Принцип інтегрованості.** Усі компоненти системи повинні працювати разом і інтегруватися для досягнення загальної мети. Важливий аспект - забезпечення взаємодії між різними частинами системи.

**Принцип багатомірності,** який полягає в тому, що вивчаються різні характеристики систем, які об'єднують в групи (кластери): об'єкт описується як сукупність деяких характеристик та взаємозв'язків між ними.

**Принцип кінцевої мети** – означає, що у цілеспрямованій системі все має бути підпорядковане глобальній мети. Будь-яка спроба зміни, вдосконалення та управління в такій системі повинна оцінюватися з точки зору того, допомагає чи заважає вона досягненню кінцевої мети. Це накладає особливу відповідальність на вибір мети та її чітке трактування. Розпливчасті, не повністю визначені кінцеві цілі спричиняють неясності в структурі та управлінні системою, і, як наслідок, невірні дії в системі. Такі дії можуть бути і наслідком невіри в кінцеву мету або можливість її досягнення.

**Принцип функціональності** стверджує, що будь-яка структура тісно пов'язана з функцією системи та її частин, і досліджувати (створювати) структуру необхідно після з'ясування функцій у системі. Насправді цей принцип, зокрема, означає, що у разі надання системі нових функцій корисно переглядати її структуру, а чи не намагатися втиснути нову функцію в стару схему. Так, перебудова виробництва, пов'язана із запровадженням автоматизації, веде як до виникнення нових підрозділів (обчислювальний

центр, група системних програмістів, група створення та супроводу банку даних), так і до розбудови структури наявних. Ці зміни зачіпають, звісно, і систему управління.

Принцип модульної побудови вказує на можливість розгляду замість частини системи сукупності її вхідних та вихідних впливів. Він стверджує корисність абстрагування від зайвої деталізації за збереження можливості адекватно описувати системи. Принцип ієрархії акцентує увагу на корисності відшукування чи створення системі ієрархічного (домінуючого) характеру зв'язків між елементами, модулями, цілями.

Принцип розвитку досить добре пояснений у його формулюванні. Поняття розвитку, змінюваності при збереженні якісних особливостей виділяється майже у будь-якій природній системі, а в штучних можливість розвитку, удосконалення, як правило, закладається в основу створення системи. При модульній побудові такий розвиток зазвичай зводиться до заміни та додавання модулів (частин). Так, можливість розширення функцій та модернізації закладаються у принципи побудови банків даних та знань, програмних комплексів, багатоцільових роботів та інших складних технічних систем. Слід зазначити, що межі розширення функцій зазвичай визначені і досить обмежені. Навряд чи розумно створюватиме універсальний програмний засіб, здатний керувати верстатом і грати в шахи

Принцип децентралізації рекомендує, щоб керуючі впливи і рішення виходили не тільки з одного центру (головного елемента). Ситуація, коли управління виходять із одного місця, називається повної централізацією. Таке становище вважається виправданим лише за особливої відповідальності за все, що відбувається у системі, і за нездатності елементів системи самостійно реагувати зовнішні впливи. Система з повною централізацією буде негнучкою, що не пристосовується, не має «внутрішньої активності». Цілком імовірно, що в такій системі канали інформації, що ведуть до головного елемента, виявляться перевантаженими, а сам цей елемент, не в змозі переробити таку кількість інформації, почне видавати неправильні управління.

У системах, де стійких механізмів регуляції немає, неминуче наявність тієї чи іншої ступеня централізації. У цьому виникає питання оптимальному поєднанні команд ззовні (згори) і команд, вироблюваних усередині цієї групи елементів. Загальний принцип такого поєднання простий: ступінь централізації має бути мінімальним, що забезпечує виконання поставленої мети.

Принцип невизначеності стверджує, що ми можемо мати справу з системою, в якій нам не все відомо чи зрозуміло. Це може бути система з нез'ясованою структурою, з непередбачуваним перебігом процесів, із значною ймовірністю відмов у роботі елементів, з невідомими зовнішніми впливами та ін. Приватним випадком невизначеності виступає випадковість – ситуація, коли вид події відомий, але може або наступити, або наступити.

### 1.3. Методи системного аналізу: кількісні та якісні методи, моделювання.

Методи системного аналізу забезпечують науково обґрунтоване рішення завдань, як при вдосконаленні систем, так і при їх проектуванні. Вони дозволяють правильно сформулювати проблему, поставити дослідницькі та проєктні цілі, визначити альтернативи вирішення завдань, масштаб невизначеності в кожному з розглянутих варіантів, вибрати критерії їх оцінки, порівняти альтернативи, вибрати найбільш підходящий варіант і прийняти рішення.

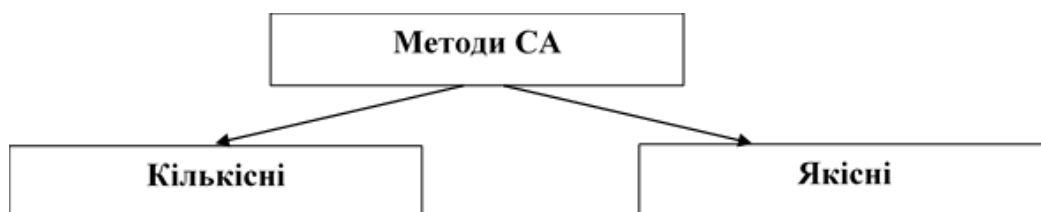


Рис. 1.3. Кількісні та якісні методи системного аналізу

У системному аналізі застосовуються різні методи для вирішення задач, аналізу та розробки рішень.

Кількісні методи дозволяють отримати аналітичну залежність між параметрами, що описують поведінку системи для вирішення прикладних задач: прогнозування поведінки, моделювання, управління (забезпечення бажаної поведінки).

*Основні методи:*

- Апроксимація
- Кореляційний і регресійний аналіз
- Математичне та імітаційне моделювання
- Автоматичні та автоматизовані системи управління

Якісні методи застосовують людський досвід та моделювання прийняття рішень людиною на етапах постановки проблеми, аналізу варіантів, прийняття рішення, а також для характеристики правильності і точності варіантів. Їх ключова відмінність: аналітичні залежності відсутні або слабо виражені

*Основні якісні методи системного аналізу:*

- Мозковий штурм
- Методи створення сценаріїв
- Методи експертних оцінок (парні порівняння, метод аналізу ієрархій)
- Метод «Дельфі»
- Дерево цілей

Поетапний підхід у системному аналізі передбачає послідовне проходження чотирьох взаємопов'язаних стадій рис 1.4:

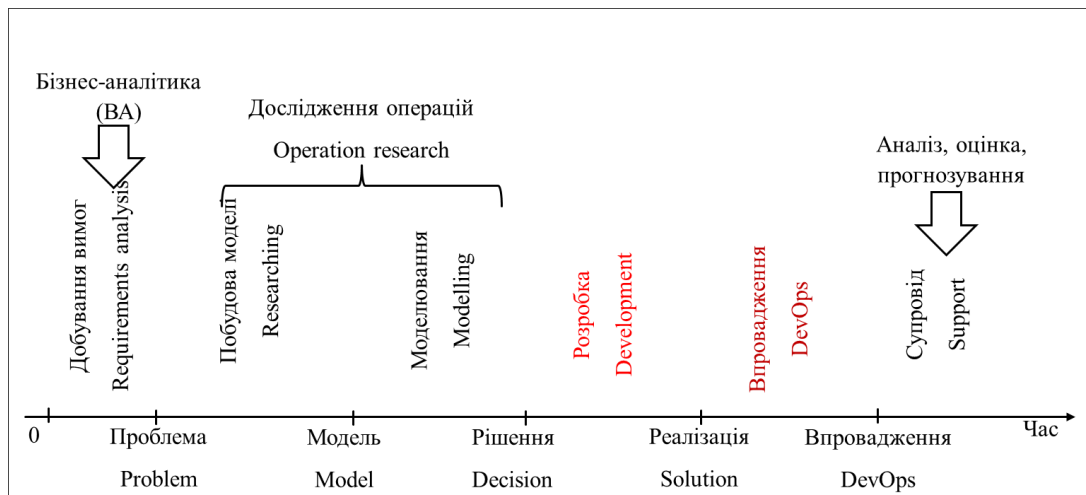


Рис 1.4. Системний аналіз в процесі розробки програмного забезпечення

На першому етапі системного аналізу здійснюється формулювання та коректне розуміння проблемної ситуації, яка лежить в основі потреби в розробці чи вдосконаленні інформаційної системи. Основним завданням цього етапу є встановлення чіткої мети та контексту майбутнього рішення шляхом збору та узагальнення експертних знань замовника і кінцевих користувачів. З цією метою проводяться структуровані інтерв'ю чи анкетування представників зацікавлених сторін. Під час таких інтерв'ю досліджуються питання, що стосуються призначення системи та її функціонального призначення, а саме: з якою метою потрібна нова система або модернізація існуючої, які конкретні бізнес-процеси вона має автоматизувати або спростити, а також які найбільш критичні «болі» (pain points) сьогодні проявляються у поточному способі організації цих процесів. Окрім того, обов'язковим є уточнення очікуваних показників ефективності (КРІ), таких як середній час реакції на запити, пропускна здатність системи, максимальне число одночасних користувачів та інші метрики, що характеризують продуктивність і надійність майбутнього рішення.

Паралельно з інтерв'ю здійснюється аналіз наявної інформаційної системи (або, в разі її відсутності, відповідних ручних процедур), зокрема проводиться збір даних щодо тих операцій, які сьогодні виконуються користувачами вручну. Для цього часто використовують методи спостереження, аналізу протоколів роботи або опитування працівників, що безпосередньо взаємодіють із процесом. В результаті цього дослідження виявляються «вузькі місця», які можуть проявлятися, наприклад, у надмірному часі очікування обробки запиту (як у випадку з call-центрами), у високому рівні помилок, зумовлених дублюванням даних чи некоректною інформацією, а також у втраті важливої інформації внаслідок відсутності централізованого сховища. Накопичена інформація дозволяє консолідувати уявлення про проблеми бізнесу та обґрунтувати потребу у змінах.

На завершальному підкроці першого етапу здійснюється побудова контекстної схеми взаємодії майбутньої системи із зовнішнім середовищем, де формалізуються ролі (Actors) та ключові бізнес-процеси на високому рівні

абстракції. Така схема відображає, зокрема, взаємозв'язки між користувачами, іншими інформаційними системами та зовнішніми базами даних, що полегшує подальші етапи моделювання та проектування архітектури рішення. У підсумку першого етапу сформовано єдине цілісне бачення проблеми, узагальнено вимоги зацікавлених сторін та визначено основні обмеження й критерії успішності майбутньої системи.

На етапі «Модель» здійснюється формалізація та систематизація інформації, зібраної на попередньому етапі формулювання проблеми. Цей етап покликаний перевести неструктуровані відомості про вимоги, бізнес-процеси й обмеження в чіткі, формальні уявлення про досліджувану систему. Загалом у рамках цієї стадії розрізняють два основні підходи: побудову якісних моделей і створення кількісних моделей [3].

Якісні моделі являють собою графічні чи логічні схеми, які на високому рівні демонструють структуру системи, її внутрішню архітектуру, основні потоки даних та взаємозв'язки між компонентами. Під час формування таких моделей системний аналітик використовує універсальні засоби нотації, зокрема діаграми потоків даних (DFD), що відображають перетворення інформації на рівнях контексту та детального декомпозованого представлення; UML-діаграми, серед яких діаграми випадків використання (Use Case) ілюструють ключові сценарії взаємодії акторів із системою, діаграми класів описують сутності й їхні атрибути зі зв'язками, а діаграми активностей (Activity) ілюструють послідовність операцій при реалізації функціональних вимог; нотація IDEF0, що фокусується на визначенні вхідних й вихідних даних, елементів керування й механізмів для кожної функції системи; а також ER-моделювання (Entity–Relationship), яке призначене для формалізації структури даних і встановлення типів сутностей, їхніх атрибутів і зв'язків між ними. У результаті якісного моделювання формується сукупність діаграм, які формально описують функціональні потоки, структуру даних, послідовність сценаріїв і взаємозв'язки між класами, сервісами та інтерфейсами.

Кількісні моделі використовують математичні формалізми або методи імітаційного моделювання для проведення числового аналізу ключових характеристик системи, зокрема параметрів продуктивності, надійності та ризиків. Найбільш поширеним прикладом є застосування теорії черг. Якщо бізнес-процес передбачає обслуговування «заявок» (документованих звернень, телефонних дзвінків, веб-запитів тощо), теорія черг дозволяє отримати кількісну оцінку таких характеристик, як середня кількість запитів у черзі ( $L_q$ ), середній час перебування заявки в системі ( $W$ ) і коефіцієнт завантаженості каналів обслуговування ( $\rho$ ), що визначається як співвідношення інтенсивності надходження заявок ( $\lambda$ ) до добутку кількості паралельних каналів ( $c$ ) на інтенсивність обслуговування одним каналом ( $\mu$ ), тобто  $\rho = \lambda / (c \cdot \mu)$ .

Кількісні моделі подібного роду можуть бути аналітичними моделями класу M/M/1 (одноканальна система масового обслуговування з одним

сервером), де передбачено експоненціальний розподіл інтервалів надходження заявок і часу обслуговування або реалізованими як імітаційні експерименти, що дозволяють розглядати складніші розподіли інтенсивностей та сценарії пікових навантажень. Такий підхід забезпечує обґрунтування проєктних рішень із урахуванням очікуваних показників продуктивності, імовірності відмов чи перевантажень, а також дозволяє порівняти альтернативні варіанти архітектури системи з точки зору забезпечення заданих рівнів якості обслуговування.

Таким чином, етап «Модель» поєднує формування графічно-логічного опису системи, що демонструє структуру, відносини й послідовності бізнес-процесів, із математичним аналізом її критичних характеристик, що дозволяє створити єдине когерентне представлення системи як із точки зору її функціонування, так і з погляду ефективності її роботи під навантаженням.

На етапі «Рішення» здійснюється безпосередня розробка рекомендацій щодо проєктування, оптимізації або реорганізації програмного забезпечення на основі даних і висновків, отриманих із якісних та кількісних моделей. Спочатку системний аналітик інтерпретує результати якісного моделювання (DFD, UML, ER) і виявляє реальні, а не умовні вузькі місця в існуючій архітектурі та бізнес-процесах. Зокрема, аналіз DFD може засвідчити, що інформація з форми «Уточнення деталей» у CRM-підсистемі не надходить до підсистеми аналітики, що свідчить про втрату даних і необхідність зміни алгоритму обміну між компонентами. UML Use Case-діаграми можуть виявити відсутність певного функціоналу, наприклад секції «Запит документації», що означає, що частина бізнес-вимог узагалі не реалізується в системі. ER-модель, у свою чергу, демонструє невідповідність між бізнес-правилами (наприклад, умова «один клієнт може мати багато контрактів») та реальною структурою бази даних, яка обмежує цю можливість лише одним записом.

На основі такого аналізу системний аналітик формулює декілька альтернативних варіантів реалізації, проте жоден із них не є остаточним до порівняльної оцінки. Кожен варіант уточнюється з точки зору досягнення визначених бізнес-цілей, технологічних обмежень і ресурсних можливостей. У процесі порівняння варіантів виконується багатокритеріальна оцінка, яка може включати аналіз витрат і вигод (TCO/ROI), прогнозовані строки окупності, а також рівень відповідності кожного рішення заданим нефункціональним вимогам. Важливо, що саме на цьому етапі чітко визначаються ключові нефункціональні вимоги: продуктивність (performance), масштабованість (scalability), надійність (reliability), безпека (security) та підтримуваність (maintainability).

Після порівняння альтернатив системний аналітик обґрунтовує вибір конкретного варіанта та деталізує технічну архітектуру системи. Це передбачає опис високорівневої структури компонентів, їхніх взаємодій, вибір стеку технологій і інструментів (наприклад, серверних платформ, баз даних, фреймворків, механізмів обміну повідомленнями). Відповідно до

обраної архітектури уточнюються параметри інтеграційних з'єднань, стандарти обміну даними й API-інтерфейси. Водночас із цим розробляються деталізовані специфікації, що визначають основні програмні модулі, їхній функціонал, межі відповідальності та взаємозв'язки.

Ключовою частиною фази «Рішення» є підготовка фінансово-економічного обґрунтування, яке демонструє доцільність обраного варіанту з точки зору загальних витрат на володіння (TCO) та очікуваних вигоди (ROI). Така оцінка враховує масштаби інвестицій у розробку, ліцензійні витрати, потреби в інфраструктурі та поточні операційні витрати, а також прогнозоване підвищення ефективності бізнес-процесів, зменшення помилок і втрат даних, скорочення часу обробки транзакцій та інші економічно значущі метрики. Завершуючи етап «Рішення», система аналітики має повний набір документів: порівняльний аналіз альтернатив, архітектурні специфікації, перелік технологій, узгоджені нефункціональні вимоги та обґрунтування аналітичних розрахунків, які необхідні для переходу до наступного етапу - реалізації і впровадження.

На етапі «Реалізація» відбувається безпосередня власне розробка програмного забезпечення, його тестування, розгортання, інтеграція з існуючими системами та забезпечення подальшої підтримки створеної системи відповідно до раніше узгоджених технічних специфікацій і архітектурних рішень. Організація процесу розробки спирається на вибір конкретної методології, що визначає послідовність виконання ключових фаз: у разі застосування класичної моделі «водоспаду» (Waterfall) роботи відбуваються послідовно – від аналізу вимог через проєктування й написання коду до тестування, впровадження і подальшої підтримки. Головною перевагою такої моделі є чітка структура й ясний поділ етапів, що спрощує контроль виконання завдань і погодження артефактів на кожному кроці; натомість її недоліком є низька адаптивність до змін після початку розробки, оскільки необхідність повернення до попереднього етапу вимагає значних витрат часу та ресурсів.

Під час реалізації системний аналітик спільно з архітекторами, розробниками й тестувальниками контролює, щоб усі попередньо визначені рішення, зокрема структури баз даних, сервіси й інтерфейси, були реалізовані згідно з технічними умовами. Тестування – як модульне, так і інтеграційне, системне й приймальне – проводиться у відповідності до сформованого плану тестування, що містить набір тест-кейсів, критерії успіху й пороги прийнятного рівня дефектності. Автоматизоване розгортання, засноване на CI/CD-пайплайнах, забезпечує швидке виявлення та усунення дефектів без затримок, притаманних ручним релізам, і дозволяє гарантувати стабільне перенесення конфігурацій і залежностей середовищ розробки, тестування й продакшну.

Нарешті, після успішного запуску рішення до продакшен-середовища виникає потреба у безперервній підтримці, моніторингу працездатності та періодичному вдосконаленні системи. Це передбачає налаштування системи

збору метрик і логів, організацію процесу обробки інцидентів та виконання заходів із запобігання повторним відмовам, а також регулярний аналіз продуктивності задля своєчасного масштабування або оптимізації окремих компонентів. Таким чином, саме комплексне поєднання організації розробки, належного тестування, автоматизованого розгортання та послідовної підтримки визначає життєздатність проєкту в довгостроковій перспективі.

#### **1.4. Алгоритм системного аналізу в процесі розробки програмного забезпечення**

Постановка та стадії розробки ПЗ [4].

*Склад алгоритму:*

1. Постановка проблеми (неформальна)
2. Визначення стейкхолдерів та їх цілей.
3. Формування критеріїв (числових, якісних) як показників досягнення цілей.
4. Генерація рішень - альтернатив.
5. Добування знань чи вимог
6. Структурування знань чи вимог
7. Формалізація вимог
8. Вибір альтернатив
9. Моделювання роботи системи
10. Формування технічного завдання на проєкт.
11. Аналіз, оцінка і прогнозування роботи готового проєкту

##### **1. Постановка проблеми (неформальна)**

Ідентифікувати проблему та виконати максимально точну її постановку.

Проаналізувати логічну структуру проблеми (підпроблеми, аспекти, нюанси).

Виконати аналіз історичного розвитку проблеми (в минулому і в майбутньому).

Виявити зовнішні зв'язки проблеми з іншими проблемами).

З'ясувати, чи можна принципово вирішити проблему.

Опис проблеми - відповідь на питання:

«Які обставини та попередній досвід змушують саме цих зацікавлених сторін, у цьому конкретному культурному середовищі, яке включає ці цінності, сприймати такий стан речей як проблему?»

Всі проблеми, як формалізовані, так і напівструктуровані, повинні бути сформульовані таким чином, щоб стати завданням вибору шляхів, які підходять для досягнення мети проєкту.

**2. Визначення стейкхолдерів та їх цілей.** При ідентифікації проблем в англійській літературі часто використовується термін, «стейкхолдери» (stakeholders), що в перекладі означає «зацікавлені особи». У список зацікавлених осіб рекомендується включити:

- **замовника**, тобто того, хто ставить задачі (описує проблему);

- **осіб, які приймають рішення**, тобто ті, від чийх повноважень безпосередньо залежить вирішення проблеми (не завжди співпадають із замовником);

- **учасники**, як *активні* елементи ті, чий дії потрібні при вирішенні проблеми, так і *пасивні* ті, на кого вплинуть (позитивні чи негативні) наслідки вирішення проблеми;

- **проектна команда**, яка буде вирішувати проблему і вносити зміни в життя інших стейкхолдерів.

**На цьому етапі системного аналізу визначається, що потрібно зробити, щоб усунути проблему.**

Основні труднощі при визначенні цілей полягають в наступному:

- мета – це бажане майбутнє, тобто мета схожа на антипод проблеми, в якому легко визнати неточність або помилитися;

- іноді за цілі вважають методи, бо мета для одного рівня ієрархії, для іншого – спосіб досягнення мети.

- формування цілей людини визначається системою його цінностей, які іноді бувають суперечливими;

- цілі змінюються з плином часу.

**При виявленні цілей корисні такі методи:**

- включати в цілі протилежні висловлювання і «подвійні» цілі;

- виявити не тільки «бажані», а й «небажані» цілі, щоб передбачити появу нових проблем.

Поступово узагальнюючи цілі, слід досягти максимального рівня абстракції. Тоді цілі стають критеріями.

**3. Формування критеріїв (числових, якісних) як показників досягнення цілей.**

Можливість порівнювати системи існує лише в тому випадку, коли цілі, що їх характеризують, **вимірювані**. Система  $s_0$  може бути визнана кращою (оптимальною) на множині систем  $S$ , якщо не існує жодної  $s' \in S$ , яка б задовольняла вимозі  $s' > s_0$ .

Ціль  $A$  кількісно вимірювана на множині порівняних систем  $S$ , якщо на  $S$  задана дійсна функція  $u(s)$ , яка має властивість упорядкованості. Така функція має назву критерій. Критерії бувають числові й нечислові.

Упорядкованість може бути строгою або нестроною, відповідно особа, що приймає рішення (ОПР) може встановити між двома системами по відповідності цілі  $A$  одне з наступних відношень:

система  $s'$  однозначно краще системи  $s''$ ;

система  $s'$  однозначно не краще системи  $s''$ ;

система  $s'$  порівняно однакова з системою  $s''$ ;

система  $s'$  не порівняна з системою  $s''$ .

Останнє відношення встановлюється наприклад тоді, коли ціль  $A$  - *частково вимірювана*. Передостаннє - коли різниця в показниках системи не перевищує *роздільної здатності* ОПР, тобто одна система чи рішення краще від іншої на величину, прийняту за одиницю виміру критерію.

**Кожен критерій задається на певній шкалі вимірювань. Розрізняють наступні види шкал (за зростанням складності і деталізації):**

1) Шкала найменувань. В ній число чи інших символ, які можна впорядковувати використовується лише для ідентифікації системи, демонстрації її відмінності від інших систем (приклад – сторінки в книзі).

2) Рангова (порядкова) шкала. Встановлює строгу відповідність значень шкали і критерію. Числові значення дозволяють чітко судити про “краще” та “гірше” значення, але не про різницю та кратність цих оцінок (приклад – оцінки знань).

3) Інтервальна шкала. Числові оцінки цієї шкали дозволяють не тільки порівнювати їх, а й виконувати лінійні перетворення. Найпоширеніші з таких шкал – вимірювання температури та часу – шкали Кельвіна, Цельсія та Фаренгейта, або ж Юліанський, Григоріанський та іудейський календарі – відрізняються одна від одної на 2 константи: початок відліку та одиницю виміру.

4) Шкала відношень – подальший розвиток інтервальної шкали, над її значеннями припустимі як лінійні перетворення, так і більш складні математичні операції. Виміри та такій шкалі дозволяють порівнювати не тільки самі значення критеріїв, а і будь-які їх співвідношення. Шкала критеріїв використовується лише для величин, що мають об’єктивну нульову точку (початок відліку).

#### **4. Генерація рішень - альтернатив.**

Найскладнішим і найбільш творчим етапом системного аналізу є формування множини альтернатив. Щоб осмислено згенерувати максимальну кількість альтернатив, можна використовувати різні методи:

пошук альтернатив у літературі;

залучення декількох кваліфікованих фахівців, які мають відповідний досвід і підготовку (експертів);

збільшення кількості альтернатив за рахунок їх поєднання,

формування проміжних варіантів з раніше запропонованих альтернатив;

модифікація доступних альтернатив;

включення альтернатив до навпаки запропонованої, зокрема, і «нульової» альтернативи («нічого не робити»);

інтерв'ювання зацікавлених осіб;

генерація альтернатив, розрахованих на різні часові проміжки (довгострокові, короткострокові, аварійні) тощо.

На практиці добре зарекомендували себе такі організаційні форми генерування альтернатив, як метод мозкового штурму, синергетика (пошук аналогій, асоціативне мислення), метод сценаріїв та ділові ігри.

#### **5. Добування знань чи вимог**

Добування знань чи вимог – формування у системного аналітика найбільш повного з можливих уявлень про предметну область і способи прийняття рішення в ній.

На цьому перенесення компетентності від стейкхолдерів до системного аналітика, з використанням різних методів:

- аналіз текстів;
- діалоги (вважається найбільш ефективним);
- експертні гри;
- лекції (вважається найменш ефективним);
- дискусії;
- інтерв'ю;
- спостереження та інші.

Підсумок: вся сукупність вимог усіх стейкхолдерів до майбутнього програмного забезпечення

## **6. Структурування знань чи вимог**

**Структурування (або концептуалізація)** знань або вимог – розробка неформального опису інформації про предметну область у вигляді графа, таблиці, діаграми або тексту, що відображає основні концепції і взаємозв'язки між поняттями предметної області.

**На етапі структурування виявляється структура** отриманої інформації про предметну область, тобто визначаються:

- список основних понять (термінів) та їх властивостей;
- зв'язки між поняттями;
- структура вхідної та вихідної інформації;
- стратегія прийняття рішень та її обмеження.

Результатом є сценарії поведінки користувача у всіх можливих ситуаціях (user stories) та варіанти використання майбутнього програмного забезпечення (use cases).

## **7. Формалізація вимог**

Формалізація вимог – розробка технічного завдання у вигляді моделей представлення, які з одного боку, відповідають структурі інформації про предметну область, а з іншого – дозволяють реалізувати прототип системи на наступному етапі програмної реалізації.

Все частіше на цьому етапі використовується симбіоз моделей представлення, адже докладно складну предметну область складно описати якоюсь однією стандартною моделлю.

На даній стадії будується формалізоване уявлення концептів предметної області на основі вибраної моделі представлення. Традиційно на цьому етапі використовуються:

- Параметричні діаграми;
- Уніфіковані мови моделювання (UML)
- Об'єктно-орієнтовані мови, засновані на ієрархії класів та об'єктів

## **8. Вибір альтернатив**

При використанні будь-якого з методів прийняття рішень необхідно задати:

- модель (чи множину моделей) функціонування системи;
- множину поставлених цілей (мотивів);

- множину ресурсів, що є в наявності (можливостей);
- множину ситуацій, що складається з пар стану ОПР та стану середовища і об'єкта управління (суб'єктивний стан і об'єктивний стан);
- множину припустимих рішень.

Вибір того чи іншого рішення робиться, як правило, на основі його оцінок. Тобто, кожне рішення з множини рішень має відображення на множині оцінок, впорядкованих за ступенем бажаності (критеріїв), що відносяться до множини наслідків рішення, яке теж є відображенням множини рішень

### **9. Моделювання роботи системи**

**Модель** - це штучний, рукотворний об'єкт будь-якого характеру (умоглядний або матеріально реалізований), який замінює або відтворює досліджуваний об'єкт. (Тарасенко)

**Модель** - це уявлення об'єкта, системи або поняття в якійсь формі, відмінній від форми її фактичного існування. (Згуровський)

Основний принцип моделювання: заміна існуючого об'єкта (системи) його аналогом дозволяє виділити приховану від спостереження суть оригіналу.

Моделювання буває математичне та імітаційне. Про нього – буде далі докладніше

### **10. Формування технічного завдання на проєкт.**

ТЗ повинно включати в себе такі відомості про об'єкт розробки:

1. Найменування об'єкта розробки, та область застосування.
2. Підстава для розробки (замовник, порядок фінансування).
3. Мета розробки (з переліком критеріїв).
4. Джерела розробки (стейкхолдери).
5. Технічні вимоги (докладно).
6. Економічні показники (витрати, ефект, термін окупності).
7. Порядок контролю і приймання об'єкта.

### **11. Аналіз, оцінка і прогнозування роботи готового проєкту**

I. Критерії користувачів:

- зрозумілість і «прозорість» роботи системи,
- зручність інтерфейсів.

II. Критерії експертів (фахівців у предметній області):

- оцінка дій системи в порівнянні з власними рішеннями,
- оцінка якості підсистеми допомоги і пояснення.

III. Критерії колективу розробників:

- ефективність реалізації (продуктивність, час відгуку),
- дизайн,
- широта охоплення предметної області,
- відсутність протиріч,
- кількість тупикових ситуацій (система не може знайти рішення),
- аналіз чутливості програми до незначних змін в представленні знань, вагових коефіцієнтах та інших.

**1.5. Значення системного аналізу для прийняття рішень.** Системний аналіз є важливим інструментом для прийняття рішень, оскільки він дозволяє:

Оцінити ситуацію з різних точок зору.

Визначити основні проблеми і можливі варіанти рішень.

Оптимізувати ресурси та процеси для досягнення максимального ефекту.

Прогнозувати наслідки прийнятих рішень.

## **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Дайте формальне визначення «система», «елемент системи» і «середовище» та наведіть по одному прикладу для технічної і соціально-економічної системи.

2. Що таке система: перерахувати основні ознаки системи та пояснити їх на прикладі технічної системи.

3. Порівняйте кількісні та якісні методи системного аналізу: у яких випадках кожен із них є більш доцільним? Наведіть приклади методів.

4. Опишіть алгоритм системного аналізу в процесі розробки програмного забезпечення: основні етапи та їхні взаємозв'язки.

6. Які обмеження має системний підхід при аналізі складних проектів і як їх можна мінімізувати (організаційно та методологічно)?

## **Тема 2. Структурний системний аналіз**

**2.1. Цілі і завдання структурного аналізу, визначення графа, види графів.**

Однією з найважливіших характеристик будь-якої системи є її структура. Під структурою системи, будемо розуміти сукупність елементів та зв'язків між ними, які визначаються виходячи з функцій та цілей, поставлених перед системою [5].

До одного з головних завдань структурного аналізу систем відноситься побудова наочної формальної моделі, що відображає існуючу сукупність відносин елементів як між собою, так і із зовнішнім середовищем. Структурна модель системи найчастіше багаторівнева, причому конкретизація структури дається на стільки рівнях, скільки їх потрібно створення повного уявлення про досліджувані властивості системи.

Загальна мета структурного аналізу полягає в тому, щоб, виходячи із заданого опису елементів системи та безпосередніх зв'язків між ними отримати висновок про структурні властивості системи в цілому та основні її підсистеми. При вирішенні практичних завдань структурного аналізу систем доцільно приймати три рівні опису зв'язків між елементами: 1) наявність зв'язку; 2) напрямок зв'язку;

На першому рівні, коли виходять лише з наявності або відсутності зв'язків між елементами, система, що вивчається, може відповідати неорієнтований граф, вершинами якого є елементи системи, а ребрами - існуючі безпосередні зв'язки між елементами. Основні завдання структурного аналізу цього рівні зводяться, наприклад, до наступних:

- визначення зв'язності (цілісності) системи; якщо система не є зв'язковою, то ставлять завдання виділення ізольованих зв'язкових підсистем зі списками елементів, що входять до них;
- виділення циклів (контурів);
- визначення мінімальних і максимальних перерізів (за кількістю зв'язків), що розділяють елементи та підсистеми один від одного.

З другого краю рівні, коли заданий напрям зв'язку, системі відповідає орієнтований граф, напрями дуг якого збігаються з напрямками зв'язків. На цьому рівні результати структурного аналізу виявляються змістовнішими.

До них, наприклад, можна віднести:

- визначення зв'язності системи;
- топологічну декомпозицію з виділенням сильно зв'язкових підсистем;
- перерахування вхідних та вихідних полюсів та відповідно до цього виділення вузлів прийому та видачі інформації;
- виділення рівнів у структурі та визначення їх взаємозв'язку;
- визначення максимальних та мінімальних шляхів;
- визначення характеристик топологічної значущості елементів;
- отримання інформації про слабкі місця структури та ін.

На третьому рівні опису враховуються не тільки наявність та спрямованість зв'язку, але і його "вага", "сила" в деяких одиницях виміру. Тут вже вдається уточнити топологічні оцінки структурних характеристик системи, що отримуються на перших двох рівнях, а також вирішити деякі додаткові завдання, наприклад, виділення типових структурних конфігурацій при багатому режимному характері функціонування системи.

Основи подібної формалізації опису структур закладено теоретично графів.

**Визначення графа, види графів.** Нехай визначено кілька елементів  $V$ . Граф  $G = G(V)$  вважається певним, якщо задане деяке сімейство поєднань елементів або пар виду  $E = (a, b)$ , де  $a, b \in V$ , що вказує, які елементи вважаються пов'язаними. Відповідно до геометричної інтерпретації пара  $E = (a, b)$  називається ребром, а елементи  $a, b$  - кінцевими точками ребра чи вершинами. Якщо порядок розташування кінців байдужий, тобто. якщо  $E = (a, b) = (b, a)$ , то кажуть, що  $E$  неорієнтоване ребро; якщо цей порядок важливий, то  $E$  називають орієнтованим ребром - дугою; у своїй, а називають початковою вершиною,  $a, b$  - кінцевою.

Теоретично графів прийнято також така термінологія. Ребро  $E$  *інцидентне* вершин  $a, b$ , а вершини  $a, b$  *інцидентні* ребру  $E$ . Граф, складений лише з неорієнтованих ребер, називається *неорієнтованим*, а граф, складений

з орієнтованих ребер, - *орієнтованим*. Графи, у яких частина ребер орієнтована, частина – неорієнтована, називаються змішаними.

Неорієнтований граф може бути перетворений на орієнтований за допомогою процесу подвоєння, що полягає в заміні кожного ребра  $E$  двома ребрами з тими ж кінцями та приписування їм протилежних орієнтацій. Граф називається *кінцевим*, якщо число ребер звичайно, і *нескінченим* - інакше. Граф, що складається із ізольованої вершини, називається *нуль-графом*, а граф, ребрами якого є всілякі пари для двох різних вершин  $b$  з  $V$  - *повним графом*. У орієнтованому повному графі є пари ребер по одному у кожному напрямі, що з'єднують будь-які дві різні вершини  $(a, b)$ .

## 2.2. Методи формалізованого завдання графа.

**А. Графічне уявлення.** Цей спосіб є найбільш наочною формою уявлення відносин між елементами.

**Б. Матричне уявлення.** Існують різні форми матричного уявлення графа  $G = G(V)$ . Матриця суміжності вершин для неорієнтованого графа має вигляд  $A = \| a_{ij} \|_{n,n}$ , де  $n$  - число вершин у графі, а елементи  $a_{ij}$  визначаються таким чином:

$$a_{ij} = \begin{cases} 1, & \text{при наявності зв'язку;} \\ 0 & \text{при відсутності зв'язку.} \end{cases}$$

Передбачається, що нумерація вершин графа вже проведена. Для неорієнтованого графа матриця суміжності симетрична.

У матриці суміжності вершин для орієнтованого графа елементи визначаються так:

$$a_{ij} = \begin{cases} 1, & \text{якщо із вершини } i \text{ можна перейти в вершину } j; \\ 0 & \text{в протилежному випадку.} \end{cases}$$

Вигляд матриці суміжності орієнтованих графів істотно залежить від обраного порядку нумерації вершин  $i$ , вибравши певний принцип нумерації вершин для деяких видів графів (без контурів), можна звести матрицю  $A$  до трикутного вигляду, де  $a_{ij} = 0$  при  $j > i$ .

Аналогічним чином можна побудувати відповідні матриці суміжності ребер, де елемент  $a_{ij}$  вважається ненульовим, якщо ребра графа мають загальну вершину.

Матриця інцидентності  $B = \| b_{ij} \|_{n,m}$ .  $i = \overline{1,n}$ ;  $j = \overline{1,m}$ , де  $n$  – число вершин, а  $m$  – число ребер, визначається для неорієнтованого графа як

$$b_{ij} = \begin{cases} 1, & \text{якщо } i \text{ – та вершина інцидентна даному } j \text{ – му ребру (є зв'язок);} \\ 0, & \text{якщо нема зв'язку.} \end{cases}$$

а для орієнтованого графа:

$$b_{ij} = \begin{cases} \text{якщо } i \text{ – та вершина є початок } j \text{ – го ребра;} \\ -1, & \text{якщо } i \text{ – та вершина є кінець } j \text{ – го ребра;} \\ \text{якщо } i \text{ – та вершина не інцидентна } j \text{ – му ребру.} \end{cases}$$

В. Множинне представлення. В цьому випадку для орієнтованого графа  $G(V)$  задається безліч вершин  $V$  і відповідність  $G$ , яке показує, як пов'язані між собою вершини. Відповідність  $G$  у цьому випадку називається відображенням множини  $V$  в  $V$ . Для кожної вершини  $i$  відповідність визначає безліч вершин  $G(i)$ , в яку можна безпосередньо потрапити з вершини  $i$ . У ряді випадків  $G(i)$  називається безліччю правих інцидентів.

Безліч  $G^{-1}(i)$  визначає всі вершини, з яких можна безпосередньо потрапити у вершину  $i$ , і тому називається зворотним відповідністю (відображенням). За аналогією з  $G(i)$  безліч  $G^{-1}(i)$  називається безліччю лівих інцидентів.

Приклад 2.1. Подання структури системи на рис. 2.1а, а вигляді графа рис. 2.1 б.

Матриця суміжності вершин  $A$ :

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Матриця інцидентів  $B$ .

$$B = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Множинне завдання структури системи:  $G(1) = (2,3)$ ;  $G(2) = \emptyset$ ;  $G(3) = (5,4)$ ;  $G(4) = (2)$ ;  $G(5) = (1,2)$ ;  $G^{-1}(1) = (5)$ ;  $G^{-1}(2) = (1,5,4)$ ;  $G^{-1}(3) = (1)$ ;  $G^{-1}(4) = (3)$ ;  $G^{-1}(5) = (3)$ .

Якби граф на рис. 2.1 б був неорієнтованим, матриця суміжності мала б вигляд:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Аналогічно можна скласти матрицю інцидентів для неорієнтованого графа.

**Визначення часткового графа та підграфа.** Граф  $H$  називається частковим графом графа  $G$ , якщо його безліч вершин  $V(H)$  міститься у

множині вершин  $V$  графа  $G$  і всі ребра графа  $H \in$  ребрами графа  $G$  (див. рис.2.1, в). Граф  $G_1(D)$  називається підграфом графа  $G(V)$ , якщо безліч вершин  $D$  міститься у множині вершин  $V$  і ребра графа  $G_1$  є всіма ребрами графа  $G$ , є всіма ребрами графа  $D$ . кінці яких лежать у множині  $D$  (рис.2.1, а ). Якщо  $D = V$ , то  $G(D) = G(V)$ .

**Визначення ланцюга, шляху, циклу, контуру.** Ланцюгом називається така послідовність ребер  $E_0, E_1, \dots, E_{k-1}, E_k, \dots$ , коли кожне ребро  $E_{k-1}$  стикається одним з кінців з ребром  $E_k$ ; ланцюг можна позначити послідовністю вершин, які вона містить.

Наприклад, для графа на рис. 2.2 послідовності вершин  $(1, 2, 3, 4, 5), (1, 3, 5)$  є ланцюгами. Поняття ланцюга зазвичай використовується для неорієнтованих графів.

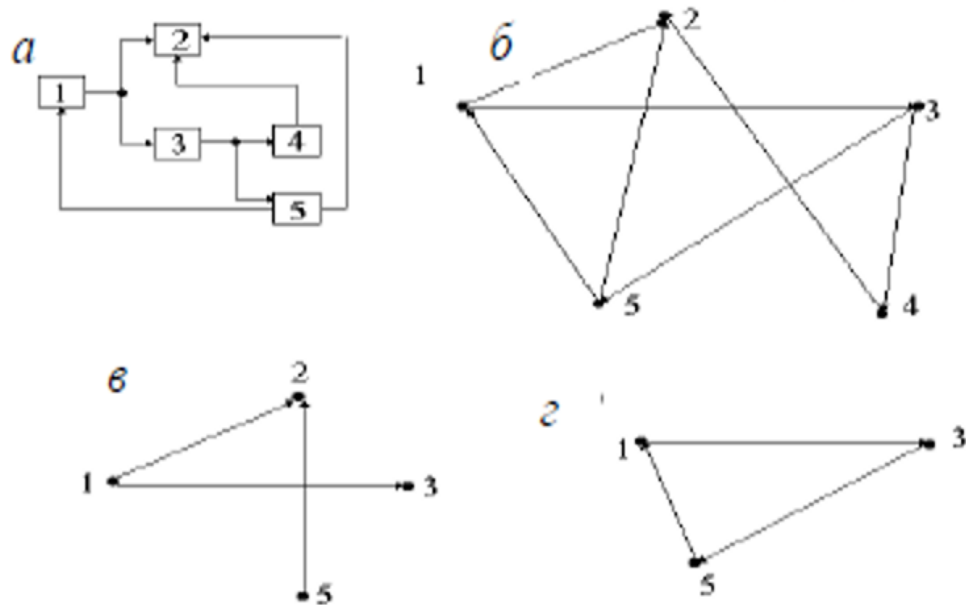


Рис. 2.1. Відображення структури системи як орієнтованого графа.  
 а -структура системи; б – її граф; в – частковий граф; г - підграф.

*Шляхом* називається така послідовність дуг, коли кінець кожної попередньої дуги збігається з початком наступної; наприклад, для графа на рис. 2.2 б послідовність дуг  $(1, 3), (3, 5), (5, 1)$  є шляхом. Поняття шляху зазвичай використовується для орієнтованих графів.

Циклом називається такий кінцевий ланцюг, який починається і закінчується в одній вершині; наприклад, для графа на рис. 2.2 ланцюг  $(1, 4, 3, 1)$  є циклом; дане поняття має сенс лише неорієнтованих графів.

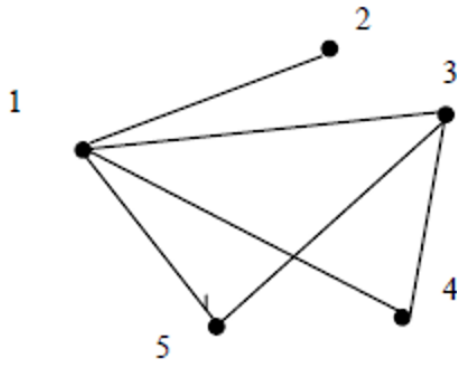


Рис. 2.2. Неорієнтований граф.

*Контуром* називається такий кінцевий шлях, у якого початкова вершина першої дуги збігається з кінцевою вершиною останньої дуги; для графа на рис. 2.1 б послідовність дуг (1, 3), (3, 5), (5,1) - контур.

*Довжиною шляху* називають число ребер (дуг), що входять у ланцюг (шлях) графа.

Матриця суміжності вершин  $A$  є матрицею безпосередніх шляхів графа, що мають довжину, що дорівнює одиниці. Загальна кількість транзитних шляхів від вершини до вершини  $j$  довжиною  $k$  може бути отримана в результаті зведення в  $k$ -у ступінь матриці  $A$ .

Елемент матриці  $A^k - a_{ij}^{(k)}$  - визначає кількість шляхів довжиною  $k$  від вершини  $i$  до вершини  $j$ .

**Ступінь вершини.** Число ребер, інцидентних вершині неорієнтованого графа, називають ступенем вершини та позначають  $p(i)$ . Число дуг орієнтованого графа, які мають своєю початковою вершиною вершину  $i$ , називають напівступенем результату вершини  $i$  та позначають  $p^+(i)$ . Аналогічно число дуг, які мають своєю кінцевою вершиною вершину  $j$ , називають півступенем заходу вершини  $j$  та позначають  $p^-(j)$ . Цілком очевидно, що сума напівступеня заходу всіх вершин графа, а також сума напівступеня результату всіх вершин дорівнює загальній кількості дуг орієнтованого графа  $G(V)$ . тобто.

$$\sum_{i=1}^n p^+(i) = \sum_{j=1}^n p^-(j) = m, \quad (2.1)$$

де  $m$  - число дуг графа, а  $n$  - число його вершин.

Для неорієнтованого графа маємо

$$\frac{1}{2} \sum_{i=1}^n p(i) = m, \quad (2.2)$$

**Зв'язність графа.** Для неорієнтованих графів вводиться поняття слабкої зв'язності, чи навіть зв'язності. Граф  $G(V)$  називається слабо зв'язним (зв'язковим), якщо для будь-яких вершин графа  $i$  і  $j$  існує ланцюг з вершини  $i$  у вершину  $j$ .

Для орієнтованих графів вводиться додатково поняття сильної зв'язності. Граф  $G(V)$  називається дуже зв'язковим, якщо для будь-яких вершин графа  $i, j$  існує шлях з вершини  $i$  до вершини  $j$ .

Граф, зображений на рис. 2.2, є слабо зв'язковим; граф на рис. 2.3 а - сильно зв'язний, що розпадається на два сильно зв'язних підграфа.

Теоретично графів показано, що будь-який неорієнтований граф може бути розкладений на сукупність зв'язкових підграфів, а будь-який орієнтований граф - на сукупність сильно зв'язкових підграфів.

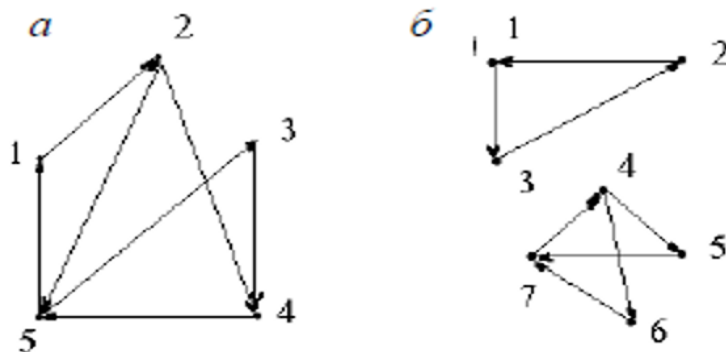


Рис. 2.3. Сильно зв'язаний граф (а) і незв'язний граф, що розпадається на зв'язані підграфи (б).

**Порядкова функція на графі. Поняття рівня.** Метою введення порядкової функції на графі без контурів є розбиття безлічі вершин графа на підмножини, що не перетинаються, упорядковані так, що якщо вершина входить в підмножину з номером  $i$ , то наступна за нею вершина - в підмножину з номером, великим  $i$ . Отримані непересічні підмножини називаються рівнями. Алгоритм упорядкування зводиться до наступного.

1. До підмножини нульового рівня  $N_0$  включаються всі вершини  $i$ , у яких

$$G^{-1}(i) = \emptyset \text{ (порожня підмножина).}$$

Здійснюється послідовна нумерація вершин:  $1, 2, \dots, l$ .

2. До підмножини першого рівня  $N_1$  включаються всі вершини  $i$ , у яких  $G^{-1}(i) \subset N_0$ .

Проводиться послідовна нумерація вершин:  $l + 1, l + 2, \dots, l + r$ .

3. До підмножини другого рівня  $N_2$  включаються всі вершини  $i$ , у яких  $G^{-1}(i) \subset (N_0 \cup N_1)$ .

Реалізується послідовна нумерація вершин:  $l + r + 1, l + r + 2, \dots, l + r + p$ .

4. До підмножини третього рівня  $N_3$  включаються всі вершини  $i$ , у яких  $G^{-1}(i) \subset (N_0 \cup N_1 \cup N_2)$ , після чого проводиться подальша нумерація вершин.

Цей процес ведеться до тих пір, поки не будуть пронумеровані всі вершини графа.

На рис. 2.4 показані невпорядкований та впорядкований графи без контурів. Для графа за наявності контурів відповідно до розглянутого вище алгоритмом виділяються спочатку сильно зв'язні підграфи, що утворюють класи, а далі, оскільки граф класів немає контурів, у ньому можна зробити впорядкування і запровадити поняття рівнів.

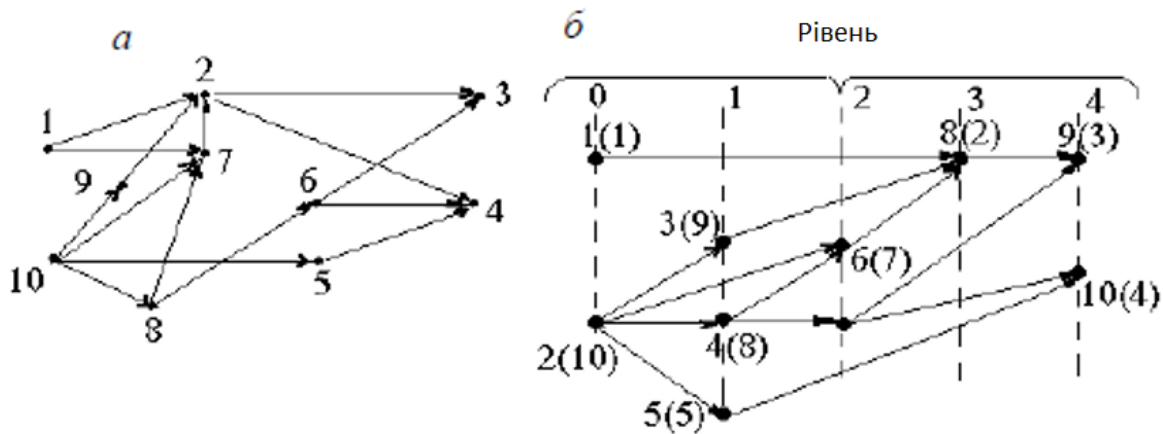


Рис. 2.4. Введення порядкових функцій на графі.  
 а – невпорядкований граф; б – упорядкований.

**Числова функція на графі.** Числовою функцією на графі зазвичай задають або на вершинах, або на дугах (ребрах) графа.

Числова функція на вершинах графа вважається заданою, якщо кожній  $i$ -й вершині  $a_i$  графа  $G(V)$ ,  $a_i \in V$ , ставиться у відповідність деяке число  $I_i = I(a_i)$  з деякої множини  $L$ .

Числова функція на дугах (ребрах) для орієнтованого (неорієнтованого) графа вважається заданою, якщо кожній дузі  $(a_i, a_j)$  або ребру ставиться у відповідність число  $q = q(a_i, a_j)$  з деякої множини  $Q$ . У деяких випадках числова функція на графі задається комбінованим способом як у вершинах, і на дугах.

Значення функції шляху  $S$  через вершини  $a_1, a_2, \dots, a_j, \dots$  ( $a_j \in S$ ) при заданні числової функції на вершинах графа визначається відповідно до адитивної форми

$$I_S = \sum_{a_i \in S} I(a_i), \quad (2.3)$$

або з мультиплікативною формою

$$I_S = \prod_{a_i \in S} I(a_i), \quad (2.4)$$

Аналогічним чином визначається значення функції по дорозі через вершини  $a_1, a_2, \dots, a_i, \dots$  при заданні числової функції на дугах (ребрах) графа:

$$q_S = \sum_{(a_i, a_j) \in S} q(a_i, a_j), \quad (2.5)$$

$$q_S = \prod_{(a_i, a_j) \in S} q(a_i, a_j), \quad (2.6)$$

Відповідно до даних визначення може бути поставлена задача знаходження шляхів через безліч вершин (дуг), що володіють певною властивістю, з максимальним (мінімальним) значенням числової функції. у вигляді завдання динамічного програмування. Наприклад, визначення максимального шляху на графі без контурів відповідно до (2.5) реалізується на основі наступного функціонального рівняння динамічного програмування:

$$q_S^{max}(a_i, a_j) = \max_{a_i \in G^{-1}(a_j)} [q_S^{max}(a_1, a_i) + q(a_i, a_j)], \quad (2.7)$$

де  $q_S^{max}(a_1, a_j)$  - максимальне значення функції на шляхах  $S$  з деякої початкової вершини  $a_1$  у вершину  $a_j$  (аналогічно  $q_S^{max}(a_1, a_j)$ );

$G^{-1}(a_j)$  - безліч лівих інцидентів для вершини  $a_j$ ;  $q(a_i a_j)$  – значення функції на дузі  $(a_i a_j)$ .

З використанням даного функціонального рівняння зазвичай передбачається, що це вершини у графі впорядковані.

Визначення максимальних та мінімальних шляхів має численні додатки при проектуванні систем: у задачах мережного та календарного планування для визначення критичного шляху, у транспортних задачах, у завданнях контролю та технічної діагностики та ін.

Приклад 2.2. Для графа на рис. 2.5 знайти максимальний шлях із вершини  $a_1$  у вершину  $a_7$ .

Для вершини  $a_1$  приймаємо  $q_S^{max}(a_1 a_1) = 0$ .

Для вершин  $a_2, a_3, a_4$ :  $q_S^{max}(a_1 a_2) = 2$ ;  $q_S^{max}(a_1 a_3) = 3$ ;  $q_S^{max}(a_1 a_4) = 5$ .

Для вершини  $a_5$ :  $q_S^{max}(a_1 a_5) = \max(2 + 1; 3 + 2; 4 + 1) = 5$ .

Для вершини  $a_6$ :  $q_S^{max}(a_1 a_6) = \max(3 + 3; 4 + 4) = 8$ .

Для вершини  $a_7$ :  $q_S^{max}(a_1 a_7) = \max(5 + 2; 8 + 1) = 9$ .

Значення функції максимальному шляху дорівнює дев'яти, а сам шлях виділено на рис. 2.5 жирною лінією.

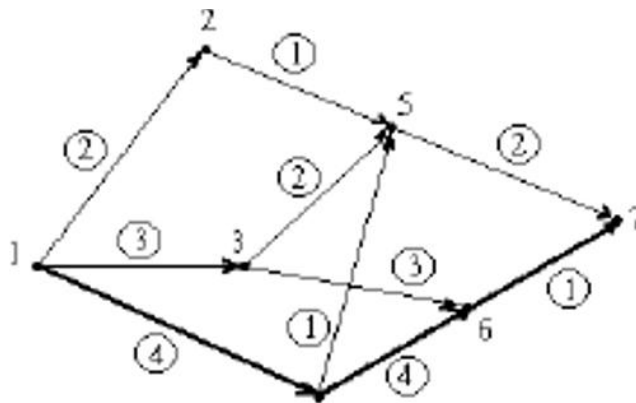


Рис. 2.5. До визначення максимального шляху графі.

### 2.3. Структурно-топологічні властивості систем.

При проведенні структурного аналізу систем часто-густо необхідно вміти визначати деякі структурні характеристики систем і давати їм кількісну оцінку. Доцільність визначення таких параметрів у тому, що вже на ранній стадії проектування виникає необхідність оцінювати якість структури системи та її елементів з позицій системного підходу. Розглянемо деякі з таких параметрів.

**Складність структури.** Цей кількісний параметр дозволяє виявити наявність обривів у структурі, висячі вершини та ін. Найбільш повно кількісно зв'язність елементів орієнтованого графа визначається матрицею зв'язності  $C = \|c_{ij}\|$ .

Елементи матриці можна обчислити на основі матриці  $A_\Sigma = \sum_{k=1}^n A^k$ .

Елемент  $c_{ij} = 1$ , якщо  $a_{ij}^\Sigma \geq 1$ ;  $c_{ij} = 0$ , якщо  $a_{ij}^\Sigma = 0$ .

Для неорієнтованих графів зв'язність всіх елементів у структурі відповідає виконанню наступної умови:

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \geq n - 1, i \neq j, \quad (2.8)$$

Права частина нерівності визначає необхідну мінімальну кількість зв'язків у структурі неорієнтованого графа, що містить  $n$  вершин.

**Структурна надмірність.** Структурний параметр, що відображає перевищення загальної кількості зв'язків над мінімально необхідним, називається структурною надмірністю  $R$ , яка визначається таким чином:

$$R = \frac{1}{2} \left[ \sum_{i=1}^n \sum_{j=1}^n a_{ij} \right] \frac{1}{n-1} - 1, \quad (2.9)$$

Для систем з максимальною надмірністю, що мають структуру типу "повний граф" (рис. 1.4),  $R > 0$ ; для систем з мінімальною надмірністю  $R = 0$ ; для нескладних систем  $R < 0$ .

Рівномірний розподіл зв'язків у структурі неорієнтованого графа, що має  $m$  ребер та  $n$  вершин, характеризується середнім ступенем вершини  $\bar{\rho} = 2m/n$ .

Тоді, ввівши поняття відхилення  $\rho_i - \bar{\rho}$ , де  $\rho_i$  - дійсна міра вершини заданого графа, можна визначити квадратичне відхилення заданого розподілу ступенів вершин від рівномірного:

$$\begin{aligned} \varepsilon^2 &= \sum_{i=1}^n (\rho_i - \bar{\rho})^2 = \sum_{i=1}^n \rho_i^2 - 2\bar{\rho} \sum_{i=1}^n \rho_i + 4m^2/n^2 =, \\ &= \sum_{i=1}^n \rho_i^2 - 8m^2/n + 42/n^2 \end{aligned} \quad (2.10)$$

Цей показник характеризує недовикористання можливостей заданої структури, що має  $m$  ребер і  $n$  вершин, у досягненні максимальної зв'язності.

**Структурна компактність.** Для кількісної оцінки структурної компактності вводиться параметр, який відбиває близькість елементів між собою. Близькість двох елементів  $i$  і  $j$  між собою визначатимемо через мінімальну довжину шляху для орієнтованого графа (ланцюга - для неорієнтованого)  $d_{ij}$ . Тоді величина

$$Q = \sum_{i=1}^n \sum_{j=1}^n d_{ij}, i \neq j, \quad (2.11)$$

відбиває загальну структурну близькість елементів між собою у системі.

Для кількісної оцінки структурної компактності часто використовують відносний показник:

$$Q_{\text{від}} = Q / Q^1_{\text{пг}}, i \neq j, \quad (2.12)$$

де  $Q_{\text{пг}} = n(n-1)$  - значення компактності для структури системи типу "повний граф".

Структурну компактність можна характеризувати й іншим показником - діаметром структури:

$$d = \max_{ij} d_{ij}, \quad (2.13)$$

**Ступінь централізації у структурі.** Для кількісної оцінки ступеня централізації у структурі використовується поняття індексу центральності:

$$\delta = (n-1)(2Z_{\text{max}} - n) / (Z_{\text{max}}(n-2)), \quad (2.14)$$

де  $Z_{\text{max}}$  - максимальне значення величини.

$$Z_i = \frac{Q}{2} (\sum_{j=1}^n d_{ij})^{-1}, \quad i = \overline{1, n}, \quad (2.15)$$

Для структур систем, що мають максимальний ступінь централізації (рис. 2.6, в),  $\delta = 1$ ;

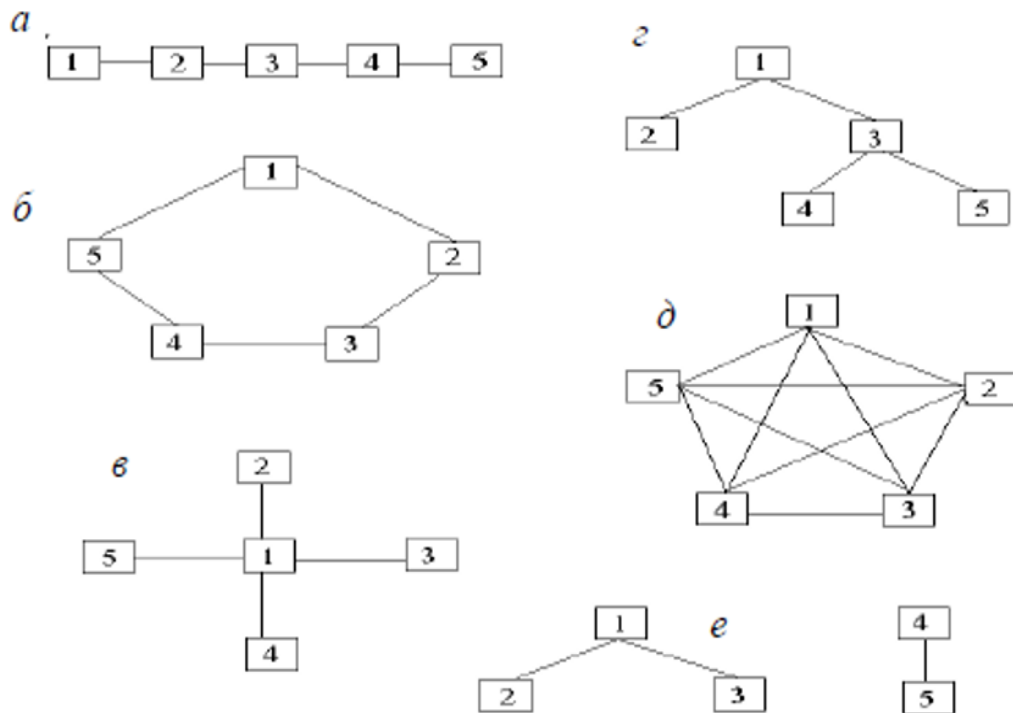


Рис. 2.6. Основні види структур:  
а – послідовна (“ланцюжкова”), б – кільцева,  
в – радіальна, д – повний граф, е – незв’язна.

**Ранг елемент** використовується при поданні системи у вигляді орієнтованого графа. Ця характеристика дозволяє розподілити елементи системи у порядку їх значимості. Значимість елемента визначається лише числом зв’язків даного елемента коїться з іншими. Чим вище ранг елемента, тим більше він пов’язаний з іншими елементами системи. Приблизно ранг елемента можна оцінити за формулою:

$$r_i \approx \frac{\sum_{j=1}^n a_{ij}^{(k)}}{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^{(k)}}, \quad i \neq j, \quad (2.16)$$

де  $a_{ij}(k)$  – елементи матриці  $A_k$ ,  $k = 3 \div 4$ .

**Приклад 2.3.** Введені кількісні характеристики можуть бути використані при порівняльній оцінці топологічних властивостей структур систем. З погляду топології внутрішніх зв’язків виділяють основні види структур, показані на рис. 2.6. Розглянемо кількісні оцінки топологічних властивостей цих структур, які у табл. 2.1.

Таблиця 2.1 – Кількісні оцінки топологічних властивостей

Тип структури (За рис. 2.6)	Показник				
	$R$	$\varepsilon^2$	$Q$	$d$	$\delta$
$a$	0	1,2	1,0	4	0,7
$b$	0,25	0	0,5	2	0
$v$	0	7,2	0,6	2	1,0
$z$	0	3,2	0,7	3	0,7
$\partial$	1,5	0	0	1	0
$e$	-0,25	-	-	-	-

З табл. 2.1 видно:

– для незв'язних структур  $R < 0$ ; для структур без надмірності (послідовна, радіальна, деревоподібна)  $R = 0$ ; для структур з надмірністю зв'язків (кільцева, типу “повний граф”)  $R > 0$ ;

– Структури з  $R = 0$  (послідовна, радіальна, деревоподібна) розрізняються за показником  $\varepsilon^2$ ; найбільша нерівномірність зв'язків у радіальної структури:

– найбільшу близькість елементів (показник  $Q$ ) має структура типу “повний граф”; найменшу - послідовна: у радіальної та кільцевої структур, невиразних за показником  $d$ , різні значення  $Q$ ;

– радіальна та деревоподібна структури, що мають однакові або близькі значення  $R$ ,  $Q$ ,  $d$ , значно різняться за показниками  $\varepsilon^2$  та  $\delta$ , що відповідає фізичному змісту, бо відхід від повної централізації у структурі веде до великої рівномірності розподілу зв'язків за елементами.

Розглянуті вище структурні характеристики були отримані тільки на основі інформації про склад елементів та їх зв'язки. Подальший розвиток аналізу структур систем то, можливо заснований на обліку неструктурної інформації з допомогою запровадження числових функцій на графах. Це суттєво збагачує можливості структурного аналізу.

## ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ

1. Що таке граф у контексті структурного аналізу системи? Назвіть і проілюструйте три види графів, які використовуються для моделювання систем.

2. Опишіть формалізований підхід до завдання графа (вершини, ребра, ваги). Як задаються обмеження і цілі на графі?

3. Що таке топологічні властивості графа (зв'язність, діаметр, центр) і яку інформацію вони дають про систему?

4. Наведіть приклад задачі оптимізації на графі, що відображає структурні властивості системи (короткий опис задачі, ціль, обмеження).

5. Як структурний аналіз допомагає виявляти «вузькі місця» системи - опишіть метод (наприклад, аналіз потоків або критичний шлях) і приклад застосування.

### **Тема 3. Опис систем на етапі аналізу**

#### **3.1. Огляд методики опису систем: функціональний, структурний, поведінковий аналіз**

В системному аналізі існують кілька підходів до опису систем, кожен з яких фокусується на різних аспектах її функціонування і взаємодії елементів. Три основні методики опису систем - це функціональний, структурний і поведінковий аналіз [3].

##### **Функціональний аналіз:**

Функціональний аналіз – це процес виявлення, опису та структурування функцій, які має виконувати система.

Основні аспекти:

**1. Функції та процеси.** Функції системи визначають ключові можливості та завдання, які система має виконувати. Вони відповідають на питання "що?" - які результати чи операції повинна забезпечувати система. Наприклад, для інформаційної системи функціями можуть бути:

Обробка даних;

Зберігання та пошук інформації;

Забезпечення безпеки даних;

Генерація звітів та аналітика.

Функції формулюється на основі вимог користувачів і бізнес-потреб. Вони є основою для створення технічних специфікацій, оскільки кожна функція має відповідати певним критеріям якості, ефективності та точності. Функції часто подаються на високому рівні абстракції, що дозволяє чітко зрозуміти, які можливості система повинна надати, без заглиблення у деталі реалізації. Це допомагає узгодити очікування замовника і розробника на ранніх етапах проекту.

Процеси розкривають механізми та послідовність дій, які використовуються для виконання визначених функцій. Вони відповідають на питання "як?" - яким чином система досягає необхідних результатів. Наприклад, для функції «обробка даних» можуть існувати процеси збору, очищення, аналізу та збереження даних. Функції часто розбиваються на підфункції, кожна з яких може мати свій набір процесів. Наприклад, функція «управління користувачами» може включати процеси реєстрації, автентифікації, управління профілями та відновлення паролів.

Взаємозв'язок функцій і процесів забезпечує, що кожна функція має відповідний набір процесів, які її реалізують. Це дозволяє системно підійти до розробки, де чітко видно, які конкретні дії забезпечують досягнення загальних цілей системи.

Вхідні та вихідні дані. Аналіз визначає, які дані або ресурси потрібні для виконання функцій, та які результати очікуються. Вхідні дані – це інформація, ресурси або сигнали, які надходять до системи для забезпечення виконання її функцій. Вони можуть бути як цифровими, так і фізичними, і їх тип залежить від специфіки системи.

Аналіз вхідних даних дозволяє:

1. Визначити потреби системи: які дані необхідні для виконання кожної функції? Наприклад, для системи управління клієнтами (CRM) потрібні дані про контакти, історію взаємодії та транзакції.

2. Оцінити якість і доступність ресурсів: чи є дані актуальними, повними та доступними в потрібний момент? Наприклад, при обробці фінансових операцій важливо мати доступ до оновленої інформації про курси валют або баланс рахунків.

3. Налагодити процес інтеграції: вхідні дані можуть надходити з різних джерел (бази даних, сенсори, інтерфейси користувача, веб-сервіси), і їх інтеграція забезпечує коректну роботу системи.

Ієрархія функцій. Часто система розбивається на підсистеми та окремі функції, що дозволяє спростити аналіз через функціональне декомпозицію.

Ієрархія функцій є однією з ключових концепцій функціонального аналізу, що дозволяє розбивати складну систему на більш дрібні, зрозумілі та керовані компоненти. Це робиться через процес функціональної декомпозиції, який має такі аспекти:

Визначення ієрархії функцій. На початковому етапі аналізу система описується як набір основних функцій, які відображають її загальні можливості. Ці функції відповідають ключовим завданням системи і формують базову концептуальну модель.

Декомпозиція на підфункції: кожна основна функція може бути розбита на підфункції або підсистеми, що дозволяє деталізувати конкретні аспекти її виконання. Така розбивка допомагає визначити, які саме компоненти системи відповідають за окремі завдання, та як вони взаємодіють між собою.

Структурування за рівнями: розбиття функцій на рівні дозволяє створити ієрархічну модель, де верхній рівень охоплює загальні функції, а нижчі рівні деталізують процеси та алгоритми їх реалізації. Це робить систему більш зрозумілою і полегшує подальшу розробку та підтримку.

Вимоги до системи. Функціональний аналіз тісно пов'язаний з формуванням функціональних вимог, що є основою для подальшого проектування

Функціональний аналіз безпосередньо пов'язаний із збором і структуруванням функціональних вимог, оскільки: На початковому етапі проводяться інтерв'ю, збираються дані від стейкхолдерів і аналізуються існуючі процеси. Це дозволяє визначити, які функції необхідні для задоволення потреб користувачів та бізнесу. За допомогою функціональної декомпозиції система розбивається на окремі функції та підфункції. Такий підхід дозволяє точно описати кожен частину системи, її ролі та взаємозв'язки.

Основні аспекти формування вимог

При формуванні вимог важливо враховувати кілька ключових аспектів:

**Однозначність:** Вимоги повинні бути сформульовані так, щоб не було місця для різночитань. Це забезпечує спільне розуміння між розробниками, аналітиками та замовниками.

**Повнота:** Опис системи має охоплювати всі критичні функції та сценарії використання. Неповнота може призвести до того, що деякі необхідні можливості системи залишаться не реалізованими.

**Вимірність:** Кожна вимога повинна містити критерії оцінки, за якими можна визначити, чи задовольняє система поставлені задачі.

**Тестованість:** Вимоги мають бути сформульовані таким чином, щоб їх можна було перевірити під час тестування системи

**2. Структурний аналіз:** Структурний аналіз фокусується на описі **як** система організована, які елементи складають систему і як ці елементи взаємодіють між собою. Цей аналіз спрямований на розробку **структури** системи, що дозволяє зрозуміти, як система розділяється на підсистеми, модулі або компоненти. Структурний аналіз дозволяє окреслити «скелет» системи, визначаючи, з яких основних компонентів вона складається, які модулі чи підсистеми існують і як вони організовані. Аналізуючи структуру, досліджують, як компоненти системи взаємодіють між собою, які існують зв'язки (наприклад, інформаційні потоки, зв'язки управління, залежності) та як ці зв'язки впливають на загальну функціональність системи. Подібно до функціональної декомпозиції, структурний аналіз часто будується за принципом ієрархічного розбиття. Це дозволяє розглянути систему на різних рівнях абстракції – від високорівневого бачення до детального опису окремих модулів.

**Опис архітектури системи:** Основна мета структурного аналізу – отримати чітку картину архітектури системи, що включає як логічну, так і фізичну структуру. Це сприяє кращому розумінню, як система організована для досягнення поставлених функцій. Аналіз дозволяє виявити критичні залежності між компонентами, що може допомогти в подальшій оптимізації, плануванні інтеграції та розробці. Результати структурного аналізу стають основою для створення різноманітних моделей, таких як блок-схеми, діаграми класів або компонентів, що візуально демонструють будову системи.

**Приклад:** В системі управління базою даних структурний аналіз буде визначати, як зберігаються та взаємодіють між собою дані, якими є таблиці, індекси, зв'язки між даними тощо.

**3. Поведінковий аналіз:** Поведінковий аналіз зосереджений на описі **як** система поводить себе в різних умовах. Це включає вивчення того, як система реагує на різні вхідні дані або події, як змінюється її стан у часі. Поведінковий аналіз також часто включає в себе моделювання сценаріїв або обробку різних варіантів ситуацій, які можуть виникнути в процесі роботи системи.

**Поведінковий аналіз** досліджує динаміку роботи системи, тобто як система реагує на зовнішні події, як змінюються її стани у часі та які процеси

відбуваються під час виконання функцій. Фокус на час та зміни: На відміну від структурного аналізу, який зосереджується на статичній архітектурі системи, поведінковий аналіз фіксує зміни, які відбуваються в системі у відповідь на різні впливи та події.

Основні аспекти поведінкового аналізу

Моделювання сценаріїв: Аналізує конкретні сценарії використання системи (use cases), які відображають послідовність дій користувача та реакцію системи на кожному етапі.

Вивчення станів системи: За допомогою діаграм станів (state diagrams) описується, як система переходить з одного стану в інший у відповідь на внутрішні або зовнішні події. Це допомагає виявити потенційні проблеми або вузькі місця в логіці роботи системи.

Послідовність взаємодій: Діаграми послідовності (sequence diagrams) та діаграми діяльності (activity diagrams) демонструють, в якій послідовності виконуються дії, як відбувається обмін повідомленнями між компонентами та які залежності існують між окремими процесами.

Відображення часових аспектів: Поведінковий аналіз враховує час виконання операцій, затримки, цикли повторення та асинхронні процеси, що є важливим для систем, де критичні часові параметри (наприклад, в режимі реального часу).

Приклад: У програмному забезпеченні поведінковий аналіз буде стосуватися того, як система реагує на введення користувача, як вона обробляє помилки, або як змінюється її поведінка в залежності від змін у середовищі.

### **3.2. Виявлення і управління вимогами**

Управління вимогами – це процес запису, аналізу, трасування, пріоритезації і узгодження вимог та контролю змін і доведення їх до зацікавлених сторін. Це безперервний процес протягом всього життя проекту [6].

Виявлені вимоги повинні характеризуватися:

- повнотою – вимога повинна описувати всю функціональність, яку необхідно реалізувати;
- коректністю – вимога повинна точно описувати бажану функціональність;
- виконуваністю – вимога повинна бути реалізована за даних умов, обмежень системи та операційного середовища;
- необхідністю – вимога повинна відображати можливість, яка дійсно необхідна користувачам, або яка потрібна для відповідності зовнішнім системним вимогам, або стандартам;
- пріоритетністю – вимоги повинні мати пріоритети для визначення послідовності їх реалізації;
- однозначністю – вимоги повинні трактуватися однаково всіма користувачами;

- можливістю бути перевіреними – необхідно розробити декілька тестів або застосувати інші прийоми для перевірки, щоб встановити, чи дійсно в програмній системі реалізована кожна вимога.

Неповні, неузгоджені, нездійсненні або двозначні вимоги не перевіряються.

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін.

Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектною групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проєкту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними.

Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покривають також і користувальницькі вимоги.

Відслідковування вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягнути стану повного відстеження. Відстежувати треба навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як: ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відслідковування вимог, реалізована в системі функція може бути простежена назад до людини або групи, яка замовляла її під час збору вимог. Ця особливість може, наприклад, використовуватися в процесі розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача. Відслідковування може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

### **Завдання управління вимогами.**

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та реалізацією і завершенням.

1) Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються

процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проєкту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили починають впливати на проєкт в середині циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху - гнучкість у мисленні та діях.

Результатом стадії дослідження є документ - специфікація вимог, схвалений усіма членами проєкту. Пізніше, в процесі розробки, цей документ виконуватиме роль обмежувача для запобігання розповзання меж проєкту або непотрібних змін. Оскільки проєкт удосконалюється, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи певні програмні інструменти. Ці інструменти керують вимогами, використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, експортуючи дані вимог.

2) Аналіз здійсненності. На стадії аналізу здійсненності визначається вартість вимог. Для користувацьких вимог поточна вартість роботи порівнюється з майбутньою вартістю використання проєкту. Задаються питання такі як: «Скільки нам зараз варті помилки проєктів?» Або, «Яка вартість реконструкції проєкту, якщо будуть помилки в експлуатації?». Фактично, потреба в додаткових експлуатаційних характеристиках часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо новий, більш простий в використанні продукт?»

Технічна вартість пов'язана з вартістю розробки та реалізації проєкту та вартістю технічного забезпечення проєкту. «Чи є у нас потрібні люди, щоб розробити і реалізувати проєкт?» «Чи потребуємо ми нове устаткування для підтримки успішної експлуатації продукту проєкту?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий проєкт (технологія, продукт, програма) мати достатню ефективність аби перенести частину тягаря користувачів на нього і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо інструмент - комп'ютер або новий додаток на комп'ютері, новий продукт чи технологія.

Результатом стадії аналізу здійсненності є бюджет і графік проєкту.

3) Дизайн зовнішнього вигляду. Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проєкт може перейти до стадії проєктування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проєкту.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

4) Розробка та впровадження. На стадії розробки і впровадження, основна діяльність управління вимогами - це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створений продукт дійсно відповідає вимогам. Основним інструментом, що використовується на цій стадії, є створення прототипу (моделі) і подальша реалізація проєкту. Результати перевірки прототипу (моделі) дозволяють внести остаточні корективи до проєкту перед початком його реалізації. Це економить час і витрати замовника проєкту.

5) Завершення. Управління вимогами не закінчується фізичною реалізацією проєкту чи початком випуску продукту. З цього моменту отримані дані збираються і використовуються під час фази дослідження для наступних аналогічних проєктів чи нового покоління продукту. Таким чином, процес починається знову.

## **2. Вимоги категоризуються кількома способами.**

Залежно від проблемної галузі і застосовуваних бізнес-аналітиками методологій виділяють різні типи вимог:

бізнес-вимоги – визначають цілі, завдання і потреби організації. Бізнес-вимоги дають змогу описати причини, з яких розпочинається проєкт, можливості, які цей проєкт має забезпечити, і метрики, якими можна виміряти результат проєкту;

вимоги користувачів – описують потреби зацікавлених осіб (замовників, співробітників організації);

функціональні вимоги – описують можливості, які має реалізувати проєктне рішення, і інформацію, якою воно буде керувати;

вимоги якості обслуговування – визначають зовнішні умови, за яких проєкт має залишатися ефективним;

припущення і обмеження – ідентифікують аспекти предметної області, які не належать до функціональних вимог, але накладають певні обмеження або впливають на проєкт;

вимоги реалізації – описують можливості проєктованої системи, необхідні на етапі її впровадження.

Щодо технічних ІТ проєктів (особливо програмних продуктів), то категоризація вимог може бути подана в такому вигляді:

Вимоги споживача – вираження фактів та припущень, які описують очікування до системи в термінах цілей, середовища, обмежень та міри ефективності й придатності. Споживачі в програмних проєктах – це ті, хто виконують вісім первинних функцій системної інженерії, з особливим наголосом на операторі, як на ключовому споживачі.

Архітектурні вимоги – пояснюють що має бути зроблено з ідентифікацією необхідної системної архітектури.

Структурні вимоги – вимоги пояснюють що має бути зроблено з ідентифікацією необхідної структури системи.

Поведінкові вимоги – пояснюють що має бути зроблено з ідентифікацією необхідної поведінки системи.

Функціональні вимоги – пояснюють що має бути зроблено з ідентифікацією необхідної задачі, дії чи діяльності, які мають виконуватись.

Нефункціональні вимоги – це вимоги, що задають критерій для оцінки операцій системи, замість її поведінки.

Вимоги продуктивності – вказують, до якої межі місія чи функції повинні бути виконані; зазвичай вимірюються в термінах кількості, якості, охоплення, своєчасності чи готовності. При аналізі вимог вимоги продуктивності (як добре воно має бути зроблено) будуть розроблятися для всіх виявлених функцій, базуються на факторах життєвого циклу системи і характеризуються в термінах ступеня визначеності їх оцінок, ступеня критичності успіху системи і їх відношення до інших вимог.

Вимоги дизайну передбачають вимоги «будувати до», «кодувати до», і «купувати до» для продуктів, і «як виконати» для процесів виражених в технічних пакетах даних та інструкціях.

Успадковані вимоги – це вимоги, які враховуються вимогами вищого рівня, чи перетворені з них. Наприклад вимога великої дальності, чи високої швидкості може спричинити вимогу дизайну малої ваги.

Розподілені вимоги – це вимоги, які визначені поділом чи іншим перерозміщенням високорівневих вимог в кілька низькорівневих вимог. Наприклад, стокілограмовий пристрій, що складається з двох підсистем, може спричинити вимоги ваги не більше 70 та 30 кілограм для конкретних систем нижчого рівня.

### **3.3. Аналіз вимог.**

Аналіз вимог полягає в визначенні потреб та умов, які висуваються щодо нового чи зміненого продукту, враховуючи можливі конфліктні вимоги різних замовників, таких як користувачі чи бенефіціари.

Аналіз вимог є критичним для успішної розробки проєкту. Вимоги мають бути задокументованими, вимірними, реальними, пов'язаними з

бізнес-потребами і описаними з рівнем деталізації, достатнім для розуміння розробниками і виконавцями.

Аналіз вимог включає три види діяльності:

- Виявлення вимог – включає комунікації з користувачами для визначення їх вимог. Також це називають збором вимог.

- Аналіз вимог – включає виявлення недоліків вимог (неточностей, неповноти, неоднозначностей чи суперечностей) і їх виправлення.

- Запис вимог – передбачає документування в різних формах, таких як опис звичайною мовою, прецедентами, користувацькими історіями чи специфікаціями процесу.

Аналіз вимог може бути довгим та важким процесом, що вимагає використання тонких психологічних навиків. Нові проекти змінюють середовище і стосунки між людьми, тому важливо розпізнати всі зацікавлені сторони, взяти до уваги всі їхні потреби і переконатись, що вони розуміють наслідки, які приносить проєкт. Аналітики можуть використати кілька методів, щоб отримати від споживача вимоги. Історично це включає проведення інтерв'ю, чи фокус-груп (яку в цьому контексті частіше називають «майстернею вимог») і створення списків вимог. До сучасніших підходів відносять прототипування та прецеденти. За потреби аналітик використає комбінацію цих методів, щоб встановити точні вимоги зацікавлених сторін таким чином, щоб система відповідала бізнес-потреbam.

**Аналіз вимог передбачає такі розділи:**

1) Визначення зацікавлених сторін

Зацікавлені сторони (ЗС) це особи чи організації, які мають дійсний інтерес до проєкту. Вони можуть впливати на проєкт прямо чи опосередковано. Зазвичай, зацікавлені сторони не обмежуються організацією, що найняла аналітиків. До зацікавлених сторін також відносять:

- кожного, хто використовуватиме продукти проєкту;
- будь-кого, хто отримає вигоди від проєкту (функціональні, політичні, фінансові та соціальні бенефіціари);
- кожен, хто бере участь в придбанні чи закупці продукту проєкту (наприклад, при розробці продуктів для масового ринку відділи менеджменту продукту, маркетингу і іноді продаж діють як сурогатні споживачі, які направляють розробку продукту);
- організації, що регулюють аспекти системи (фінансові, безпеки та інші регулятори);
- люди та організації, що протистоять системі (негативні зацікавлені сторони);
- організації, які відповідають за систему і які в майбутньому будуть взаємодіяти з продуктом проєкту, що розробляється;
- ті організації, які горизонтально інтегруються з організацією, для якої розробляють проєкт.

**2) Інтерв'ю з зацікавленими сторонами**

Інтерв'ю з ЗС є рядовим підходом, що використовується в аналізі вимог. Вони використовуються як спосіб отримання висококонцентрованих знань, які часто не можна виявити при спільних сесіях розробки вимог, де увага зацікавленої сторони підпорядкована забезпеченню більш крос-функціонального контексту. Більш того, особистий характер інтерв'ю має більш розслаблюючий характер, при чому хід думок може бути детальніше пояснений.

### **3) Спільні сесії розробки вимог**

Вимоги часто мають крос-функціональні наслідки, які невідомі окремим зацікавленим сторонам і часто пропускаються чи неправильно описуються протягом інтерв'ю з ЗС. Ці крос-функціональні наслідки можуть бути виявлені при проведенні спільних сесій розробки вимог в контрольованому середовищі, яке стимулює кваліфікований посередник, де ЗС беруть участь в дискусії з метою виявлення вимог, аналізують їх деталі і розкривають крос-функціональні наслідки. Тут мають бути присутні спеціально призначені секретар та бізнес-аналітик для документування дискусії. Використання навиків навченого посередника для управління дискусією звільняє бізнес-аналітика від цих обов'язків, що дозволяє йому сфокусуватись на процесі визначення вимог.

### **4) Документування вимог**

Традиційним способом документування вимог є складання списку вимог в стилі контракту. В складній системі такий список вимог може розтягуватись на сотні сторінок.

Відповідною метафорою може бути надзвичайно довгий список покупок. Такі списки не дуже цінуються в сучасному аналізі, так як вони показали малу успішність в досягненні своїх цілей. Тим не менш, їх іноді можна побачити і сьогодні.

Переваги таких списків:

- надає чіткий попунктовий список вимог;
- надає готовий контракт між спонсорами проєкту та розробниками;
- для великого проєкту може надати опис високого рівня.

Недоліки списків:

- вони розтягуються на сотні сторінок, тому їх майже неможливо прочитати цілком і отримати повне розуміння майбутнього проєкту;

- абстрагують всі вимоги, тому є мало справжнього контексту, тобто така абстракція робить неможливим побачити те, як вимоги сполучаються чи працюють разом; заважає правильно розставляти пріоритети між вимогами; збільшує подібність та ймовірність неправильної інтерпретації вимог (чим більше людей їх прочитають, тим більша кількість різних інтерпретацій з'явиться); означає, що надзвичайно важко впевнитись, що ви маєте більшість вимог;

- створюють фальшиве відчуття взаємного розуміння між зацікавленими сторонами та розробниками;

- дають ЗС фальшиве відчуття безпеки, що розробники мусять досягти певних цілей. Тем не менш, через природу таких списків вони упускають критичні вимоги, які виявляються пізніше в процесі впровадження та експлуатації. Розробники можуть використати ці відкриті вимоги, щоб переглянути умови договору на свою користь;

- не допомагають в проєктуванні проєкту.

Як альтернативу до великих, попередньо-описаних списків вимог можна використати історії користувачів (описують чого користувач хоче досягти), щоб описати вимогу в повсякденних термінах.

### **5) Вимірювані цілі**

Найкращі практики беруть складений список вимог як підказку, і постійно запитують «Чому?», поки не стане зрозумілою справжня бізнес-ціль. Зацікавлені сторони і розробники можуть скласти тести, що вимірюють рівень досягнення цілей. Такі цілі змінюються повільніше, ніж довгий список конкретних, але невимірних вимог. Як тільки невеликий набір критичних, вимірних цілей буде встановлений, швидке прототипування та короткі фази розробки можуть продовжити приносити справжні цінності для зацікавленої сторони задовго до середини проєкту.

### **б) Прототипи**

В середині 1980-тих прототипування розглядалось як найкраще рішення до проблеми аналізу вимог. Прототипи – це макети продукту, технології програмного забезпечення. Макети дозволяють користувачам візуалізувати ще не створене. Прототипи допомагають користувачам уявити як буде виглядати кінцевий продукт і спростити для користувачів прийняття рішень. Після введення прототипів спостерігаються значні покращення в комунікаціях між користувачами й розробниками. Раннє бачення продукту проєкту приводить до меншої кількості змін у майбутньому і тому зменшує загальну вартість проєкту.

Тим не менш, протягом останнього десятиліть прототипування, хоча й зарекомендувало себе як корисна техніка, але не розв'язало проблему вимог:

- як тільки менеджери бачать прототип, вони перестають розуміти що проєкт ще не буде завершений протягом деякого часу;

- дизайнери часто думають, що вони змушені використовувати склеєний докупи прототип в реальному проєкті, бо вони бояться «втратити час» починаючи заново;

- прототипи загалом допомагають в конструкторських рішеннях та проєктуванні зовнішнього вигляду продукту проєкту та вони не можуть сказати, які вимоги були спочатку;

- дизайнери і кінцеві користувачі можуть занадто сфокусуватись на конструюванні зовнішнього вигляду продукту проєкту і занадто мало на створенні продукту, який повинен покращити бізнес-процес.

Прототипи добре працюють для програмних продуктів, технічної продукції, але не дуже корисні для процесів, які можуть включати складні розрахунки чи запити до баз даних.

## 7) Прецеденти

Прецеденти – це технологія документування потенційних вимог до нового чи зміненого програмного продукту чи товару. Кожен прецедент надає один чи більше сценаріїв, які виражають те, як продукт взаємодіє з користувачем чи іншою системою, щоб досягти конкретної бізнес-цілі. Прецеденти зазвичай уникають технічного жаргону, віддаючи перевагу мові кінцевого користувача чи експерта в предметній області. Інженери вимог та ЗС часто виступають співавторами прецедентів.

Прецеденти є оманливо простими інструментами для опису поведінки продукту. Прецедент містить текстовий опис всіх способів, якими користувачі можуть працювати з продуктом. Прецеденти не описують жодних внутрішніх процесів у системі і не пояснюють, як система має реалізовуватись. Вони просто показують кроки, які користувач має здійснити, щоб виконати свою задачу. Так можна описати всі способи взаємодії з продуктом.

### 4. До проблем аналізу вимог можна віднести:

- 1) Проблеми з зацікавленою стороною, яка перешкоджає збору вимог:
  - замовники не розуміють чого їм треба чи не мають чіткого уявлення про свої вимоги;
  - замовники не вкладуть нічого в набір письмових вимог;
  - замовники наполягають на нових вимогах після фіксації ціни та графіку розробки;
  - спілкування з замовниками відбувається повільно;
  - замовники часто не беруть участі у оглядах чи не мають змоги брати участь;
  - замовники неграмотні технічно чи економічно;
  - замовники не розуміють процес розробки і реалізації проєкту;
  - замовники не знають про сучасні технології.

Це може привести до ситуації, в якій вимоги замовника продовжують змінюватись навіть тоді, коли почалась реалізація проєкту.

### 2) Проблеми з інженерами/розробниками

Можливі проблеми які можуть спричинити розробники та інженери протягом аналізу вимог:

- технічний персонал та кінцеві користувачі можуть говорити різними мовами. Вони можуть помилково вірити в те, що вони перебувають в ідеальній згоді, поки не буде наданий закінчений продукт;
- інженери та розробники можуть спробувати зробити вимоги, які підходять до існуючої системи чи проєкту, замість того, щоб розробляти систему чи проєкт спеціально під потреби клієнта;
- аналіз часто може проводитись інженерами чи програмістами, а не персоналом з навиками комунікації та знанням предметної області для правильного розуміння потреб клієнта.

### 3.4. Функціонально - фізичний аналіз технічних об'єктів (ФФА).

Є частиною найбільш загального функціонально-вартісного аналізу. Метою ФФА є поглиблене вивчення конструкцій та структури ТО, який потрібно удосконалити [7].

При такому вивченні насамперед необхідно зрозуміти та уточнити таке:

- які функції виконує кожен елемент і як елементи функціонально пов'язані між собою;
- які фізичні операції (перетворення) виконує кожен елемент і як елементи функціонально пов'язані між собою;
- на основі яких фізико-технічних ефектів працює кожен елемент ТО та як вони взаємопов'язані між собою.

При виконанні цих питань з'являється чітке та цілісне уявлення про пристрій ТО з функціональної та фізичної точок зору.

Аналіз ТО виконують у наступній послідовності:

Побудова конструктивної функціональної структури (ФС).

Будь-який ТО можна розділити кілька елементів кожен із яких має цілком певну функцію із забезпечення роботи ТО чи його елементів.

Граничний поділ ТО можливий до неподільних елементів із мінімальним числом функцій. Серед усіх елементів особлива увага приділяється головним елементам (позначається  $E_0$ ). Елементи ТО позначаються  $E_0, E_1, \dots, E_N$ .

Як приклад розглянемо конструкцію побутової електроплитки, представлену на рис. 3.1.

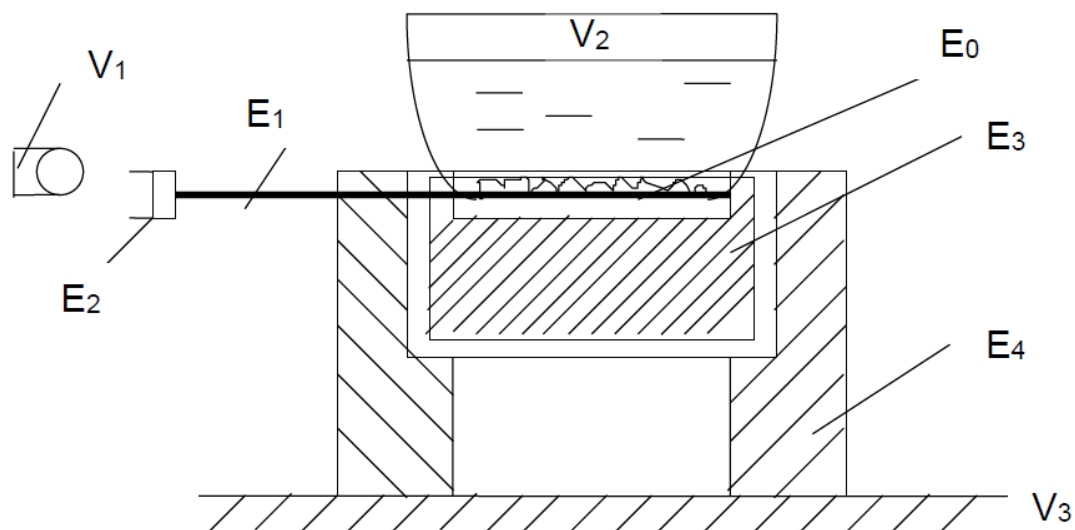


Рис. 3.1 Конструкція побутової електроплитки.

Аналіз функцій електроплитки представляється як таблиці 3.1.

Конструктивна ФС є орієнтований граф, вершинами якого є найменування елементів ТО та об'єктів ОС, а ребрами - функції елементів. Конструктивна ФС побутової електроплитки представлена на рис. 3.2.

Аналогічно аналізу функцій ТО можна провести аналіз технологічних процесів. При цьому для технологічних процесів ФС є граф вершинами якого є об'єкти Е, що обробляються, а ребрами елементарні операції Ф із зазначенням режимів обробки.

Таблиця 3.1 – Аналіз функцій електроплитки

Позначення	Найменування	Позначення	Опис
Е0	Спіраль	Ф0	Нагріває ємність з рідиною (V2) до кипіння.
Е1	Дріт	Ф1	Проводить струм від електромережі (V1) до спіралі (Е0)
Е2	Гніздо	Ф2	З'єднує та роз'єднує провід (Е1) з електромережею (V1)
Е3	Вогнетривкий елемент	Ф3' Ф3" Ф3'''	Зменшує теплову дію спіралі (Е0) на стіл ( V3 ) . Підтримує спіраль (Е0) у заданому положенні. Ізолює спіраль (Е0).
Е4	Корпус	Ф 4	Передає вплив маси ємності з рідиною (V2) на стіл (V3) .

### Побудови функціональної структурної структури.

Взаємопов'язаний набір фізичних операцій (ФО), що реалізують один певний потік перетворень речовини, енергії або сигналів, або кілька взаємопов'язаних потоків називатимемо потоковою функціональною структурою.

Потокова ФС є граф, вершинами якого є найменування елементів ТО чи найменування операцій Коллера Е, а ребрами - вхідні АТ і вихідні СТ потоки (чинники).

Розрізняють два різновиди потокової ФС: конкретизована потокова ФС, у якій у вершинах графа вказані найменування елементів; абстрагована потокова ФС, яка має у вершинах графа зазначені найменування операцій Коллера. Конкретизована потокова ФС побутової електроплитки представлена на рис. 3.3.

Список операцій Коллера наведено нижче в табл. 3.2. Потокова абстрагована ФС побутової електроплитки представлена на рис. 3.4. У табл. 3.2.,  $G_A$ ,  $G_B$  – два якісно відмінні види енергії, речовини або сигналу, що мають різні властивості, що вимірюються різними величинами.

Таблиця 3.2 – Список операцій Коллера

№ п / п	Найменування прямої операції	Узагальнена структурна формула	Назва зворотної операції Колера	Узагальнена структурна формула
1	Випромінювання	$G_A \rightleftarrows$	Поглинання	$G_A \leftleftarrows$
2	Провідність	$G_A \rightarrow G_B$	Ізолювання	$G_A \rightarrow$
3	Збір	$G_A^* \rightarrow \bar{G}_A$	Розсіювання	$G_A \rightarrow \bar{G}$
4	Проведення	$G_A \rightarrow G_A$	Непроведення	$G_A \bar{G}_A$
5	Перетворення	$G_A \rightarrow G_B$	Зворотне перетворення	$G_B \rightarrow G_A$
6	Збільшення	$G_{A1} < G_{A2}$	Зменшення	$G_{A1} > G_{A2}$
7	Зміна напрямку	$G_A \quad G_A \uparrow$	Зміна напрямку	$G_A \quad G_A \downarrow$
8	Вирівнювання	$G_A \rightleftarrows \rightarrow$	Коливання	$\rightarrow \rightleftarrows G_A$
9	Зв'язок	$G_A \overset{\circ}{\text{---}} G_A$	Переривання	$G_A \overset{\circ}{\text{---}} G_A$
10	З'єднання	$G_A + G_B > G_{AB}$	Роз'єднання	$G_{AB} < G_A + G_B$
11	Об'єднання	$G_{A1} + G_{A2} > G_{A1+A2}$	Поділ	$G_{A1+A2} < G_{A1} + G_{A2}$
12	Нагромадження	$G_A \rightarrow \circ$	Видача	$\rightarrow G_A \circ$
13	Відображення	$G_A \rightarrow G_B$	Зворотне відображення	$G_B \rightarrow G_A$
14	Фіксування	$G_A \rightarrow \leftarrow G_B$	Розфіксування	$G_A \leftarrow \rightarrow G_A \circ$

$G_{A1}, G_{A2}$  – два кількісно відмінні стани енергії, речовини або сигналу, що вимірюються однією і тією ж фізичною величиною.  $G_{AB}$  – енергія, речовина або сигнал, що є композицією з двох різнорідних компонентів  $G_A, G_B$  мають якісну відмінність,  $G_{A1+A2}$  – енергія, речовина або сигнал, що являють собою композицію з двох різнорідних компонентів  $G_{A1}, G_{A2}$  різняться кількісно.

### Опис фізичного принципу дії (ФПД).

Елементарна фізична операція може бути реалізована за допомогою одного фізико-технічного ефекту (ФТЕ).

Побудова ФПД виробляє потокової абстрагованої ФС.

У вершинах графа вказують фізичні об'єкти і наскільки можна ФТЕ. Ребра – види енергії, інформації чи речовини. Граф ФПД побутової електроплитки представлений на рис. 3.5

На цьому етапі формулюють головний критерій розвитку та вдосконалення ТО та виділяють другорядні критерії розвитку ТО, на які накладають обмеження. Складають список недоліків ТО та намічають шляхи їх подолання. Якщо метою аналізу є розробка комп'ютерної системи, складається її функціонально-алгоритмічна структура.

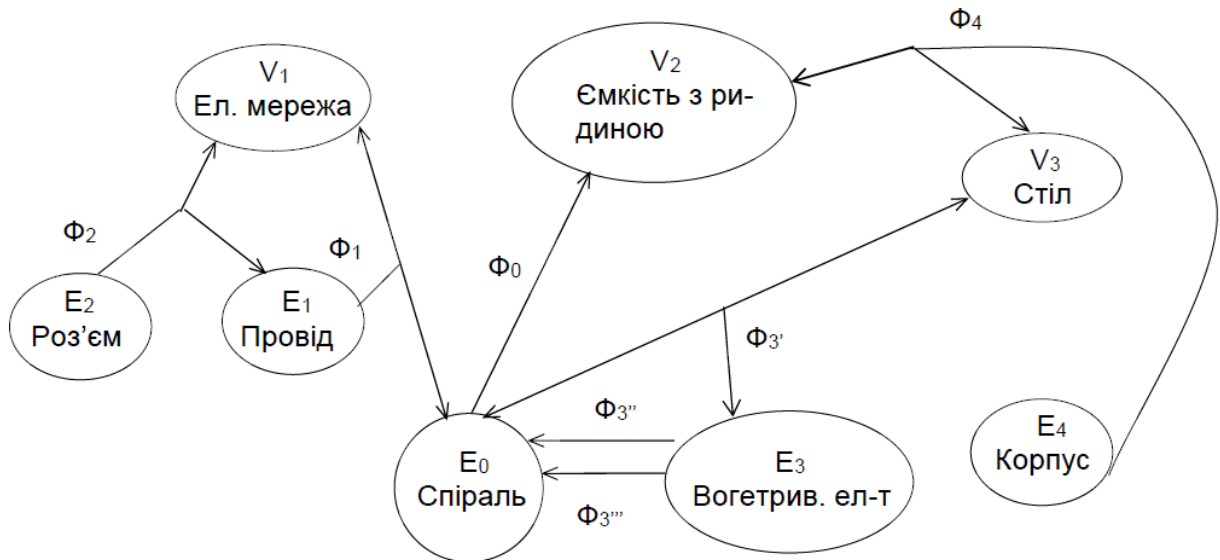


Рис. 3.2 – Конструктивна багатofункціональна структура.

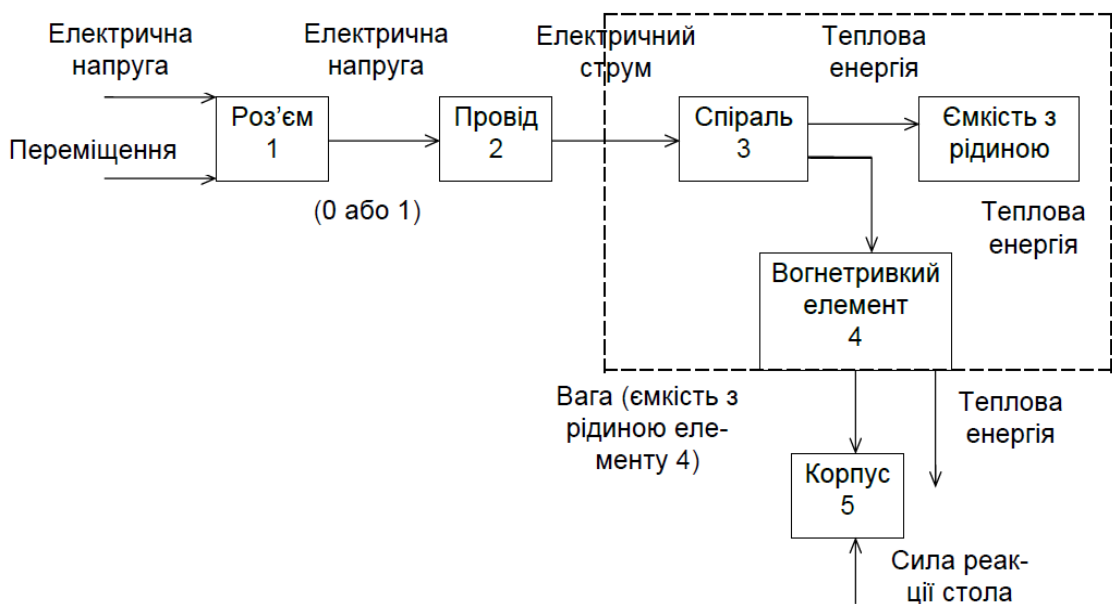


Рис. 3.3 – Конкретизована потокова багатofункціональна структура.

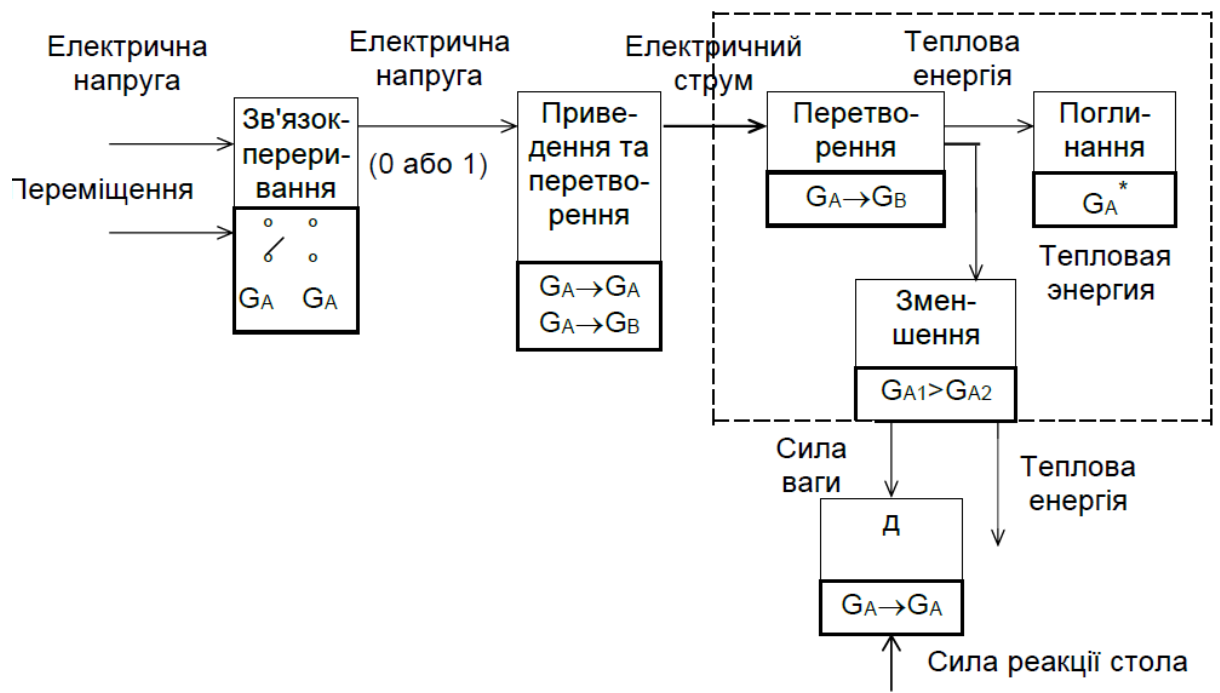


Рис. 3.4 – Поточкова абстрагована функціональна структура побутової електроплитки.

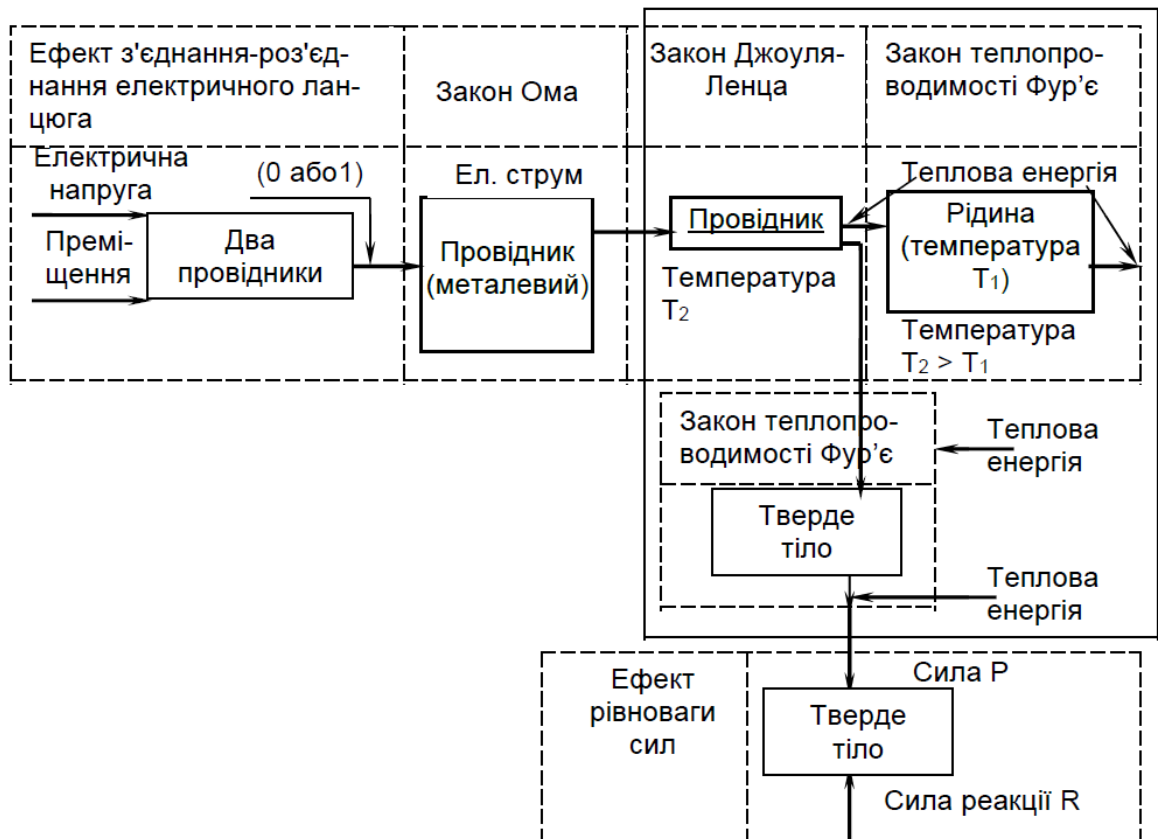


Рис. 3.5 – Граф фізичного принципу впливу

### 3.4. Використання інструментів для документування та моделювання систем

Моделі UML, що пояснюють функціональність системи [8].

Діаграма варіантів використання

Діаграма варіантів використання UML, Use Case Diagram – одне з найпростіших представлень системи. Її базові «будівельні елементи» – актори і варіанти використання.

Діаграма задумана так, щоб дати найбільш загальне уявлення про функціональність системи (її компоненти), не вдаючись до деталей взаємозв'язків функцій. Тому основний вид відношення, використовуваний в діаграмі, – асоціація між актором і варіантом використання.



Рис. 3.6 – Use Case Diagram

Інші види відносин – відношення включення (include), розширення (extend) і узагальнення/генералізувати.

Включення служить для позначення підлеглих варіантів використання (коли один або більш за варіанти використання містять виклики однієї і тієї ж функціональності).



Рис. 3.7 – Use Case Diagram

Розширення в точності відповідає точці розширення, яку використовуємо при описі варіанту використання.

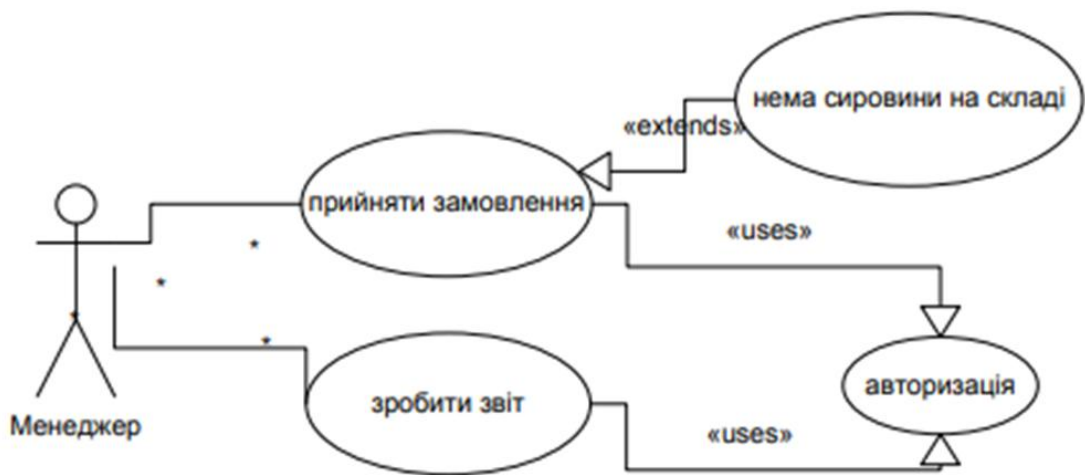


Рис. 3.8 – Use Case Diagram

Відношення узагальнення може застосовуватися як до акторів, так і до варіантів використання, з метою вказівки спеціалізації одні щодо інших.

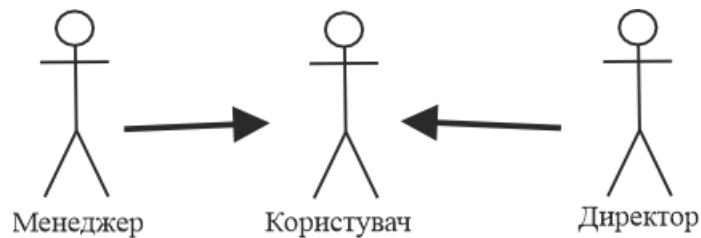


Рис. 3.9 – Use Case Diagram

Діаграма дій Якщо діаграма варіантів використання дає «вигляд зверху» на функціональність системи, діаграма дій UML, навпаки, дозволяє детально ілюструвати окремий варіант використання і його сценарії.

Основні компоненти опису системи:

- Функції (дії)
- Символи «старт» і «стоп»
- Потоки управління
- Розгалужувачі
- Лінійки синхронізації.

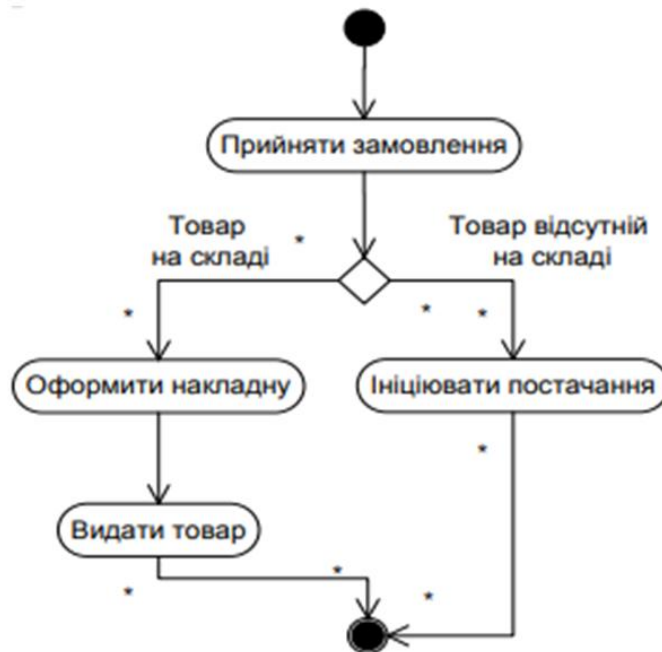


Рис. 3.10 – Діаграма дій

Діаграма дій дозволяє проілюструвати варіант використання з необхідним ступенем деталізації.

Лінійний варіант використання приводить до діаграми дій з лінійним потоком управління між діями. Дії варіанту використання з альтернативними сценаріями реалізується через розгалужувачі. Лінійки синхронізації дозволяють описувати такі складні конструкції, як синхронізацію початку (закінчення) паралельних в часі процесів. Окрім стандартного формату опису, UML пропонує варіант з «плаваючими доріжками». Цей формат зручний для опису випадку, коли у варіанті використання беруть участь декілька акторів.

**Діаграма станів.** Діаграма станів в аналізі вимог використовується, коли потрібно досліджувати поведінку системи, як кінцевого автомата. Це уявлення прийшло в UML з теорії систем.

У загальному випадку діаграма станів описує, як система поводить себе в більш, ніж одному варіанті використання.

Синтаксис діаграм станів багато в чому співпадає з синтаксисом діаграм дій.

Основні компоненти опису системи:

- Прості стани;
- Складені стани ;
- Символи «старт» і «стоп»;
- Переходи;
- Лінійки синхронізації.

У мові UML під станом розуміється абстрактний метаклас, використовуваний для моделювання окремої ситуації, протягом якої має

місце виконання деякої умови. Стан може бути заданий у вигляді набору конкретних значень атрибутів класу або об'єкту, при цьому зміна їх окремих значень відобразить зміну стану модельованого класу або об'єкту.

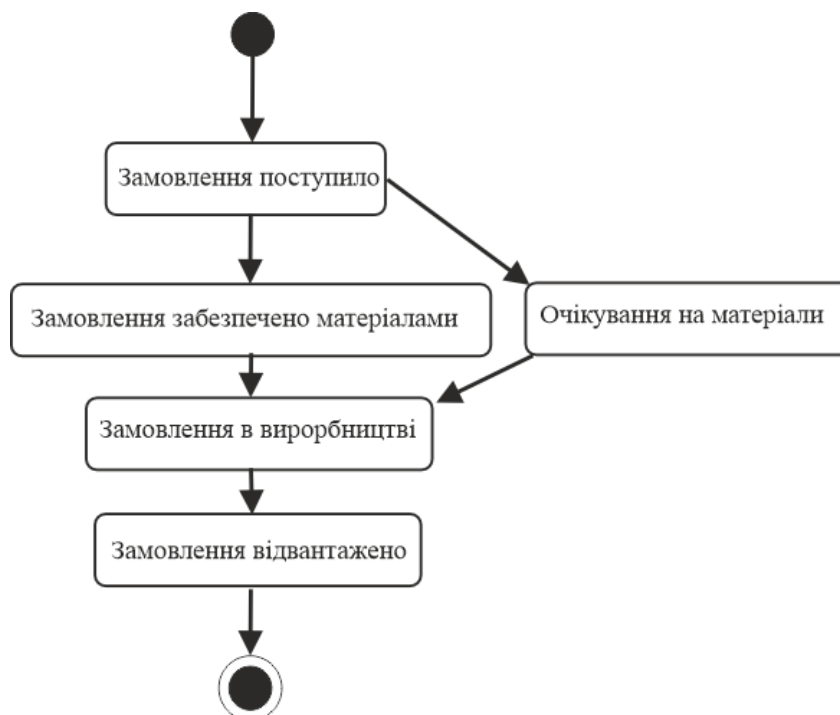


Рис. 3.11 – Діаграма станів

Перехід системи із стану в стан здійснюється при настанні подій. При цьому кажуть, що перехід спрацьовує. Перехід може бути безальтернативним, або містити альтернативи. У другому випадку перехід обумовлений настанням межових умов. Нарешті, подія може супроводжуватися виразом дії, яка відбувається у випадку, якщо спрацьовує перехід. Повний синтаксис опису переходу (написи на стрілці) наступний:

**Подія [сторожова умова] / вираз дії.** Іноді буває корисним об'єднати частину станів в один мета-стан. Графічно це виглядає, як символ стану (прямокутник з кутами, що округляють), що містить усередині себе декілька символів станів. При цьому можливі переходи між підлеглими станами, переходи між підлеглим і зовнішнім станами і переходи між складеним і зовнішнім станом.

#### **Діаграми UML, що пояснюють внутрішній устрій системи**

Деякі автори рекомендують використовувати при описі вимог діаграми UML, що описують створювану систему через її компоненти (класи, об'єкти), відносини і взаємодії між ними.

Даний підхід має свої обмеження Діаграма класів. Для створення діаграми класів необхідно:

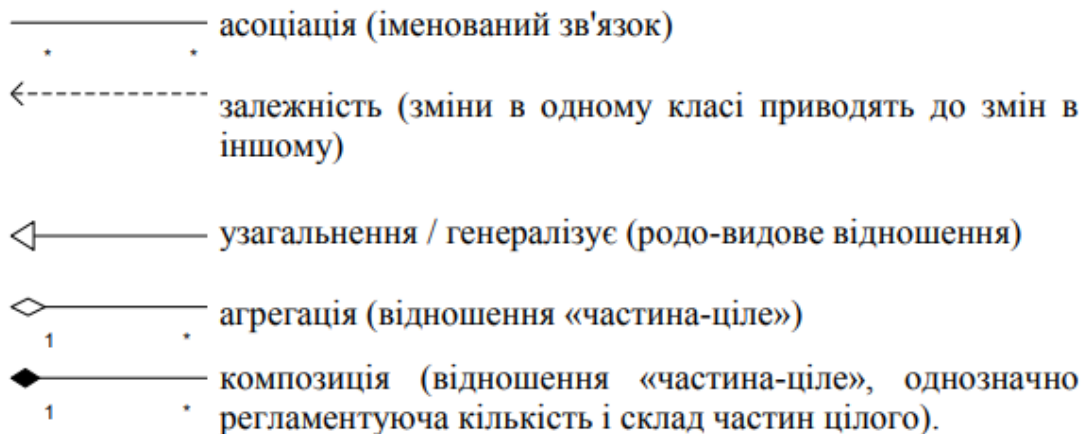
1. Здійснити пошук класів (ключових компонент проблемної області).
2. Для кожного знайденого класу визначити його ім'я, основні атрибути, операції і (або) відповідності.

3. Дослідити взаємозв'язки знайдених класів. Класи на діаграмі представляються у вигляді прямокутників, відносини – у вигляді ліній, що зв'язують прямокутники. Лінії різного типу графічно відрізняються різним штрихуванням і стрілками.

Прийнято виділяти 3 рівні абстракції класів: концептуальний рівень, рівень специфікації, рівень реалізації. Аналіз вимог розумно супроводжувати діаграмами, що відображають концептуальний рівень.

На даному рівні при описі класів доцільно вказати їх найменування і відповідальності. атрибути і операції можна не вказувати, або ввести тільки найосновніші, відклавши їх дослідження на пізніші стадії деталізації.

Відносини, що підлягають аналізу на концептуальному рівні, – це:



Діаграма класів показує статичну структуру проблемної області. Для аналізу взаємодії об'єктів – екземплярів класу в ході реалізації варіанту використання в UML передбачено дві діаграми взаємодії: діаграма кооперації і діаграма послідовності.

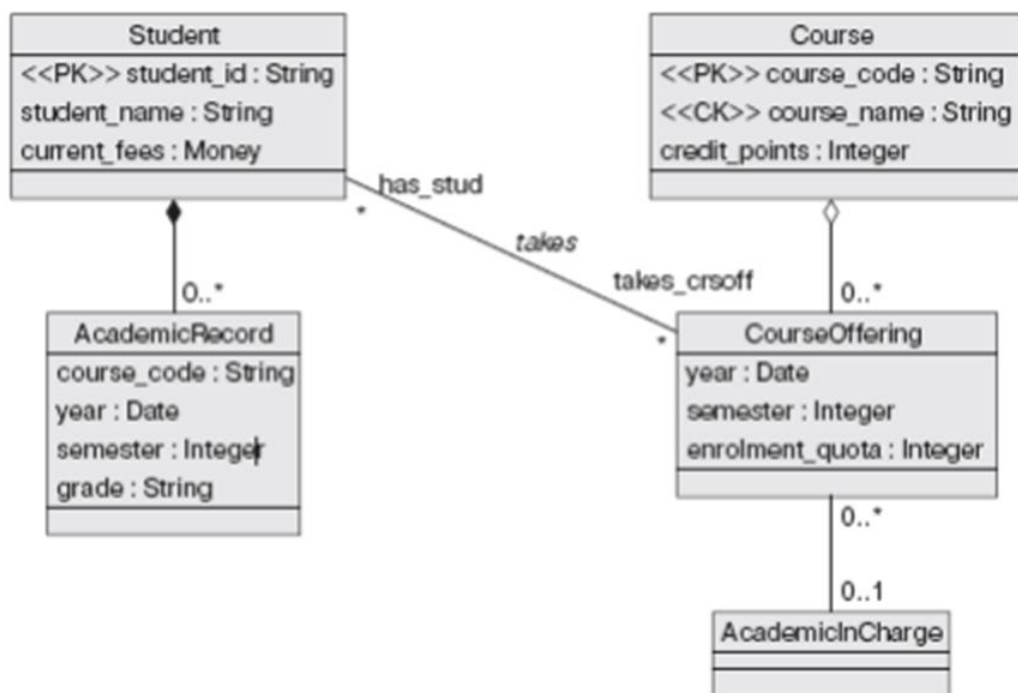


Рис. 3.12 – Специфікація агрегацій (запис на університетські курси)

Якщо діаграма класів у багатьох випадках і може розглядатися, як артефакт, що пояснює структуру проблемної області, то діаграми взаємодії навряд чи варто розглядати як виразний мовний засіб, що ілюструє вимоги до системи в діалозі «Замовник-Виконавець». Проте, у відповідності з Принципом 3 свобода вибирання мовних засобів, аналітик, що вирішив використовувати дані діаграми, може ознайомитися з ними в спеціальній літературі.

**BPMN** (Business Process Modeling Notation) - це нотація для моделювання бізнес-процесів, що дозволяє чітко представити діяльність в рамках організаційних процесів.

**CASE-системи** (Computer-Aided Software Engineering) використовуються для автоматизації процесів моделювання, аналізу, проектування і тестування програмних систем.

### **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Порівняйте функціональний, структурний і поведінковий підходи до опису системи: що описує кожен і які моделі/діаграми застосовуються?
2. Опишіть життєвий цикл управління вимогами: виявлення, формалізація, верифікація, трасування, управління змінами. Які артефакти формуються на кожному етапі?
3. Які методи аналізу вимог ви застосуєте для перевірки повноти, непротиворічності та досяжності вимог? Наведіть конкретні приклади технік.
4. Що таке функціонально-фізичний аналіз (ФФА) технічного об'єкта і які питання він допомагає вирішити при проектуванні?
5. Назвіть і опишіть інструменти (CASE/MBSE/СУБД, UML/BPMN) для документування й моделювання систем; які їхні сильні та слабкі сторони у процесі аналізу?

#### **Тема 4. Побудова ієрархічної структури систем**

##### **4.1 Огляд ієрархічних структур в системах: дерева рішень, організаційні діаграми, діаграми класифікацій**

Ієрархічна структура в системному аналізі відображає спосіб організації елементів системи в декілька рівнів, де кожен елемент на вищому рівні управляє або впливає на елементи на нижчому рівні. Це один із основних інструментів для організації складних систем і процесів[9].

**Складна ієрархічна система** - це множина структурно взаємопов'язаних ієрархічною підпорядкованістю і функціонально взаємозалежних систем, які є об'єднаними для досягнення мети в конкретних умовах за існуючих ресурсів.

Прикладами складних ієрархічних систем можуть служити сучасні виробничі об'єкти, космічні системи зв'язку, навігації, дистанційного зондування, сучасні системи управління регіонами, корпораціями, багатопрофільними фірмами тощо. Аналіз таких систем не зводиться лише до встановлення типів елементів чи типів відношень. Суттєве значення у

даному випадку має ієрархічність структури не лише топології цих систем, а й систем управління.

**Ієрархія** (грец. *ієραρχία*, від *ієρός* – святий, священний і *αρχή* – влада) – розташування частин чи елементів цілого у порядку від нижчого до вищого

#### **Типи ієрархічних систем.**

**Домінантна ієрархія** - це структура організації системи, в якій певні елементи або підсистеми мають вищий рівень впливу, контролю чи пріоритету порівняно з іншими. Вона визначає підпорядкування та розподіл ролей у складних системах, що дозволяє ефективно управляти їхньою роботою.

Домінантна ієрархія може застосовуватися в різних контекстах, таких як управління організацією, біологічні системи, технічні системи та соціальні структури.

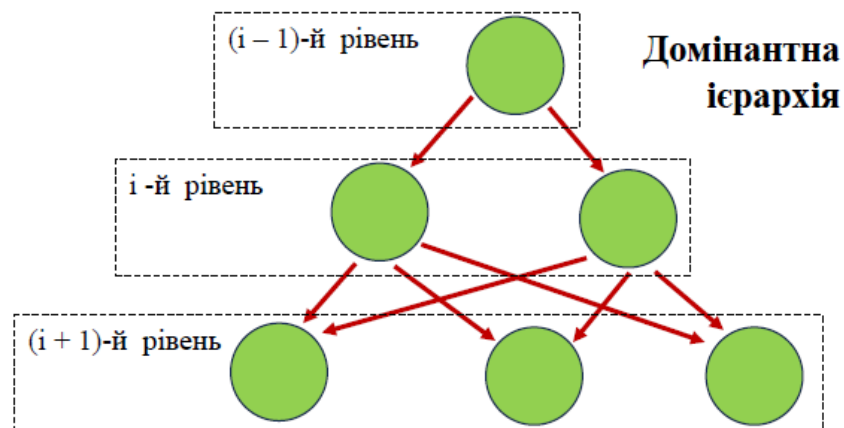


Рис. 4.1 Домінантна ієрархія.

Домінантна ієрархія визначає взаємозв'язки між елементами системи, забезпечуючи ефективне управління, розподіл ресурсів та прийняття рішень. Вона є ключовим елементом у багатьох сферах – від природи до технологій та бізнесу.

**Холархія** (від грец. "holos" - ціле і "archia" - влада) - це ієрархічна структура, яка складається з **холонів** - автономних, але взаємопов'язаних елементів, що можуть функціонувати як самостійні одиниці, але одночасно є частинами більшого цілого.

На відміну від традиційної жорсткої ієрархії, холархія має більш гнучку структуру, де **кожен рівень може бути як самостійною одиницею, так і частиною більшої системи**. Це дозволяє організаціям, організаціям або системам ефективно функціонувати в складних і динамічних умовах.

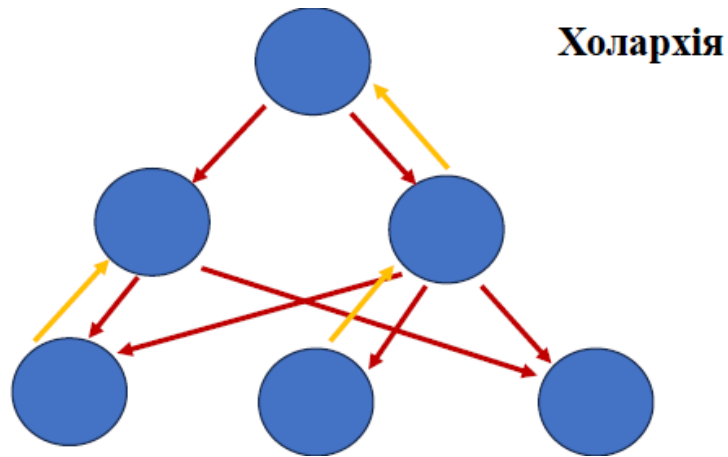


Рис. 4.2 Холархія.

Приклади холархії:

- корпоративна структура: команди → відділи → філії → корпорація.
- децентралізовані компанії (наприклад, холдинги): Фірма має кілька автономних підрозділів, які приймають власні рішення, але підпорядковуються загальній стратегії.
- комп'ютерні мережі: локальні мережі → корпоративні мережі → інтернет.

Кожна частина може працювати автономно, але разом вони утворюють єдину глобальну систему.

**Домінантна неповна ієрархія** - це структура, в якій існують доміантні (головні) елементи, але не всі підпорядковані елементи мають чітко визначене місце або рівень у загальній ієрархії. Це означає, що деякі елементи можуть бути автономними, мати декілька підпорядкувань або взаємодіяти без чіткої лінійної структури. Цей тип ієрархії характерний для складних, децентралізованих або адаптивних систем, де структура може змінюватися залежно від умов або зовнішніх впливів.

**Домінантна  
неповна ієрархія**

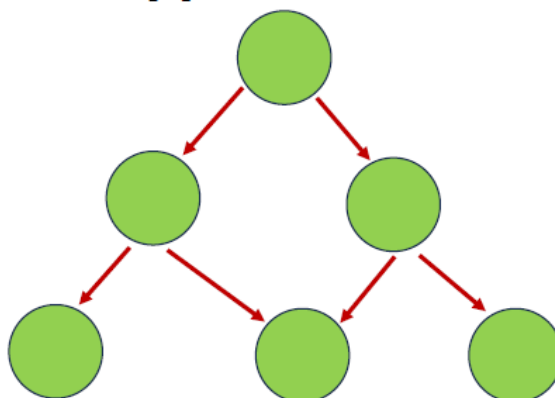


Рис. 4.3. Домінантна неповна ієрархія.

Приклади доміантної неповної ієрархії:

- комп'ютерні мережі (P2P-системи, блокчейн): хоча деякі вузли можуть мати більшу обчислювальну потужність або контроль (наприклад,

майнери в блокчейні), загальна структура залишається децентралізованою та змінюється динамічно.

– хмарні обчислення: є головні сервери, але частина завдань розподіляється між автономними вузлами, які можуть змінювати свою роль залежно від навантаження.

**Модулярна ієрархія** - це структура, в якій система розділена на **відносно незалежні модулі** (блоки, підсистеми, компоненти), що взаємодіють між собою згідно з певними правилами. Кожен модуль може виконувати окрему функцію або групу функцій, а взаємодія між ними забезпечує цілісну роботу всієї системи.

Головною особливістю модулярної ієрархії є **гнучкість, масштабованість і можливість модифікації окремих компонентів без порушення всієї системи.**

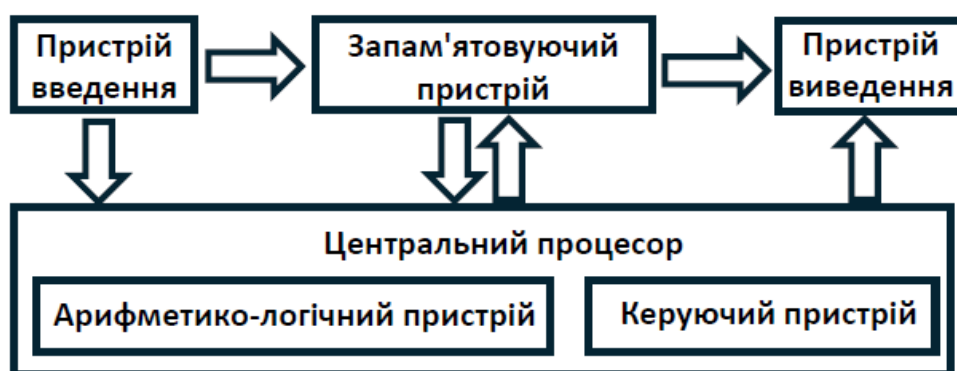


Рис. 4.4. Модулярна ієрархія [9]

Приклади модулярної ієрархії:

– комп'ютерні системи: процесор, оперативна пам'ять, відеокарта, жорсткий диск - кожен із цих компонентів є модулем, який може бути замінений або оновлений без зміни всієї системи.

– корпоративна структура: великі компанії будуються за принципом модульності: фінансовий відділ, HR, маркетинг, виробництво – кожен з підрозділів функціонує відносно автономно, але підпорядковується загальній стратегії.

– інтернет: побудований за принципом модульності: клієнт-серверна архітектура, автономні домени, маршрутизатори та вузли, що можуть змінюватися без порушення всієї мережі.

– мікросервісна архітектура в програмуванні: великі ІТ-системи поділяються на окремі мікросервіси (наприклад, авторизація, обробка платежів, керування замовленнями), які можуть оновлюватися або замінюватися незалежно від інших

Як приклади таких систем можна навести гнучкі комп'ютеризовані виробництва, здатні у процесі функціонування змінювати номенклатуру та обсяг продукції, диспетчерські служби великих аеропортів, морських портів, здатні одночасно обслуговувати від кількох десятків до кількох сотень повітряних або морських суден за мінливих погодних умов, тощо.

Ілюстрацією здатності цілеспрямованих систем до адаптації до умов зовнішнього середовища є дії персоналу електростанцій, великих металургійних, гірничодобувних, хімічних та інших виробництв у критичних і аварійних режимах.

В управлінні проектами ієрархічні структури використовуються для побудови організаційних діаграм, що відображають управлінські зв'язки між учасниками проекту. Це дозволяє чітко визначити, хто відповідає за виконання конкретних завдань і прийняття рішень.

У бізнес-аналізі ієрархії використовуються для розбиття процесів на етапи і підетапи. Це дозволяє зрозуміти, як окремі процеси взаємодіють між собою, а також оптимізувати ресурси та час на кожному етапі.

В розробці програмних продуктів ієрархічні структури використовуються для організації коду в модулі та класи, що полегшує роботу з проектом, його тестування та підтримку.

У сфері інженерії ієрархічні структури використовуються для представлення технічних систем, наприклад, в управлінні мережами, де кожен вузол в мережі має свої функції і зв'язки з іншими вузлами.

#### 4.2 Застосування концепту ієрархії в різних напрямках системних досліджень

Характерною особливістю цілеспрямованих систем є те, що вони мають інтелект - природний, штучний чи створений у результаті поєднання обох його видів. Більшість відомих цілеспрямованих систем належить до класу організаційно-технічних або експертних, Головними елементами є оператори, а також різного роду технічні засоби підтримки рішень, що мають інтелектуальну складову.



Рис. 4.5. Модулярна ієрархія [9]

Цільовий аналіз—призначений для виявлення часткових цілей поведінки складної системи для досягнення головної мети:

Структурно-функціональний аналіз—дає змогу визначення, зокрема, множини функцій для досягнення мети в певних умовах функціонування

Техніко-економічний аналіз—дає змогу визначати необхідні ресурси та витрати для поставленої мети

Метод аналізу ієрархій - МАІ є ієрархічною процедурою для ієрархічного подання елементів, що визначають суть проблеми.

Метод полягає в декомпозиції проблеми та мети її вирішення на більш прості складові частини (частинні критерії).

Метод формалізує якісну інформацію від експертів за допомогою побудови та опрацювання матриць попарних порівнянь послідовності суджень.

Формування ієрархічної структури системи відповідно до поставленої мети та існуючих обмежень на ресурси;

Визначення функціонально-повного набору елементів для кожного рівня ієрархії та їхніх характеристик

- Інформаційний аналіз – дає можливість визначати обсяг, повноту та інші показники інформації про систему та середовище для керування системою у штатних та поза штатних ситуаціях;

- Організаційно-процедурний аналіз – застосовують для виявлення способів організації процесів керування та вибору процедур для досягнення мети:

- Техніко-економічний аналіз дає змогу визначати необхідні ресурси для поставленої мети, передбачати ефект від впровадження розробленої системи, визначати витрати усіх видів ресурсів не лише на технічні засоби, а й на реалізацію процедур керування;

**Управлінське рішення** – це вибір, який містить відповідь на певні питання, що виникають у процесі реалізації кожної функції управління (планування, організування, мотивування, контролювання). **Управляти** – значить вирішувати, яким чином досягти поставленої мети.

### 4.3 Роль ієрархії в системному аналізі та управлінні

Найсуттєвішу розбіжність ієрархій визначає розбіжність понять *рівня* в ієрархії. Необхідність введення кількох понять рівня зумовлено складністю і різноманіттям цілей, задач, функцій, властивостей та можливостей реальних багаторівневих ієрархічних систем, а також різноманіттям властивостей, особливостей і наслідків штатних, позаштатних, критичних та надзвичайних ситуацій їхнього функціонування.

Так, у роботі [10] введено три поняття рівня.

**Ешелон** - термін, що визначає рівень організаційної ієрархії. Ієрархічна структура об'єкта який відповідає поняттю «ешелон», передбачає, що

реальний об'єкт можна зобразити у вигляді багаторівневої організаційної ієрархічної системи, яка має такі властивості:

1) Складається із множини точно виділених і розподілених за рівнями підсистем;

2) Має повноваження, чітко розподілені між рівнями і підсистемами одного рівня, виходячи з формування, вибору і прийняття рішень у визначеній сфері відповідальності;

3) Забезпечує прямий і зворотний зв'язок з управління між підсистемами різних рівнів, а між підсистемами одного рівня - прямий і зворотний зв'язок із взаємодії. Такі системи прийнято називати багаторівневими і багатоцільовими.

**Страта** - термін, який характеризує рівень опису або абстрагування. Ієрархічна структура об'єкта що відповідає поняттю «страта», припускає, що властивості реального складного об'єкта описано у вигляді деякої сукупності, у якій окремі описи наведено з різних позицій та упорядковано з урахуванням рівня їхньої значимості. Такі ієрархічні системи прийнято називати стратифікованими.

**Шар** - термін, що визначає рівень складності прийняття рішення. Ієрархічна структура об'єкта, що відповідає поняттю «шар», передбачає, що загальну процедуру ухвалення рішення реалізують у вигляді певної послідовності часткових процедур, кожна з яких забезпечує можливість одержання розв'язку з певним ступенем обґрунтованості та вірогідності за різних рівнів неповноти, невизначеності, нечіткості і суперечливості вхідної інформації. Таку ієрархічну структуру прийнято називати багатошаровою, багаторівневою або ієрархічною системою прийняття рішень.

Найважливіша особливість розглянутих об'єктів полягає в тому, що багато процедур формування та аналізу ієрархічної структури є (1.5) принципово неформалізованими, а їхня реалізація носить суб'єктивний характер. Ця особливість стосується насамперед загальної структури ієрархії.

Такі процедури, як вибір кількості рівнів (страт, ешелонів, шарів) в ієрархії, виділення елементів на кожному рівні, вибір опису елементів кожного рівня, вибір апарату опису взаємозв'язків в ієрархічній структурі, цілком залежать від ОПР. У практиці проектування таких реальних складних об'єктів, як космічні системи різного призначення, загальну організаційну ієрархію об'єкта (розподіл об'єкта в цілому на підсистеми 1-го рівня або виділення двох верхніх ешелонів, у відповідності до наведеної раніше термінології - об'єкта і підсистеми) визначає генеральний конструктор об'єкта.

Розподіл підсистем 1-го рівня на підсистеми 2-го рівня виконує головний конструктор відповідної підсистеми 1-го рівня тощо. Такий підхід дає змогу здійснювати системну інтеграцію и організаційних задач проектування планування робіт, формування колективів розробників, координування робіт різних колективів тощо.

Важливими системними поняттями є також поняття цілі системи і характеристики системи щодо цілі.

**Ціль** - це кількісна або якісна міра первинних чи вторинних властивостей системи, яку дослідник за певних обставин вважає найкращою. Отже, складну систему можна розглядати з позиції різних цілей. У цьому розумінні система задовольняє багато цілей. Ця особливість, яку ще називають характеристикою системи щодо цілі, може бути виміряна близькістю справжніх та бажаних проявів тих властивостей системи, які передбачені ціллю. Зазвичай її визначають у термінах відповідної функції, яку називають характеристичною функцією. Зазначимо, що можливі й інші визначення відстаней, а отже, й інші формули для розрахунку. Припустимо, що для деякої множини систем визначено тип цілі і відповідна характеристична функція. З кожною системою такої множини пов'язано значення характеристичної функції, яке показує ступінь відповідності системи заданій цілі. Це дає змогу визначити поняття ціленаправленої системи, для якої характеристика щодо заданої цілі більша від деякого заданого порогового значення.

Отже, ціль системи можна визначити різними способами, і це дає змогу вважати, що ціль знаходиться «в руках користувача». Поняття цілі і характеристики є базовими для визначення поняття ціленаправлених систем.

Система з позитивним ступенем ціленаправленості відносно іншої системи повинна мати деякі властивості, відмінні від властивостей останньої, тобто властивості, які пов'язані з ціллю і визначають можливість поліпшення характеристики цієї системи. Будемо називати їх властивостями вибору цілі. Такими властивостями, зокрема, є деякі додаткові змінні або стани в системах, додаткові елементи або з'єднання у структурованих системах, додаткові елементи або процедури в мети системах тощо.

Отже, для систем, що володіють властивостями ціленаправленості, необхідно відділяти змінні вибору цілі від інших змінних із вимогою, щоб змінні вибору цілі сприяли її досягненню. Дослідження способів породження станів змінних вибору цілі надзвичайно важливе для розуміння природи такого класу систем і, зокрема, для розвитку методів їхнього проектування. Таким системам завжди притаманні принципи (схема, форма) їх створення у термінах породжуваних станів змінних пошуку цілі. Виходячи з викладеного, можна сформулювати поняття ціленаправлених систем.

**Ціленаправлені системи** - це системи, орієнтовані розробником на виконання суворо визначених цілей. Вони мають чітко обумовлене цільове призначення для заданих умов, а також характеризуються набором обмежень за номенклатурою цілей і заданим діапазоном допустимих змін умов функціонування. Прикладами таких систем є найпростіші автомати і напіваавтомати з жорсткою програмою керування або найпростіші механізми. До них можна, зокрема, віднести верстати-автомати для виробництва лише конкретних механічних деталей, побутові і промислові холодильники, автономні системи опалення Щ будинків із напіваавтоматичною

підтримкою температури у певному діапазоні, автомобілі різного цільового призначення. Принципово іншим є клас цілеспрямованих систем.

**Цілеспрямовані системи** - це системи, які мають властивості сприймати вимоги середовища, зовнішнього по відношенню до неї, та формувати цілі для досягнення цих вимог за умови суттєво мінливих ситуацій, а також визначати альтернативи всіх дій зовнішнього середовища і здійснювати доцільний вибір альтернативи власних дій для досягнення цілей за наявних умов. **Цілеспрямованість** - динаміка системи, як послідовність змін стану, носить не випадковий характер.

Найважливішою властивістю цілеспрямованих систем є здатність динамічно змінювати цілі і способи їхнього досягнення у разі зміни ситуації. Цілеспрямовані системи принципово відрізняються від ціленаправлених високою гнучкістю, динамічністю і здатністю реагувати на зміну зовнішнього середовища шляхом адаптації потреб, цілей і дій у ситуаціях, що складаються. Системи даного класу можуть змінювати функції, властивості і навіть структуру як функціональних елементів, так і системи загалом.

### **Питання для самоперевірки та обговорення**

1. Що таке ієрархічна структура в системі і які типи ієрархій (дерево рішень, організаційна, класифікаційна) найчастіше застосовують у системному аналізі?
2. Поясніть, як концепція ієрархії допомагає у вирішенні складних задач (розбиття проблеми, делегування, керування ризиками) - наведіть приклад.
3. Як побудувати дерево рішень для задачі вибору архітектури програмного продукту? Які критерії і показники слід включити?
4. Які обмеження та помилки виникають при надмірній ієрархізації (зайве дроблення системи)? Як цього уникнути на практиці?
5. Обговоріть роль ієрархії в управлінні організацією: як ієрархічна модель співвідноситься з модельною картиною бізнес-процесів?

## **Тема 5. Оцінка якості систем**

### **5.1. Якість системного аналізу**

В рамках системного аналізу може бути дано якісну оцінку розвитку будь-якої системи. Системними є людське мислення та результати пізнання. Таким чином, сам процес системного аналізу відповідає ознакам системності та є системою [11].

Результатом системного аналізу є вибір з безлічі альтернатив тієї, яка оптимально задовольняє умовам поставленого завдання. Вибір альтернативи, зроблений на основі системного аналізу, дозволяє системі розвиватися

оптимальним чином, раціонально розподіляти та витратити ресурси, а також формувати передумови для свого подальшого розвитку та подальшого здійснення основних функцій системи. Отже, виникає питання достовірності вибору альтернативи, повноті обліку всіх чинників, т. е. якість системного аналізу.

Якість системного аналізу – це інформація про власні характеристики системного аналізу, що задовольняє вимогам до результатів аналізу. Рівень якості системного аналізу істотно впливає на тенденції розвитку соціально-економічної системи.

Якість системного аналізу можна як системи сегментів, надають сукупний вплив формування системи якості досліджуваного поняття (рис. 2).

Розглянемо зміст кожного із сегментів.

**Концепція.** Це певний спосіб розуміння, трактування будь-яких явищ, основна думка, керівна ідея їхнього висвітлення, провідний задум, конструктивний принцип різних видів діяльності.

**Теорії.** Це система основних ідей в тій чи іншій галузі знання, форма наукового знання, що дає цілісне уявлення про закономірність. Критерій істинності та основа розвитку теорії - практика. До системного аналізу соціально-економічних систем відноситься цілий комплекс теорій з різних сфер науки, наприклад, теорія ймовірності, теорія очікування, теорія пізнання та ін.



Рис. 5.1. Якість системного аналізу та його сегменти

**Закони.** Фахівцями виділяються дві категорії законів: закони природи, зокрема природи людського суспільства, та закони, сформульовані людьми. Закони природи об'єктивні, несуперечливі і непорушні, а закони,

сформульовані людьми, мінливі, непослідовні та суперечливі. Тому закони функціонування соціально-економічних систем є об'єктивними законами, а закони, що регулюють діяльність соціально-економічних систем, – суб'єктивні закони.

Виявлення законів та закономірностей розвитку соціально-економічних систем, можливо, здійснено в результаті проведених досліджень у цій сфері та перенесення закономірностей, характерних для інших галузей знань, на основі принципу універсальності розвитку складних систем.

Стосовно соціально-економічних систем закон – це «необхідний, суттєвий і стійкий зв'язок між елементами внутрішнього і зовнішнього середовища, що обумовлює їх упорядковану зміну».

З урахуванням специфіки системного аналізу щодо нього слід відносити цілий комплекс законів із різних галузей науки. Наприклад, закони великих чисел, синергії, попиту та пропозиції, самозбереження, діалектики і т.д.

**Парадигми.** Парадигма - це і концептуальна схема, модель постановки проблем та їх вирішення, методів дослідження, що панує протягом певного історичного періоду в науковому співтоваристві .

Наприклад, Н. І. Миронова характеризує парадигму як «ступінь спрощення», яку собі може дозволити епоха. Це призма, крізь яку сучасники дивляться на світ, міра, з якою вони підходять до явищ». Системний аналіз базується на системній парадигмі, яка відображає сучасний погляд на реальність, на процеси суспільної самоорганізації. В основу системної парадигми покладено два наукові підходи. Перший, природно-науковий, пов'язаний з уявленням про еволюцію та самоорганізацію живих систем та організмів. Другий, фізико-математичний, пов'язаний з теорією складнощів, нелінійністю, ентропією, нестійкістю, самоорганізацією, переходом хаосу в порядок іншого рівня.

**Інструменти.** Як інструменти системного аналізу використовується багатий арсенал накопичених способів обробки та аналізу інформації. Застосовуваний інструментарій системного аналізу залежить від складу інформації про об'єкт дослідження (соціально-економічної системи). Залежно від повноти умов, що характеризують стан системи, соціально-економічні системи класифікуються по комплексу параметрів від стану визначеності до стану невизначеності.

**Об'єкт дослідження.** «Об'єктом системного аналізу є: екологічні, соціальні, економічні, технологічні, інформаційно-технічні та інші системи, їх інфраструктура та ресурсне забезпечення; виробничо-технологічна, управлінська та інші види діяльності; процеси підготовки та прийняття рішень як системи правил, процедур та прийомів; інформаційні технології та системи обробки інформації».

**Внутрішнє та зовнішнє середовище системи.** На якість системного аналізу впливають внутрішній стан системи чи зовнішнє середовище. Внутрішнє середовище системи – це джерело її життєвих сил, вона містить у

собі потенціал, який дозволяє системі функціонувати та виживати протягом певного проміжку часу. Але в той же час внутрішнє середовище системи може спричинити її загибель, якщо її внутрішні зв'язки, ресурси не можуть створити умови, необхідні для її функціонування.

Зовнішнє середовище системи – це сукупність факторів різної природи (технічних, технологічних, екологічних, соціальних, економічних та ін), які не входять до складу системи, але впливають на ефективність її функціонування. Розглядаючи природу довкілля системи, необхідно враховувати її динамізм і складність.

**Управлінські рішення.** Логічним результатом системного аналізу є ухвалення управлінського рішення. Адже тільки на базі якісно проведеного системного аналізу можливе прийняття рішення, яке максимально здатне врахувати всі зазначені умови та обмеження. Наприклад, з метою оцінки результатів фінансової складової діяльності банку необхідно враховувати інфляційний чинник. Розраховуючи банківський відсоток за кредитами, необхідно враховувати розмір інфляції, інакше фінансова діяльність банку за минулий рік може бути збитковою.

Таким чином, аналізоване поняття «якість системного аналізу» є складовою категорією, оцінку якої можна проводити з позиції укрупнених етапів його проведення: якості відбору інформації, якості обробки інформації та якості прийняття управлінського рішення.

Ключовими критеріями формування якості системного аналізу є: якість дотримання процедур системного аналізу, терміновість та економічність. Терміновість передбачає проведення системного аналізу якомога стисліші терміни, а економічність – прагнення провести системний аналіз при мінімально допустимому рівні витрат.

На якість дотримання процедур системного аналізу впливають такі чинники, як якість підготовки інформації, якість обробки інформації та якість ухвалення управлінського рішення. Якість підготовки інформації передбачає виконання таких вимог: - достовірність інформації, тобто подана інформація має відповідати дійсності;

- Повнота інформації передбачає відсутність часткової інформації у формуванні будь-яких показників, відсутність пропущених даних;

- рівень обширності інформації чи наявності максимально можливої інформації, зібраної з предмету системного аналізу;

- цінність інформації, т. е. рівень її доступності широких верств населення.

Якість обробки інформації включає низку критеріїв оцінки.

Це, по-перше, оптимальність обліку ключових факторів, яка передбачає необхідність застосування при проведенні системного аналізу обмеженої кількості ключових факторів, які найбільш яскраво відображають тенденції, що склалися, і не дублюють один одного.

По-друге, достовірність, що передбачає правильність вибору ключових факторів у побудованій моделі, що визначає відповідність отриманих попередніх значень реальної реакції системи.

По-третє, надійність, т. е. здатність вибору оптимальної методики отримання результатів системного аналізу.

По-четверте, простота моделі, що означає необхідність спрощення побудованої моделі у допустимих межах.

По-п'яте, точність розрахунків – це відповідність обраних для аналізу числових значень реально протікають процесів.

І, по-шосте, наочність, яка передбачає організацію порядку проведення системного аналізу таким чином, щоб поданий висновок був очевидним та зрозумілим експертам.

Якість прийняття управлінського рішення можна охарактеризувати як оптимально прийняте рішення, що складається із складових:

- компетентність особи, яка приймає рішення (ОПР), або високий рівень знань у галузі проведення системного аналізу;
- здатність ОПР виявляти майбутні тенденції у досліджуваній галузі, що включає здатність до розробки нестандартних рішень;
- проведення адекватної оцінки альтернативних рішень;

результативність (ефективність) системного аналізу, що передбачає повноту обліку майбутніх тенденцій системи у прийнятому управлінському рішенні, і навіть те, що прийняте управлінське рішення має максимально сприяти досягненню бажаної мети. Критерії якості системного аналізу у графічному вигляді представлені на рис. 5.2

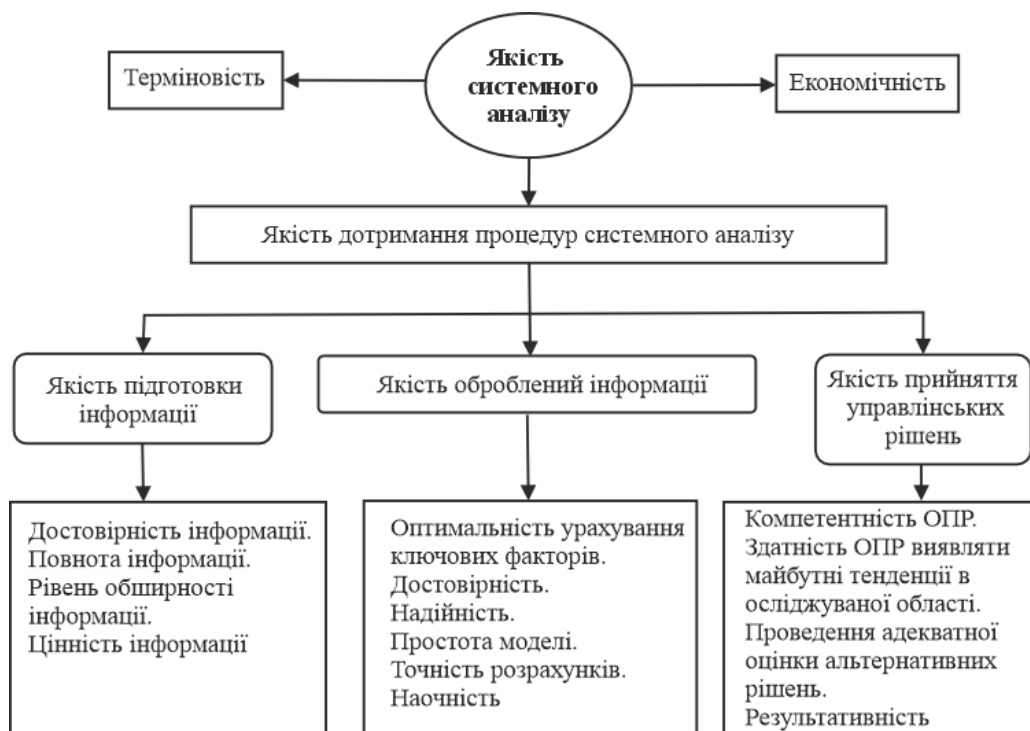


Рис. 5.2 Критерії якості системного аналізу

Для оцінки якості системного аналізу можливе використання інтегрального показника, який визначатиметься за такою формулою:

$$I_{\text{я}} = I_{\text{р}} I_{\text{с}} I_{\text{е}},$$

де  $I_{\text{я}}$  – інтегральний показник якості системного аналізу;

$I_{\text{р}}$  - Інтегральний показник результативності системного аналізу;  $I_{\text{с}}$  – показник терміновості системного аналізу;  $I_{\text{е}}$  – показник економічності системного аналізу.

Подані показники вимірюються у частках чи у відсотках.

Інтегральний показник результативності визначається як добуток показників якості підготовки інформації  $I_{\text{п}}$ , якості обробки інформації  $I_{\text{о}}$  та якості прийняття управлінського рішення  $I_{\text{у}}$ :

$$I_{\text{р}} = I_{\text{п}} I_{\text{о}} I_{\text{у}}.$$

## 5.2. Оцінка якості інформаційних систем.

Якість інформаційних систем обов'язково пов'язують із цілим комплексом дефектів, які показують свої властивості у процесі проєктування, а потім уже під час експлуатації. Якості інформаційної системи виявляються лише за наявності компонентів довкілля. Особливо важливими є властивості, що взаємодіють з елементами технічного характеру, персоналом, інформаційним оточенням [12]. У тому числі важливі властивості дефектів, що визначаються такими факторами:

- кількість розробників інформаційних систем;
- умови забезпечення процесу опрацювання;
- характеристики спеціальних засобів та елементів;
- складність виконуваних з допомогою інформаційних систем процедур;
- рівень агресивності та складності зовнішнього середовища.

На дефектабельність впливають структурні та конструктивні особливості представленої інформаційної системи, характеристики помилок, їх ймовірність та інтенсивність, що призводить до виникнення дефектів.

При оцінці дефектоскопічності проявляється наявність дефектів як відмови чи збою при налагодженні, експлуатації чи під час випробувань.

Оцінка якості інформаційних систем є досить складним видом роботи, так як інтереси і ціла купа вимог користувачів можуть бути найрізноманітнішими. Тому сформувавши універсальний підхід оцінки якості інформаційних систем неможливо. Основні завдання оцінка бере із завдань, поставлених для програмного забезпечення, оскільки багато стандартів у разі є ідентичними. В даний час застосовується лише кілька моделей якості для оцінки програмного забезпечення, що базуються на метриці, показують характеристики, що базуються на комплексі показників та критеріїв якості.

Машинне навчання може суттєво підвищити якість систем (точність прогнозів, адаптивність, автоматизацію, виявлення аномалій), але водночас

вводить нові ризики (упередження даних, непередбачувана поведінка, складність валідації, проблеми з прозорістю й підтримкою) [13].

Для критеріїв є визначення, яке показує, що вони є незалежною частиною інформаційної системи. За допомогою критеріїв можна виміряти характеристики якості ІС.

Критерії оцінки інформаційної системи останнім часом визначають ієрархічну модель, де компоненти пов'язані між собою. Спочатку визначають характеристики якості, серед яких можуть бути: зручність експлуатації, загальна та вихідна користь. Потім формують стандарти показників, яких відносять практичність, коректність, комфорт функціонування, можливість взаємодії, адаптації. Для кожного показника є своя окрема група критеріїв.

Оцінка якості інформаційних систем визначається з допомогою метрики. Такі методи контролю якості інформаційних систем дозволяють скласти якісні та кількісні оцінки.

Перший вид - метрики, що застосовують інтервальну шкалу. За допомогою такої шкали можна побачити час напрацювання вщерть, ймовірність помилок, обсяг інформації, що має.

Другий вид – метрики із порядковою шкалою. Завдяки такій шкалі можна визначити та вибудувати характеристики порівняно з опорними значеннями.

Третій вид представляють метрики з номінальною та категорованою шкалою. За допомогою цього методу можна визначити властивості та ознаки об'єкта без градації. Наприклад, інтерфейс може бути простим, складним, помірковано складним. Конструктивні критерії якості інформаційної системи використовуються для компонентів інформаційної інфраструктури і не залежать від цільового призначення.

Існує спеціальна модель класифікації критеріїв якості інформаційних систем:

Саме така модель класифікації критеріїв якості інформаційних підсистем в ієрархічному підході дає можливість оцінити рівень основних цілей та поставлених завдань.

Для забезпечення надійності ІВ застосовують сертифікацію. В даний час відсутні стандарти, які повністю відповідали б якісній оцінці якості. Сьогодні існують ДСТУ, але вони морально застаріли, тому багато хто використовує в оцінці системи міжнародні стандарти ISO9000.

### **Стандарти оцінки якості інформаційних систем.**

Оцінка якості інформаційної системи за допомогою міжнародних стандартів ISO9000 дозволяє виконувати роботи, в яких головним критерієм є можливість управління якістю продукту. Варто зазначити, що такі стандарти для інформаційної системи доповнюють ISO14000, де відображаються екологічні вимоги безпосереднього виробництва продукції.

Але важливість таких критеріїв полягає не тільки в ідеї управління якістю, а й на його контроль. Стандартно контроль інформаційних систем ґрунтується на вимірі показників за допомогою таких інструментів як методи

контролю якості інформаційної системи. Це операції технологічного спрямування та вибракування виробів, що не відповідають заявленим стандартам. Варто відзначити, що контроль інформаційного середовища є більш дієвим методом, оскільки проводить контроль процесу технологічного виробництва, що дозволяє надалі знизити ймовірність появи вибракованої продукції.

Такі методи дають можливість покращити ефективність заходів, що проводяться, завдяки меншим витратам. Немає потреби проводити постійний контроль якості продукції. У цьому попереджається поява шлюбу, що знижує виробничі витрати. Саме такий контроль є стандартами ISO9000. Не варто думати, що вони морально застаріли, оскільки були ухвалені ще 1987 року. Такі стандарти проходять оновлення кожні п'ять років.

Дані стандарти дають можливість регламентувати та визначити всі питання, що стосуються створення, функціонування, експлуатації системи якості у промисловій сфері. Вони представлена форма вимог у структурі якості, щоб постачальник міг продемонструвати свої можливості, а учасники довілля змогли оцінити показники якості ІВ.

Американське суспільство, яке відстежує контроль якості, визначило основні цілі ISO9000. До них можна віднести допомогу розвитку міжнародного обміну послуг та продукції. У цьому враховуються сфери наукового, інтелектуального, ділового, технічного характеру.

Стандарти оцінки ІВ – це сукупність всіх властивостей, у яких обумовлена можливість її застосування задоволення потреб відповідно до її призначенням. Саме кількісні характеристики кожного такого властивості визначають основні види показників.

Що стосується моделі якості програмного забезпечення, то на сьогоднішній день найбільшого поширення набула багаторівнева модель. Вона представлена стандартом ISO 9126. Спочатку вона виділяє шість основних характеристик якостей програмного забезпечення, а далі їх атрибути, які включають відповідні метрики для подальшої оцінки всієї ІС.

Основні види показників, за яких можлива оцінка якості інформаційного середовища, критерії ІВ – це безпека, достовірність та надійність.

Якщо говорити про безпеку, то інформаційна інфраструктура контролю цю властивість визначає як здатність системи забезпечити максимальну конфіденційність, і навіть цілісність всієї наявної інформації. Тобто відбувається захист інформації від будь-якого несанкціонованого доступу.

До достовірності функціонування слід зарахувати такий набір властивостей, який визначає безпомилковість перетворень інформації. Достовірність функціонування визначається достовірністю вихідної інформації.

До надійності відносять набір властивостей, що зберігають час всіх параметрів, які необхідні виконання певних функцій у відповідних умовах і заданих режимах.

Оскільки основними показниками якості інформаційних систем є надійність, достовірність, безпека, такі моменти потрібно розглянути докладніше. Також слід зазначити, що якість та ефективність інформаційних систем йдуть поруч, тому окремо розглядати характеристики даних показників неможливо. Вони є модульністю і спільністю, які дуже важливі, якщо застосовуються різні методи контролю якості інформаційних систем.

Ефективність є такий набір властивостей структури інформації, який дає можливість виконувати поставлену мету в певних умовах з конкретною якістю. Для показників ефективності важливим є ступінь пристосованості системи до здійснення поставлених цілей. При цьому вони мають на увазі модульність показників для оптимального функціонування ІВ. Найважливішим із показників вважатимуться економічну ефективність системи. Саме вона дає можливість показати доцільність виконаних витрат на створення та весь процес функціонування та експлуатації системи.

Такі показники якості інформаційних систем, як надійність настільки важливі для характеристики системи, що в даний час розроблена навіть спеціальна теорія надійності. Надійність можна віднести до комплексу властивостей, оскільки цей показник якості є набором простіших за структурою властивостей. До них належать безвідмовність, можливість проводити ремонтні роботи, довговічність. Якщо говорити про безвідмовність, то контроль інформаційних систем дозволяє зрозуміти, наскільки система може зберігати свій працездатний стан протягом певного періоду або процесу напрацювання.

До ремонтпридатності відносять пристосованість системи до того, щоб попереджати та виявляти причини прояву відмови чи ушкодження. Також сюди входить можливість технічного обслуговування та виконання ремонтів.

Довговічність – одна з властивостей системи, завдяки якій, за наявності відповідного технічного обслуговування, вона зберігає належний стан максимально довго. Тобто має наступити такий момент, коли у системи не залишається ресурсів для подальшого виконання роботи та функціонування.

Показники якості інформаційних систем також включають достовірність. Це такі властивості інформації, що відображають реальні об'єкти з певною точністю. Інформаційні системи контролю якості як достовірність передбачають наявність такого комплексу показників: поодинокі, коректованості ІВ та комплексні. Звичайно, для забезпечення достовірності інформації необхідний контроль.

Також до цих якостей можна віднести портативність, зручність використання та зручність супроводу. Завдяки яким ПЗ, а надалі і всю ІС можна аналізувати, оцінювати, переносити та змінювати при виникненні дефектів

### 5.3. Показники якості програмних і технічних систем

Адекватний набір показників якості програм залежить від функціонального призначення та властивостей кожного ПС. Відповідно до принципів особливостей ПС при системному проектуванні повинні вибиратися наменклатура та значення показників якості, які згодом відображаються в технічній документації та специфікації вимог на кінцевий продукт. Кожен критерій може використовуватися, якщо визначена його метрика і може бути зазначений спосіб її вимірювання та зіставлення з потрібним значенням. Для конкретних ПС домінуючі критерії якості виділяються та визначаються при системному проектуванні вимогами технічного завдання та його функціональним призначенням. Програми для ЕОМ як об'єкти проектування, розробки, випробувань та оцінки якості характеризуються такими узагальненими показниками:

- сферою застосування та соціальним призначенням програмного комплексу;
- конкретним типом розв'язуваних функціональних завдань із досить певною сферою застосування;
- обсягом та складністю сукупності програм, що вирішує єдине цільове завдання даного типу;
- необхідним складом та необхідними значеннями характеристик якості функціонування програм та величиною допустимої шкоди через недостатню їх якість;
- ступенем зв'язку розв'язуваних завдань із реальним масштабом часу або допустимою тривалістю очікування результатів розв'язання задачі;
- прогнозованими значеннями тривалості експлуатації та можливістю розвитку безлічі версій програм;
- передбачуваним тиражем виробництва та застосування програм;
- ступенем необхідної документованості програм.

Насправді важливо оцінювати якість програм у завершеному вигляді, а й у процесі їх проектування і розробки. Якість цього процесу значною мірою визначається сукупними витратами, необхідними досягнення заданої якості кінцевого продукту - програмного засобу. В результаті досяжна якість програм визначається якістю методів та засобів, що застосовуються при їх проектуванні та розробці.

Взаємозв'язок якості розроблених програм з якістю технології їх створення та з витратами на проектування та розробку стає особливо суттєвим, при необхідності отримання кінцевого продукту з особливо високими значеннями показників якості. Витрати різко зростають, коли показник якості наближається до межі, яка досягається при даній технології та рівні автоматизації процесу проектування та розробки. При переході до принципово іншої, більш прогресивної технології за тих самих витрат може бути досягнуто вищу якість і граничне значення можливої якості також підвищується.

Формалізації показників якості програмних засобів присвячено групу

стандартів. У базовому міжнародному стандарті ISO 9126:1991. IT. Оцінка програмного продукту. Характеристики якості та посібник щодо їх застосування - при відборі мінімуму стандартизованих показників висувалися та враховувалися наступні принципи:

- ясність та вимірюваність значень;
- відсутність перекриття між використовуваними
- відповідність ustalеним поняттям та термінології;
- можливість подальшої уточнення та деталізації.

Виділено характеристики, які дозволяють оцінювати ПС з позиції користувача, розробника та керуючого проектом (рис.8). Якість складних програмних засобів описується сукупністю показників - критеріїв, для кожного з яких мають бути визначені метрики та методи їх виміру. Ці критерії та метрики програм дозволяють описувати їх властивості як кінцевого продукту, незалежно від способу їх досягнення.

Поданий на рис. 5.3 набір 21 показників якості передбачається доповнити у кожній групі метрикою узгодженості показників у межах групи та між групами. Характеристики та субхарактеристики у стандарті визначені дуже коротко, без коментарів та рекомендацій щодо їх застосування до конкретних систем та проектів.



Рис. 5.3. Набір показників якості

Рекомендується 6 основних характеристик якості ПЗ, кожна з яких деталізується кількома (всього 21) субхарактеристиками:

- функціональна придатність - придатність для застосування, точність, захищеність, здатність до взаємодії та узгодженість зі стандартами та правилами проектування;
- надійність - рівень завершеності (відсутність помилок), стійкість до помилок і перезапуск;
- застосовність - зрозумілість, навченість та простота використання;
- ефективність - ресурсна та тимчасова економічність;

- супроводжуваність - зручність для аналізу, змінність, стабільність та тестованість;

- переносимість - адаптованість, структурованість, змішаність і впроваджуваність.

Підготовлено проєкт модернізації стандарту, в якому передбачається виділити чотири частини:

- модель показників якості та їх розвитку у життєвому циклі програмних засобів;
- зовнішні метрики якості, які відображають найбільш загальні характеристики програмного засобу;
- внутрішні метрики якості, що характеризують структуру та конструктивні особливості ПС;
- метрики якості ПС з позиції користувача:
- ефективність застосування ПС користувачем;
- продуктивність під час вирішення основних функціональних завдань;
- ступінь захищеності від зовнішніх загроз;
- задоволеність користувачів функціями та якістю вирішення завдань.

При системному проєктуванні, розробці технічного завдання та специфікацій з представленої номенклатури повинен бути обраний первинний набір показників якості, який в подальшому слід конкретизувати та розвивати. Нижче описано зміст найважливіших показників якості ПС з позиції системного проєктування.

Функціональна придатність – це набір атрибутів, що визначає призначення, номенклатуру, основні необхідні та достатні функції ПС, встановлені технічним завданням замовника чи потенційного користувача. У процесі проєктування ПС атрибути функціональної придатності конкретизуються специфікації на компоненти. Ці атрибути можна чисельно уявити точністю обчислень, відносним числом функцій, що поетапно змінюються, коефіцієнтом зміни специфікацій вимог розробниками і замовниками і т.д.

Найбільшою мірою функціональна придатність проявляється у коректності вирішення завдань, у надійності та безпеці застосування ПС. Функціональна придатність визначається насамперед якістю системного проєктування ПС. Критеріями, що відображають ці властивості можуть бути: здатність компонентів до взаємодії та ступінь стандартизації інтерфейсів, мобільність програм та їх захищеність від зовнішніх впливів. Здатність програмних та інформаційних компонентів до взаємодії можна оцінювати обсягом змін до ПС, які необхідно виконати при доповненні або виключенні певної функції, коли відсутні зміни операційного або апаратного середовища.

Коректність програм залежить від функціональної коректності застосовуваних компонентів і може розглядатися залежно від методів її досягнення та оцінювання: детерміновано, стохастично або в реальному часі. Функціональна коректність модулів визначається коректністю обробки вихідних даних та отримання результатів. Залежно від функціональних

завдань коректність модуля може оцінюватися детерміновано, повністю визначеним набором еталонних значень, або стохастично, при завданні еталонів розподілами випадкової величини. Функціональна коректність обробки даних пов'язана переважно з конкретизацією їх змісту у процесі виконання програм, і навіть під час підготовки даних зовнішнім абонентам.

Детермінована коректність програм визначається за частотою відхилення конкретних обчислюваних результатів від еталонних значень, заданих у технічному завданні, специфікаціях чи інших вихідних документах. Стохастична коректність характеризується величиною статистичного відхилення розподілів та його параметрів (середніх значень, середньоквадратичних відхилень) результатів функціонування програм від заданих як еталонів. При цьому не оцінюється кожен результат тестування, а вони узагальнюються та оцінюються інтегрально за деякою досить представницькою вибіркою. Коректність реального часу визначається величиною максимального або усередненого відхилення траєкторії зміни вихідних параметрів від заданої еталонної траєкторії обробки тестових вихідних даних. Результати тестування оцінюються інтегрально на деякому часовому інтервалі або з найбільшого відхилення від еталонної траєкторії.

Коректність структури програм визначається коректністю структури модулів та коректністю структури груп програм, побудованих з модулів. Для оцінки коректності структури програм використовується кілька приватних показників, що відрізняються ступенем охоплення тестами структурних компонентів програми при тестуванні. Коректність обробки даних визначається ступенем налагодження процесу обробки представницької вибірки значень змінних у допустимих діапазонах зміни. Для складних програм неможливо провести вичерпне тестування при всіх значеннях змінних, які можуть виникнути при реальному функціонуванні програм. Тому встановлюється ієрархія критеріїв та відповідних їм коректностей обробки даних у програмі.

Коректність міжмодульних інтерфейсів визначається двома видами зв'язків між модулями: з управління та з інформації. Зв'язки по управлінню складають виклики програмних модулів і повернення до тих, що викликали. Некоректності взаємодії модулів за інформацією дуже різноманітні, важко контрольовані та мають стохастичний характер. Ряд спотворень змінних при взаємодії модулів бракує відмовних ситуацій і відбивається лише спотворенні результуючих даних на виході модуля. Різноманітність та складність інформаційних зв'язків у великих ПС значно ускладнюють формалізацію досягнутої коректності програм. Критерії стають стохастичними та якість ПС оцінюється інтегрально за результатами загального функціонування.

Надійність програм. Показники надійності ПС значною мірою адекватні аналогічним характеристикам прийнятим інших промислових виробів. У цьому випадку реєструються тільки такі спотворення у процесі функціонування та виконання програм, що призводять до втрати

працездатності ПС або їх великих компонентів. З поняттям надійності та відмов програм тісно пов'язане поняття технологічної безпеки застосування ПС.

Основним принципом класифікації збоїв та відмов у програмах за відсутності їх фізичного руйнування є поділ за тимчасовим показником тривалості відновлення після будь-якого спотворення програм, даних чи обчислювального процесу, що реєструється як порушення працездатності. При тривалості відновлення, меншої за заданий поріг, аномалії при функціонуванні програм слід відносити до збоїв, а при відновленні перевищує по тривалості граничне значення, що відбувається спотворення відповідає відмові. Класифікація програмних збоїв та відмов за тривалістю відновлення призводить до необхідності аналізу динамічних характеристик абонентів, які є споживачами даних, оброблених досліджуваним ПС, а також тимчасових характеристик функціонування програм. Тимчасова зона перерви нормальної видачі інформації та втрати працездатності, яку слід розглядати як зону збою, тим ширша, що більш інерційний об'єкт перебуває під впливом повідомлень, підготовлених даними ПС.

Надійність функціонування ПС найбільш широко характеризується стійкістю або здатністю до безвідмовного функціонування та відновлюваністю працездатного стану після збоїв, що відбулися. Відновлюваність характеризується повнотою та тривалістю відновлення функціонування програм після перезапуску (рестарту). Перезапуск повинен забезпечувати відновлення нормального функціонування ПС, потім потрібні ресурси ЕОМ і час. Найбільш широко використовується критерій тривалості напрацювання на відмову. Для визначення цієї величини вимірюється час працездатного стану системи між послідовними відмовами чи початками нормального функціонування системи після них. Узагальнення характеристик відмов та відновлення проводиться у критерії коефіцієнт готовності. Цей показник відбиває можливість мати відновлювану систему у працездатному стані у довільний час.

До показників надійності близькі критерії рівня захищеності та безпеки застосування ПС, які характеризуються ефективністю контролю за доступом до даних, ступенем забезпечення їх таємності та безпеки, а також деякими іншими властивостями стійкості до впливів з боку зовнішнього середовища та дефектів програм та даних.

Застосовність - зручність використання ПС - поняття досить абстрактне і формалізується, що важко формалізується, проте в результаті визначає функціональну придатність і корисність застосування ПС. У цю групу показників входять критерії з різних сторін, що відображають зрозумілість, здатність до навчання та ефективність або простоту використання. Зрозумілість ПС можна описати чіткістю концепції, широтою демонстраційних можливостей та точністю представлення у документації можливих функцій. Крім того цей показник характеризується розпізнаваністю параметрів, що модифікуються, і адаптованістю ПС до

конкретного середовища та умов застосування.

До цих критеріїв примикають критерії навчання, які можуть містити тривалість підготовки користувача до повноцінної експлуатації ПС. Цей час залежить від можливості попереднього навчання та вдосконалення в процесі експлуатації, від можливостей оперативної допомоги та підказки (Help) при використанні ПС, а також від доступності та зручності використання посібників та інструкцій з експлуатації.

Критерії простоти та зручності використання враховують фізичні та психологічні характеристики користувачів та відображають рівень комфортності умов експлуатації ПС. Вони характеризують легкість управління ПС та обсяг параметрів управління, що реалізуються за умовчанням, інформативність повідомлень користувачу та уніфікованість управління екраном, ступінь доступності зміни функцій відповідно до кваліфікації користувача та кількість операцій, необхідних для запуску певного завдання. Крім того, зручність використання характеризується низкою динамічних параметрів: часом введення та відгуку на завдання, тривалістю вирішення типових завдань, часом на реєстрацію результатів.

Ефективність використання ресурсів. Незважаючи на швидке зростання ресурсів пам'яті та продуктивності ЕОМ, дуже часто потреби в них для вирішення конкретних завдань обганяють їхнє технічне збільшення і залишається актуальним завдання економного використання ресурсів. Тому серед критеріїв якості ПС помітну роль відіграють критерії ефективності використання ресурсів пам'яті та продуктивності ЕОМ при реалізації певного ПС. Тимчасова економічність ПС визначається тривалістю виконання заданих функцій. Вона залежить від швидкості обробки даних, що впливає безпосередньо на інтервал часу завершення конкретного обчислювального процесу, і від пропускнуєї спроможності тобто. від числа завдань, яке можна реалізувати на даній ЕОМ у заданому інтервалі часу. Ці показники якості тісно пов'язані з часом реакції (відгуку) ПС на запити для вирішення основних функціональних завдань [14].

Пропускна здатність комплексу програм на конкретній ЕОМ відображає кількість повідомлень або запитів на вирішення певних завдань, що обробляються в одиницю часу, яка залежить від деякого показника довкілля. При цьому виділяються складові часу центрального процесора, обміну із зовнішньою пам'яттю, взаємодії із зовнішнім середовищем та терміналами тощо.

Ресурсна економія відображає кількість і ступінь зайнятості ресурсів центрального процесора, оперативної, зовнішньої та віртуальної пам'яті, каналів введення - виведення, терміналів та каналів локальної мережі. Цей критерій визначається структурою та функціями ПС, а також архітектурними особливостями та доступними ресурсами реалізує ЕОМ.

Для сучасного використання ПС важливим показником стала переносимість або мобільність програм в інше операційне середовище чи іншу апаратну платформу. Ця властивість може оцінюватися обсягом

необхідних доопрацювань ПС, які слідують виконати задля забезпечення повноцінного функціонування ПС після перенесення. Мобільність може оцінюватися на рівні вихідних текстів програм або на рівні об'єктного коду. При аналізі та оцінці переносимості програм та даних насамперед необхідно враховувати:

- ступінь подібності архітектури та співвідношення основних параметрів апаратних платформ, між якими передбачається перенесення;
- ступінь подібності та уніфікації інтерфейсів операційних платформ із додатками, між якими передбачається перенесення прикладних програм та даних;
- наявність або відсутність при розробці вихідних програм та інформації баз даних орієнтації на майбутнє перенесення;
- обсяг і комплексованість додатків з операційним та зовнішнім середовищем при передбачуваному їх спільному перенесенні.

Показники якості бази даних. Для баз даних поки що відсутні міжнародні стандарти, що регламентують їх показники якості. Тому нижче представлений набір характеристик, який найчастіше використовується практично при виборі та оцінці баз даних. У системах баз даних домінуючого значення набувають самі дані, їх зберігання та обробка. Тому БД під час аналізу їх якості доцільно розділити на дві компоненти:

- програмні засоби системи управління базою даних (СУБД), незалежні від сфери їх застосування та смислового змісту накопичуваних та оброблюваних даних та
- інформація бази даних (БД), доступна для обробки та використання у конкретній проблемно - орієнтованій сфері застосування.

Практично весь набір показників якості ПС, викладений вище, тією чи іншою мірою, може використовуватися при аналізі та оцінці якості СУБД. Особливості полягають у зміні акцентів при виборі та впорядкуванні цих показників якості. Майже у всіх випадках найважливішими показниками якості СУБД є функціональні характеристики процесів формування та зміни інформаційного наповнення БД адміністраторами, а також доступу до даних та подання результатів користувачам БД. Якість інтерфейсу спеціалістів із БД, що забезпечується засобами СУБД, оцінюється, значною мірою, суб'єктивно, проте є низка характеристик, які можна оцінювати досить коректно.

Відмінності вимог до показників якості призвели до створення широкого спектра локальних, спеціалізованих і розподілених СУБД. Спеціалізовані СУБД характеризуються відносно вузькою сферою застосування та чіткішим виділенням домінуючої групи показників якості. У універсальних СУБД спектр показників якості ширший, що дозволяє відповідно розширювати сферу застосування конкретного типу СУБД.

Другий компонент БД є власне накопичувана і оброблювана інформація в базі даних. Показники якості, що виділяються, повинні мати практичний інтерес для користувачів БД і бути впорядковані відповідно до

пріоритетів практичного застосування. Крім того, кожен виділений для перевірки показник має бути придатним для достатньо достовірного вимірювання та порівняння з необхідним значенням при випробуваннях та сертифікації.

Функціональні показники якості інформації БД включають:

- Повноту накопичених описів об'єктів - відносна кількість об'єктів або документів, що є в БД, до загального числа об'єктів за цією тематикою або по відношенню до об'єктів в аналогічних БД за тією ж тематикою;

- достовірність - ступінь відповідності даних про об'єкти в БД реальним об'єктам поза ЕОМ в даний час, що визначається змінами самих об'єктів, некоректностями записів про їх стан або некоректностями розрахунків їх характеристик;

- ідентичність даних - відносне число описів об'єктів, які містять помилки, до загального числа документів про об'єкти в БД;

- Актуальність даних - відносна кількість морально застарілих даних про об'єкти в БД до загального числа накопичених та оброблюваних даних.

До конструктивних показників якості інформації в БД відносяться, в основному, об'ємно-тимчасові характеристики даних, що зберігаються і оброблюються:

- обсяг бази даних - кількість записів описів об'єктів чи документів у базі даних, доступних для зберігання та обробки;

- оперативність - ступінь відповідності динаміки зміни даних у процесі збирання та обробки станам реальних об'єктів або величина запізнення між появою або зміною характеристик реального об'єкта та його відображенням у базі даних;

- періодичність - проміжок часу між поставками двох послідовних, досить різняться інформацією версій БД;

- глибина ретроспективи - інтервал часу від дати випуску та/або запису до бази даних раннього документа до теперішнього часу,

- динамічність - відносне число змінюваних описів об'єктів до загального числа записів у БД за деякий інтервал часу, що визначається періодичністю видання версій БД.

Крім того, до конструктивних належать усі показники захищеності інформації. Захищеність реалізується, в основному, програмними засобами СУБД, однак у поєднанні з засобами організації даних, що їх підтримують. У розподілених базах даних показники захищеності тісно пов'язані з характеристиками цілісності даних. Ці показники відбивають ступінь тотожності даних у пам'яті віддалених компонентів розподіленої БД.

При реальному функціонуванні БД важливу роль відіграють тимчасові характеристики взаємодії кінцевих користувачів та адміністраторів БД у процесі експлуатації бази даних прямого призначення. Ці характеристики залежать від якості СУБД, а також від обсягу, структури та показників якості інформації, що використовується. Вище вони відображені критерієм ефективності використання ресурсів ЕОМ програмними засобами, у разі

СУБД. Для баз даних найважливішим ресурсом є пам'ять ЕОМ, займана інформацією БД, і навіть використання цього ресурсу. Ці показники якості впливають тимчасово реакції БД різні види запитів користувачів і пропускну здатність БД під час експлуатації.

## **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Які ключові компоненти якості системи (згідно з ISO/IEC 25010) і як кожен із них можна кількісно виміряти (приклади метрик)?
2. Опишіть методику оцінки якості системного аналізу: які критерії застосовуються для оцінювання коректності та повноти аналітичних артефактів?
3. Назвіть показники якості програмних систем (надійність, продуктивність, безпека, підтримуваність) і запропонуйте по 1–2 вимірювальні метрики для кожного.
4. Як проводити порівняльну оцінку двох інформаційних систем для вибору оптимальної? Які вагові коефіцієнти і методи багатокритеріального аналізу слід застосувати?
5. Які типи помилок у вимірюванні якості найчастіше зустрічаються (систематична похибка, неповні дані, упередженість) - як їх виявити та мінімізувати при оцінці?

## 2. ІНФОРМАЦІЙНИЙ СИСТЕМНИЙ АНАЛІЗ

### Тема 6. Марківські процеси в системах. Інформаційний критерій ефективності системи

**Марківські процеси** - це стохастичні (ймовірнісні) процеси, що характеризуються "марківською властивістю" - відсутністю пам'яті. Це означає, що ймовірність переходу системи в майбутній стан залежить лише від її поточного стану, а не від того, як система потрапила в цей стан. Вони широко використовуються в системному аналізі для моделювання динамічних процесів у різних системах, таких як комп'ютерні мережі, системи масового обслуговування, виробничі лінії та багато інших.

#### 6.1. Основні поняття марківських процесів

**Математичні моделі**, стан яких у будь-який момент часу  $\Delta t$  повністю визначено, за умовою визначення їх стану у попередні моменти часу  $\Delta t-1$  називаються **детермінованими**.

**Вірогідна модель**, на відміну від детермінованої, не дозволяє точно передбачити зміну окремих параметрів реального процесу. Але з її допомогою можна зробити точний прогноз щодо очікуваних середніх значень параметрів процесу. Крім того, дуже широке коло процесів є за своєю ймовірністю, і їх адекватний опис неможливий при використанні детермінованих моделей.

**Потоком подій** називається послідовність однорідних подій, що з'являються одна за одною у випадкові моменти часу. З потоком можна пов'язати різні випадкові події (наприклад, подія А, що полягає в тому, що протягом часу  $t$  до  $t_0+\Delta t$ , відбудеться хоча б одна подія; подія В, яка полягає в тому, що за цей же проміжок часу відбудеться рівно  $m$  подій і т.д.). Можуть бути обчислені ймовірність цих подій.

**Потік подій** називається **стаціонарним** якщо його імовірнісні показники залежить від вибору точки відліку, тобто, якщо ймовірність попадання певної кількості подій на будь-якому інтервалі часу залежить від довжини інтервалу і не залежить від того, на якій ділянці від початку координат обирається цей інтервал подій.

**Потік подій** називається **ординарним**, якщо ймовірність попадання 2-х подій на будь-яку елементарну ділянку часу  $\Delta t$  зневажливо мала проти ймовірності потрапляння однієї події. Таким чином – події в потоці з'являються не групами, а поодиночі та точний збіг моментів появи 2 подій має нульову ймовірність.

**Потік подій** називається потоком **без післядії**, якщо кількість подій, які потрапляють на будь-який інтервал часу, та не залежать від кількості подій які потрапили на будь-який інший інтервал, що не перетинається з обраним.

Відсутність післядії означає, що події, які утворюють потік, з'являються в ті чи інші моменти часу незалежно один від одного [15].

Стаціонарний, ординарний потік, що не має післядії, називається **найпростішим**. Проміжки часу між подіями в найпростішому потоці розподіляються за  $\lambda$  показовим законом з функцією розподілу  $f(t) = \lambda e^{-\lambda t}$ , де  $\lambda$  - величина, обернена середньому значенню інтервалу між подіями в потоці:

$$\lambda = 1 / M / [t].$$

*Ординарний потік без післядії називається пуассонівським.* Найпростіший потік є окремим випадком пуассонівського, а саме - **стаціонарний пуассонівський потік**.

Середня кількість подій (**математичне очікування** числа подій), що припадає на одиницю часу, називається **інтенсивністю потоку** –  $\lambda$ .

Припустимо наявність системи  $S$ , стан якої змінюється з часом (під системою розуміється: технічний пристрій, підприємство, ЕОМ і т.д.). Якщо стан системи  $S$  змінюється в часі випадково, то в системі протікає випадковий процес (процес функціонування ЕОМ, процес обслуговування клієнтів і т.д.) Конкретний перебіг кожного з таких процесів залежить від низки задалегідь не передбачуваних факторів (надходження замовлень, випадковий вихід з ладу обладнання, зміни вимог до кінцевого результату і т.д.).

*Випадковий процес, що протікає у системі  $S$ , називається марківським процесом або процесом без післядії, якщо він має відповідну властивість:* для кожного окремого моменту часу  $t$  ймовірність будь-якого стану системи в майбутньому ( $t > t_0$ ) залежить тільки від її стану в теперішньому ( $t = t_0$ ) і не залежить від того, як система потрапила в цей стан (як розвивався процес у минулому).

*Випадковий процес називається процесом з дискретними станами, якщо можливі стани системи утворюють множини  $S_1, S_2, S_3, \dots, S_n$ , тобто всі стани системи можна пронумерувати, а сам процес полягає у тому, що система кроком переходить з одного стану до іншого.*

Припустимо що система  $S$  має  $n$  станів:  $S_1, S_2, S_3, \dots, S_n$ . У будь-який момент часу система може перебувати в одному з цих станів. Позначимо  $P_i(t)$  - ймовірність того, що в момент часу  $t$  система перебуває в стані  $S_i$ . Очевидно, для фіксованого моменту часу виконується умова:  $\sum P_i(t) = 1$ .

Сукупність ймовірностей  $P_i(t)$  для кожного моменту часу характеризує поточний стан випадкового процесу, що протікає у системі.

Візуально граф станів системи зображує можливі стани системи  $S$  та її можливі переходи з певного стану в інший (рис. 6.1).

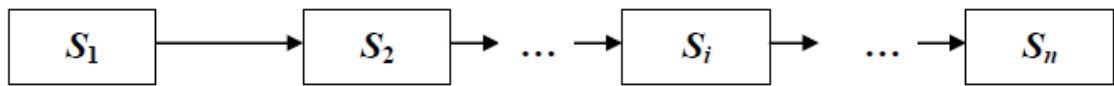


Рис 6.1 - Граф станів системи  $S$

Способи математичного опису марківського випадкового процесу, що протікає в системі з дискретними станами, залежать від того, в який певний момент часу відбуваються переходи системи зі стану в стан.

**Матрична модель:** ймовірність переходу між станами в одиничний проміжок часу, що часто задається за допомогою матриці  $P$ . Для двох станів  $S_1$  і  $S_2$  у марківському процесі, ймовірність переходу з  $S_1$  в  $S_2$  через інтервал часу  $t$  позначається  $P(S_1, S_2)$ , де  $P$  - це матриця переходів.

**Види марківських випадкових процесів.** Марківські випадкові процеси поділяються на класи. Перша класифікаційна ознака - характер станів.

**Випадковий процес (ВП)** – процес із дискретними станами системи. Якщо можливі стани системи  $S_1, S_2, S_3 \dots$  можна перерахувати, а сам процес полягає в тому, що система  $S$  миттєво (стрибок) переходить з одного стану до іншого, а моменти часу переходів є випадковими.

Існують процеси з безперервними станами (*плавний перехід зі стану до стану*). Приклад такого процесу: зміна напруги в освітлювальній мережі.

Друга класифікаційна ознака – характер функціонування системи в часі.

ВП називається **процесом з дискретним часом**, якщо переходи системи зі стану в стан можливі тільки в певні, заздалегідь фіксовані моменти часу:  $t_1, t_2 \dots$ . Якщо перехід системи зі стану на стан можливий у будь-який наперед невідомий випадковий момент часу, то ВП називається **процесом з безперервним часом**.

Розрізняють наступну класифікацію, що застосовується до випадкових марківських процесів [16]:

- Марківські ланцюги;
- Марківські послідовності;
- марківські процеси з кінцевим та нескінченним числом станів;
- дискретно-безперервні Марківські процеси;
- змішані марківські процеси.

Характер реалізацій чотирьох основних видів марківських процесів наведено на рис. 6.2.

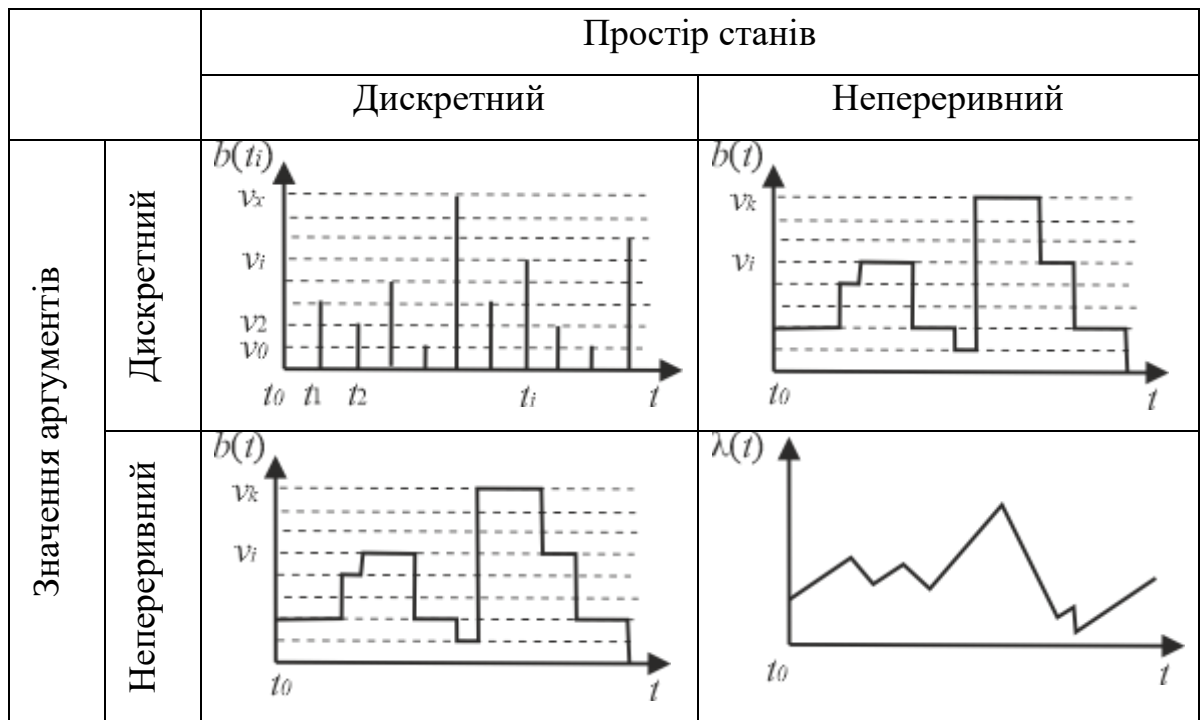


Рис. 6.2. Характер реалізацій основних видів Марківських процесів.

Розглянемо два варіанти опису марківських процесів – з дискретним та безперервним часом.

### 6.2. Випадкові процеси з дискретним часом

Випадковий процес називається процесом із дискретним часом, якщо переходи системи зі стану в стан можливі лише у певні, заздалегідь фіксовані моменти часу. У проміжках між переходами система зберігає свій стан.

В окремій системі  $S$ , яка може перебувати в станах  $S_1, S_2, S_3, \dots, S_n$ , а переходи зі стану в стан можливі в моменти часу  $t_1, t_2, \dots, t_n$ . то ці моменти часу називають «кроками» процесу та розглядають процес у системі як функцію цілочислового аргументу:  $1, 2, \dots, k$ . В цій системі позначають  $S_i(k)$  подію, яка полягає в тому, що після  $k$  кроків система перебуватиме у стані  $S_i$ .

Випадкова послідовність подій називається марківським ланцюгом, якщо для кожного кроку ймовірність переходу системи з будь-якого стану  $S_i$  в будь-який інший  $S_j$  не залежить від того, коли і як вона потрапила в стан  $S_i$ .

Марківський ланцюг описується за допомогою ймовірностей станів системи  $P_i$ . Для кожного  $k$ -го кроку сума ймовірностей  $P_i(k)$  дорівнює  $\sum P_i(k) = 1$ .

Для будь-якого кроку (моменту часу  $t_1, t_2, \dots, t_n$ ) існують ймовірності переходу з одного стану в інший стан. Деякі з цих ймовірностей можуть дорівнювати 0, якщо безпосередній перехід за один крок неможливий. Ці ймовірності називаються перехідними ймовірностями марківського ланцюга.  $P_{ij}$  - це ймовірність переходу системи зі стану  $S_i$  у стан  $S_j$ , а  $P_{ii}$  – це ймовірність затримки системи у  $i$ -му стані.

**Марківським ланцюгом** називається марківський випадковий процес з дискретним станом та дискретним часом.

Випадковий процес у разі характеризується послідовністю станів  $S(0), S(1), S(2), \dots, S(k), \dots, (*)$ , якщо  $S(0)$  - початковий стан системи (перед першим);  $S(1)$  - стан системи безпосередньо після першого кроку;  $\dots, S(k)$  - стан системи безпосередньо після  $k$ -го кроку.

В цьому випадку перехід в новий стан система здійснює на кожному кроці, а нас буде цікавити лише послідовність яку проходить ВП у своєму розвитку, і не цікавить, час в який конкретно відбувався кожен із переходів.

Модель марківського процесу з дискретним станом та дискретним часом зручно представляти у вигляді графа рис. 6.3. При цьому - стани системи (вершини) пов'язані між собою зв'язками (переходами з  $i$ -го стану  $j$ -тий стан).

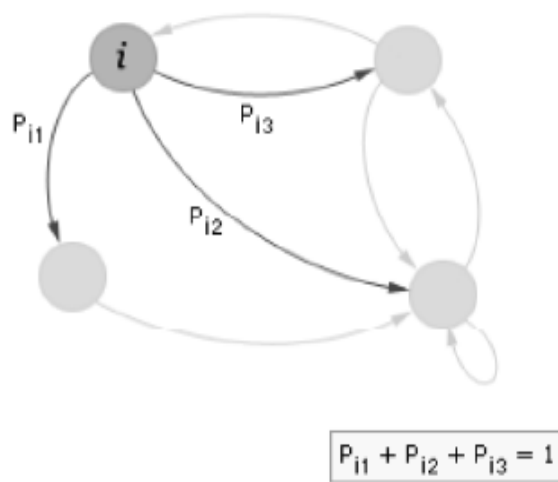


Рис. 6.3. Приклад графа станів системи з певними переходами.

У стрілок проставляються відповідні можливості переходів. Такий граф, що складається, називають **розміченим**.

Марківський процес з дискретними станами отримав назву **марківського ланцюга**.

Марківський ланцюг називається **однорідним**, якщо перехідні ймовірності не залежать від номера кроку. Інакше марківський ланцюг називається **неоднорідним**.

**Однорідний марківський ланцюг.** Припустимо що є система  $S$  яка має  $n$  станів:  $S_1, S_2, S_3, \dots, S_n$ . Для кожного стану нам відома ймовірність переходу в будь-який інший стан за один крок. Записуємо перехідні ймовірності у вигляді прямокутної матриці перехідних ймовірностей або стохастичної матриці.

$$\| P_{ij} \| = \begin{vmatrix} P_{11} & P_{12} & \dots & P_{1j} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2j} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{i1} & P_{i2} & \dots & P_{ij} & \dots & P_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nj} & \dots & P_{nn} \end{vmatrix}$$

Деякі з перехідних ймовірностей  $P_{ij}$  можуть дорівнювати 0, якщо перехід зі стану  $S_i$  в стан  $S_j$  неможливий за один крок. По головній діагоналі стоять ймовірності  $P_{ii}$  тобто ймовірність того, що система затримається в даному стані.

*Сума всіх членів матриці, що стоять у кожному рядку, дорівнює 1.*

Ймовірність  $P_{ij}$  можна записати як умовну ймовірність того, що система  $S$  знаходиться на  $k-1$  кроці в стані  $S_i$ , на  $k$ -тому кроці перейде в  $S_j$ .

$$P_{ij} = P(S_j(k) / S_i(k-1)) \quad (6.1)$$

Рядки матриці перехідних ймовірностей називаються ймовірнісними векторами. Причому кожен ланцюг Маркова визначає стохастичну матрицю та навпаки, кожній стохастичній матриці можна поставити у відповідність ланцюг Маркова. Для цього достатньо, як стани взяти рядки матриці, а її елементи вважати перехідними ймовірностями.

З кожної стохастичної матриці можна побудувати відповідний розмічений (зважений) орієнтований граф станів системи. Як вершини цього графа беруться рядки матриці та проводяться між ними дуги, за умовою якщо  $P_{ij} > 0$ . Як ваги дуг беруться значення  $P_{ij}$ .

Маючи у своєму розпорядженні розмічений граф станів системи, матрицю перехідних ймовірностей, знаючи початковий стан системи, можна знайти ймовірності стану системи після будь-якого  $k$ -го кроку.

$$P_{ij}(k) = \sum P_j(k-1) \times P_{ij}(k) - \text{рекурентна формула.} \quad (6.2)$$

**Неоднорідний марківський ланцюг.** В цьому випадку перехідні можливості змінюються від кроку до кроку – кожному кроку відповідає своя матриця перехідних ймовірностей. Ймовірність переходу системи зі стану  $S_i$ , до стану  $S_j$  на  $k$ -тому кроці дорівнює умовної ймовірності  $P_{ij}(k)$  відповідно, а система на  $k-1$  кроці була в стані  $S_i$ , а на  $k$ -тому кроці перейшла в стан  $S_j$ .

$$P_{ij} = P(S_j(k) / S_i(k-1)),$$

$$P_{ij}(k) = \sum P_j(k-1) \times P_{ij}(k) - \text{рекурентна формула.}$$

**Приклад.** Для марківських випадкових процесів із дискретним часом можливі три способи задання: граф станів, дерево логічних можливостей та матричний спосіб представлення.

$$P = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$

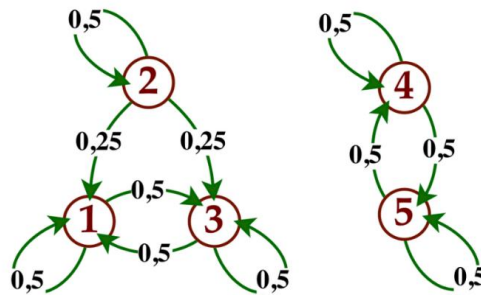


Рис 6.4. Граф стану системи з 5 вершинами

Граф з п'ятьма станами системи, у яких ймовірність переходу в інший обумовлена лише поточним станом представлено на рисунку 6.4. Та нагадаємо, що Марківським називається випадковий процес, стан якого в черговий момент часу  $t + \sigma$  залежить тільки від поточного стану в момент часу  $t$ .

### 6.3. Марківські процеси з дискретним станом та безперервним часом

Насправді значно частіше зустрічаються ситуації, коли переходи системи зі стану в стан відбувається у випадкові моменти часу, які заздалегідь вказати неможливо.

Для опису таких процесів може бути з успіхом застосована схема марківського ВП з дискретними станами та безперервним часом – **безперервний ланцюг Маркова**.

Для такого процесу ймовірність переходу зі стану  $S_i$  в стан  $S_j$ , для будь-якого фіксованого моменту часу дорівнює нулю, так само як ймовірність будь-якого окремого значення неперервної стохастичної величини.

Замість ймовірності переходу  $P_{ij}$  розглядається щільність ймовірності переходу  $\lambda_{ij}$ . З точністю до нескінченно малих великих порядків можна показати, що:

$$P_{ij}(\Delta t) = \lambda_{ij} \cdot \Delta t;$$

При розгляді безперервного ланцюга Маркова зручно уявляти собі переходи системи зі стану в стан, який відбуваються під впливом деяких подій.

У цьому випадку щільності ймовірностей переходу набувають сенсу інтенсивностей  $\lambda_{ij}$  відповідних потоків подій (як тільки відбувається перша подія в потоці з інтенсивністю  $\lambda_{ij}$ , система зі стану  $S_i$  стрибком переходить у стан  $S_j$ ). Якщо всі ці потоки пуасонівські, то процес, що протікає в системі, буде марківським.

Таким чином, для безперервних ланцюгів Маркова замість перехідних ймовірностей стрілок указуються щільність ймовірностей переходів  $\lambda_{ij}$  (рис. 6.5).

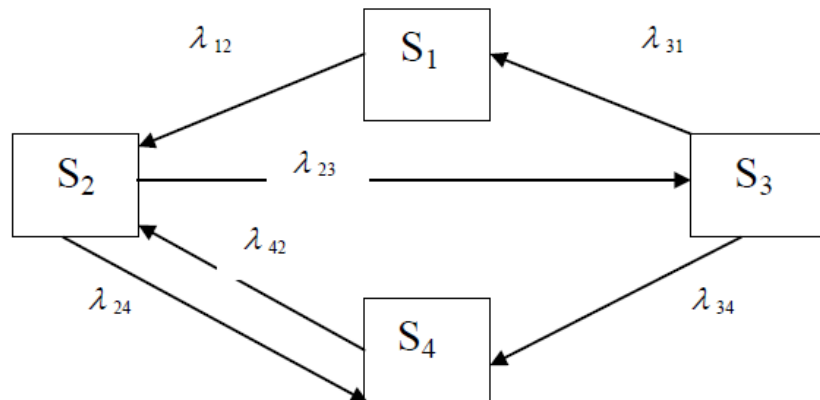


Рис. 6.5 Граф станів системи

Якщо щільність ймовірностей переходів  $\lambda_{ij}$  залежить від часу  $t$  (мінімальні значення  $\Delta t$ ), такий марківський процес називається однорідним.

Розглянемо випадок, коли система  $S$  має кінцеву кількість станів  $S_1, S_2, \dots, S_n$ . Для опису випадкового процесу, що протікає в цій системі, застосовуються імовірності станів:

$$p_1(t), p_2(t), \dots, p_n(t), (*)$$

де  $p_i(t)$  - ймовірність того, що система  $S$  в момент часу  $t$  знаходиться в стані  $S_i$

Очевидно, що для будь-якого часу  $t$ :  $\sum_{i=1}^n p_i(t) = 1$

Для знаходження ймовірностей (\*) необхідно вирішити систему диференціальних рівнянь (рівнянь Колмогорова), що мають вигляд:

$$\frac{dP_i(t)}{dt} = \sum_{j=1}^n \lambda_{ji}(t)P_j(t) - \sum_{j=1}^n \lambda_{ij}(t)P_i(t); \quad (i, j = 1, 2, \dots, n)$$

Величина  $\lambda_{ij} p_i(t)$  називається потоком ймовірності переходу зі стану  $S_i$  стан  $S_j$ .

Рівняння Колмогорова складають за розміченим графом станів системи, користуючись наступним правилом:

1) швидкість зміни ймовірності (похідна) кожного стану дорівнює алгебраїчній сумі творів інтенсивності потоку, що переводить систему за цим напрямом, на ймовірність того стану, звідки здійснюється перехід.

2) мнемонічне правило:

– знак «+», якщо стрілка в даний стан:

– знак "-". якщо стрілка з цього стану.

Це правило складання диференціальних рівнянь для ймовірностей станів є загальним і справедливим для будь-якого безперервного марківського ланцюга. З його допомогою можна абсолютно механічно, без будь-яких міркувань, записувати диференціальні рівняння для ймовірностей станів безпосередньо за розміченим графом станів.

Для графа станів рис. 6.5, рівняння мають вигляд:

$$\begin{cases} \frac{dp_1(t)}{dt} = -\lambda_{12}p_1(t) + \lambda_{31}p_3(t); \\ \frac{dp_2(t)}{dt} = -\lambda_{23}p_2(t) - \lambda_{24}p_2(t) + \lambda_{12}p_1(t) + \lambda_{42}p_4(t); \\ \frac{dp_3(t)}{dt} = -\lambda_{31}p_3(t) - \lambda_{34}p_3(t) + \lambda_{23}p_2(t); \\ \frac{dp_4(t)}{dt} = -\lambda_{42}p_4(t) + \lambda_{24}p_4(t) + \lambda_{34}p_3(t). \end{cases}$$

Для вирішення системи рівнянь Колмогорова необхідно встановити початковий розподіл ймовірностей  $P_0(0), P_1(0), \dots, P_n(0)$ .

Як правило, за винятком особливо простих систем, рішення можна отримати лише чисельними методами.

Інтегрування цих рівнянь при відомому початковому стані системи дасть ймовірності станів як функції часу.

Виникають слушні запитання:

Як поводитиметься система при  $t \rightarrow \infty$ ?

Чи існують межі функцій  $P_1(t), P_2(t), \dots, P_i(t), \dots, P_n(t)$ ?

Відповідно – якщо межі функцій існують, то відповідні ймовірності станів називаються граничними ймовірностями станів (або «фінальними», тобто кінцевими). Якщо граничні ймовірності існують, то в цьому стані має місце режим, для якого похідні дорівнюють нулю.

Система, для якої існують фінальні ймовірності, називається ергодичною, а відповідний випадковий процес - ергодичним. В цьому випадку система диференціальних рівнянь Колмогорова перетворюється на систему алгебраїчних рівнянь.

Спільно з умовною умовою  $\sum_{i=1}^n p_i(t) = 1$  ці рівняння дають можливість обчислити всі граничні ймовірності станів  $P_1, P_2, \dots, P_i, \dots, P_n$ .

У цьому граничному режимі кожна фінальна можливість може бути витлумачена як середнє відносне часу перебування системи у цьому стані.

## ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ

1. Основні поняття марківських процесів.
2. Дайте формальне визначення марківського процесу. Що означає властивість «без пам'яті».
3. Сформулюйте рівняння Галлея - Колмогорова для ланцюга Маркова дискретного часу і поясніть його смисл.
4. Марківські процеси з дискретним станом та безперервним часом.
5. Що таке інформаційний критерій ефективності системи (наприклад, ентропія станового розподілу)? Як зміни стаціонарного розподілу впливають на цей критерій і що це значить для оцінки «ефективності» системи?

### Тема 7. Моделювання систем. Моделювання роботи програмного забезпечення.

#### 7.1. Основи моделювання систем.

**Моделювання систем** - це процес створення абстракцій реальних або уявних систем, які дозволяють аналізувати їх поведінку та характеристики без необхідності втручання в саму реальну систему. Моделі є спрощеннями реальних об'єктів і процесів, що дає змогу розробникам, аналітикам та дослідникам прогнозувати, тестувати й оптимізувати поведінку систем [17].

**Статичне моделювання** - це тип моделювання, який використовується для аналізу системи в умовах сталості, без врахування змін у часі. Статичне моделювання зосереджується на властивостях системи в певний момент часу або в рамках фіксованої конфігурації системи.

**Динамічне моделювання** - це моделювання, яке вивчає поведінку системи в часі, враховуючи зміни станів, взаємодії елементів системи та зворотні зв'язки в системі. Динамічне моделювання дозволяє прогнозувати, як система буде реагувати на зміни вхідних параметрів, вплив елементів системи один на одного в часі. Як приклад можна навести – моделювання процесів управління проєктами, економічних моделей, прогнозування поведінки систем в умовах змінної завантаженості.

*Види моделей.* Розрізняють фізичні та абстрактні моделі.

**Фізичні моделі** утворюються із сукупності матеріальних об'єктів. Для їх побудови використовуються різні фізичні властивості об'єктів, причому природа застосовуваних моделі матеріальних елементів не обов'язково співпадає з використанням у досліджуваному об'єкті. Як приклад фізичної моделі наведемо макетування.

**Інформаційна (абстрактна) модель** – це опис об'єкта досліджень певною мовою. Абстрактність моделі проявляється у тому, що її компонентами є поняття, а не фізичні елементи (наприклад, словесні описи, креслення, схеми, графіки, таблиці, алгоритми чи програми, математичні

описи). Інформаційна модель - це цілеспрямовано відібрана інформація про об'єкт, яка представлена найбільш суттєвими для дослідника властивостями цього об'єкта.

Серед інформаційних (абстрактних) моделей розрізняють: дескриптивні, наочні та змішані, гносеологічні, інфологічні, кібернетичні, сенсуальні (чуттєві), концептуальні, математичні.

*Концептуальна модель* - це абстрактна модель, яка виявляє причинно-наслідкові зв'язки та властивості притаманні досліджуваному об'єкту в рамках відповідного (певного) дослідження. Основне призначення концептуальної моделі – виявлення набору причинно-наслідкових зв'язків, облік яких необхідний для отримання необхідних результатів. Один і той самий об'єкт може бути представлений різними концептуальними моделями, які будуються в залежності від мети дослідження. Так, одна концептуальна модель може відображати часові аспекти функціонування системи, інша – вплив відмов на працездатність самої системи або частини.

*Математична модель* – абстрактна модель яка представлена мовою математичних відносин. Вона має форму відповідних функціональних залежностей між параметрами системи, що враховуються відповідною концептуальною моделлю. Ці залежності конкретизують причинно-наслідкові зв'язки, виявлені в концептуальній моделі, та характеризують їх кількісно.

*Дискретне моделювання* (Discrete Event Simulation, DES). Цей метод підходить для аналізу систем, у яких зміни відбуваються на певних етапах або у відповідний момент часу, а не безперервно. Ключовим аспектом є визначення подій, які змінюють стан системи, та їх послідовність.

*Континуальне моделювання* (Continuous Simulation). Континуальне моделювання використовується для систем, в яких зміни відбуваються безперервно, і змінні параметри системи (наприклад, температура, швидкість або обсяг ресурсу) постійно змінюються в часі. У таких моделях враховуються рівняння, які описують динаміку процесів.

*Агентне моделювання* (Agent-Based Modeling, ABM). Агентне моделювання фокусується на створенні моделей, що складаються з агентів, кожен з яких має власні характеристики, поведінку та здатність взаємодіяти з іншими агентами та зовнішнім середовищем. Це дозволяє створювати складні моделі, де агенти можуть адаптуватися до змінних умов.

Таким чином, модель - це спеціальний об'єкт, який в деяких відносинах заміщає оригінал. Принципово немає моделі, яка була б повним еквівалентом оригіналу. Будь-яка модель відображає лише деякі сторони оригінального об'єкту аналізу. Тому, для одержання повних даних про об'єкту аналізу потрібно використовувати сукупність моделей. Складність моделювання, як процесу, полягає у відповідному виборі такої сукупності моделей, які в повній мірі описують реальний пристрій або об'єкт у потрібних співвідношеннях.

## 7.2. Класифікація моделей у системному аналізі

Вибір методів моделювання є важливим та складним завданням. Від грамотного вибору залежить складність моделі її адекватність, трудовитрати, пов'язані з її реалізацією. Для вибору моделей потрібно формувати її класифікацію. Класифікацією, у загальному розумінні, називають розбиття об'єктів на класи за найістотнішими ознаками [18].

Мета класифікації: обмежити вибір підходів до відображення системи; застосувати до певних класів відповідні методи системного аналізу; сформувані загальні рекомендації щодо вибору методів для відповідного класу систем.

Зв'язок між об'єктом та представленою моделлю має опосередкований характер, оскільки між ними розташовується системний опис об'єкта. Моделювання завжди передбачає прийняття припущень того чи іншого ступеня важливості, тому вже на етапі формування системного опису можуть бути відсічені несуттєві властивості об'єкта. У будь-якій задачі моделювання існує певний поріг складності, переступити який можна лише ціною відмови від вимог точності відповідних рішень.

Наприклад, при комп'ютерній реалізації розв'язання багато-критеріальних задач, точні методи їх вирішення можуть виявитись економічно не вигідними порівняно з більш простими наближеними методами. Оцінка та забезпечення якості моделі може бути виконана на вже побудованих моделях систем. Тому, якість моделі має бути «закладено» у ній ще на етапі вибору типу моделі та базуватися на досвіді дослідника. Для структурованої акумуляції такого досвіду дослідник має бути підготовлений до оцінки параметрів (показників) якості, якими він керуватиметься.

Однією з найважливіших ознак для класифікації моделей систем є класифікація за методами моделювання, покликана забезпечити оптимальний вибір типу моделі, що збігається зі специфікою об'єкта (системи), що досліджується, умовами поведінки системи, завданнями моделювання.

Серед відомих методів особливо вирізняються методи математичного моделювання. На рисунку 7.1 наведено укрупнені розділи класифікації моделей за ознакою «*методи моделювання*». Однак формалізоване відображення системи за допомогою абстрактної мови не зводиться лише до детермінованих та ймовірнісних моделей.

До класифікації включено методи спрямовані на активізацію інтуїції, які активно досліджувалися у початковий період розвитку теорії систем: «мозкова атака» (або колективна генерація ідей) та інші методи вироблення колективних рішень; методи типу «сценаріїв», методи та методики структуризації, методи експертних оцінок, морфологічного моделювання тощо. буд. До цього класу моделей можна віднести моделі вироблення колективних рішень (наприклад, у формі сценаріїв, які можна вважати словесними чи вербальними моделями), моделі структуризації, організації складних експертиз.

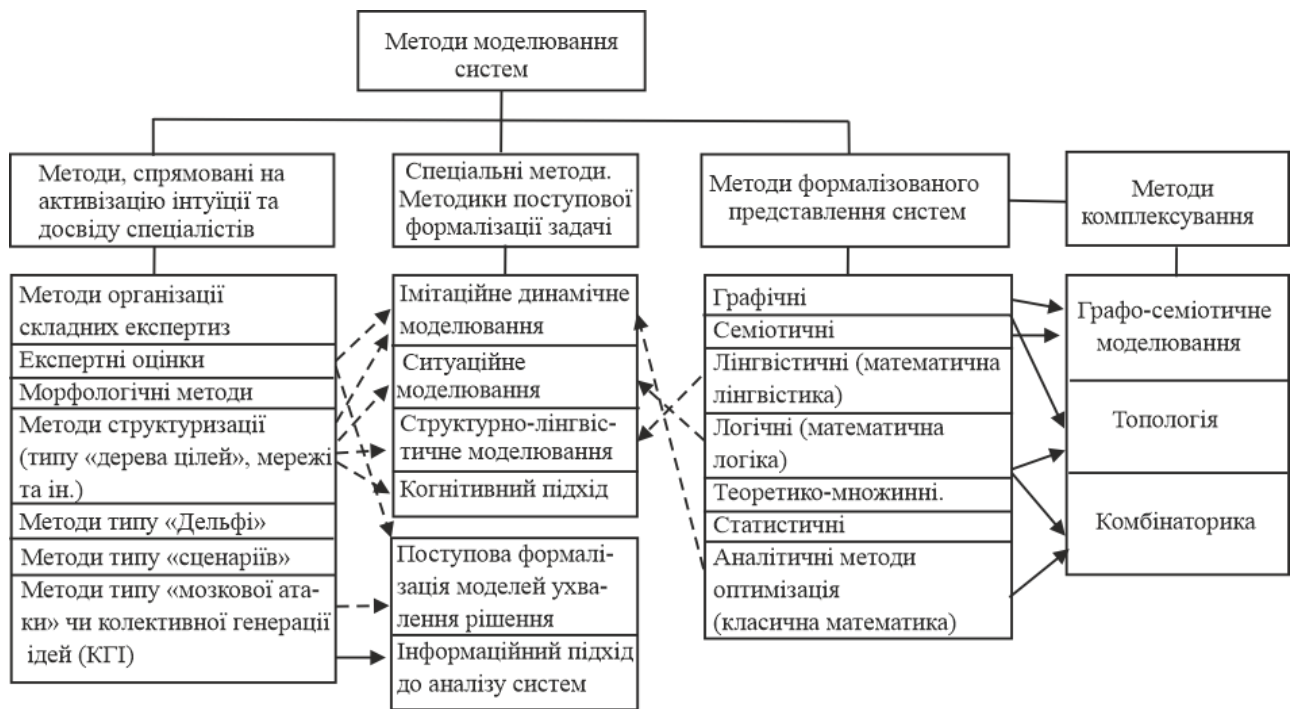


Рис 7.1. Методи моделювання систем

Самостійними розділами теорії моделювання можна також вважати спеціальні методи або підходи – ситуаційне, структурно-лінгвістичне, когнітивне, логіко-лінгвістичне, лінгво-комбінаторне моделювання, а також моделювання, засноване на інформаційному підході до аналізу систем, що забезпечує перехід від менш формалізованої моделі до більш формалізованої.

Найбільшого поширення набули такі спеціальні методи моделювання систем:

*Імітаційне динамічне моделювання* (System Dynamics Simulation Modelling). Запропоновано Дж. Форрестером (США) у 1950-х рр.

*Ситуаційне моделювання*. Ідею запропоновано Д.А. Поспеловим та реалізована Ю.І. Кликовим та Л.С. Загадський (Болотової).

*Лінгво-комбінаторне моделювання*. Запропоновано для моделювання погано формалізованих систем М.Б. Ігнат'євим.

*Логіко-лінгвістичний моделювання*. Є розвитком структурно-лінгвістичного моделювання, широко поширеного в 1970-і рр.. в інженерній практиці і заснованого на використанні для реалізації ідей комбінаторики структурних уявлень різного роду, з одного боку, і засобів математичної лінгвістики, з іншого. системами розвиває Б.Л. Кукор, використовуючи як мовні (лінгвістичні) засоби та інші методи дискретної математики (теоретико-множинні уявлення, засоби математичної логіки та семіотики).

*Теорія інформаційного поля та інформаційних ланцюгів* (Інформаційний підхід до моделювання та аналізу систем). Підхід, що базується на ідеї поступової формалізації завдань (проблемних ситуацій) з невизначеністю шляхом послідовного використання спеціальних методів.

*Системно-структурний синтез.* Системно-структурні методи моделювання розроблялися з самого початку розвитку теорії систем на основі ієрархічних і мережевих структур як засоби дослідження об'єктів і процесів з невизначеністю, коли не можуть бути одразу отримані математичні моделі.

*Когнітивний підхід* (від лат. *cognitio* - знання, пізнання). Базується на ідеях когнітивної психології.

До *математичних* можна віднести *теоретико-множинні моделі, моделі математичної логіки, теорії графів*. Якщо основою класифікації покласти не математичні методи, а методи формалізованого уявлення систем, то до цього класу також відносяться моделі математичної логіки (лінгвістичні) та семіотичні моделі.

У пошуках терміну для виділеного класу моделей, об'єктів та процесів, які не можуть бути одразу відображені аналітичними та іншими формалізованими моделями, пропонувалося віднести класифіковані моделі за ступенем абстрагування, виділяючи поряд із формальними та формалізованими моделями словесні. Проте словесні, точніше – вербальні (у термінах предметної області) відображення завдання моделювання зазвичай вважається постановкою завдання, а не представленням моделі, а етапи процесу моделювання розглядаються як процес перекладу вербального опису в формальне (чи формалізоване).

Представимо узагальнені групи (класи) методів:

– аналітичні методи класичної математики, включаючи інтегро-диференціальне обчислення, методи пошуку екстремумів функцій, варіаційне обчислення; методи математичного програмування – лінійного, нелінійного, цілочисленого, динамічного, що широко застосовуються в теорії оптимізації; перші роботи з теорії ігор;

– статистичні (теоретичні розділи математики – теорія ймовірностей, математична статистика, та напрямки прикладної математики, що використовують стохастичні уявлення – теорію масового обслуговування, методи статистичних випробувань (засновані на методі Монте-Карло), методи висування та перевірки статистичних гіпотез А. Вальда та інші;

– теоретико-множинні, логічні, лінгвістичні, семіотичні подання (методи дискретної математики), що становлять теоретичну основу розробки мов моделювання, автоматизації проектування, інформаційно-пошукових мов;

– графічні (теорія графів та різного роду графічні уявлення інформації – типу діаграм, гістограм та інших графіків).

Також зазначимо що на перетині аналітичних та теоретико-множинних уявлень виникла алгебра груп; паралельно, в рамках алгебри груп та теорії множин почала розвиватися комбінаторика; теоретико-множинні та графічні уявлення стали основою виникнення топології; статистичні та теоретико-множинні методи ініціювали виникнення теорії нечітких множин Л. Заде,

яка, у свою чергу, стала початком розвитку нового напрямку - нечітких формалізації.

Практично неможливо створити єдину класифікацію, куди входять всі розділи сучасної математики яка застосовується в системному аналізі. У той самий час наведені напрями допомагають зрозуміти особливості конкретних методів, використовують відповідні засоби, напрями чи їх поєднання, допомагають вибирати методи для конкретних об'єктів аналізу..

Нові методи часто виникають на основі поєднання існуючих раніше. Так, методи, які іноді називають комплексованими (комбінаторика, топологія), починали розвиватися паралельно в рамках лінійної алгебри, теорії множин, теорії графів, а потім оформилися в самостійні напрямки.

### 7.3. Аналітичні і статистичні методи

Ці групи методів отримали найбільше поширення в практиці проектування та управління сучасних систем. Але, для представлення проміжних та остаточних результатів моделювання широко використовуються графічне представлення систем (графіки, діаграми).

Аналітичними, у аналізованій класифікації, названі методи, які відображають реальні об'єкти та процеси у вигляді набору точок (безрозмірні при суворих математичних доказах), які здійснювали будь-які переміщення у просторі або взаємодіють між собою.

Основу понятійного (термінологічного) апарату цих уявлень складають поняття класичної математики (величина, формула, функція, рівняння, система рівнянь, логарифм, диференціал, інтеграл тощо).

На базі аналітичних уявлень виникли і розвиваються математичні теорії різної складності - від апарату класичного математичного аналізу (методів дослідження функцій, їх виду, способів подання, пошуку екстремумів функцій) до нових розділів сучасної математики, такі як математичне програмування (лінійне, нелінійне, динамічне), диференціальні ігри.

Ці теоретичні напрями стали основою багатьох прикладних застосувань в теорії автоматичного управління, теорії оптимальних рішень тощо. При моделюванні систем застосовується широкий спектр символічних уявлень, що використовують мову класичної математики. Однак, далеко не завжди ці символічні уявлення адекватно відображають реальні складні процеси, їх не можна вважати строгими математичними моделями.

Більшість із напрямів математики не містять засобів постановки завдання та доказу адекватності моделі. Остання доводиться експериментом, який у міру ускладнення проблем стає все більш складним, дорогим, який завжди безперечний та має відповідну реалізацію.

До складу цього класу методів входить відносно новий напрямок математики – математичне програмування, яке містить засоби постановки задачі та розширює можливості доказу адекватності моделей.

Статистичні уявлення сформувалися як самостійний науковий напрямок у середині минулого століття (хоча виникли значно раніше). Основу їх становить відображення явищ та процесів за допомогою випадкових (стохастичних) подій та їх поведінок, які описуються відповідними ймовірнісними (статистичними) характеристиками та статистичними закономірностями.

*Графічні методи.* Поняття графа спочатку було запроваджено Л.А. Ейлер. Графічні уявлення дозволяють наочно відображати структури складних систем і процесів. З цієї точки зору вони могли розглядатися як проміжний метод між методами формалізованого уявлення систем та методами активізації фахівців. Справді, такі засоби, як графіки, діаграми, гістограми, деревоподібні структури можна віднести до засобів активізації інтуїції фахівців.

Розглянуті нижче підходи і методи виникали і розвивалися як самостійні, і для узагальнення в теорії систем спочатку їх називали якісними (застерегаючи умовність цієї назви, оскільки при обробці одержуваних результатів можуть використовуватися і кількісні уявлення) або експертними оскільки вони являють собою підходи в тій чи іншій формі, що активізують виявлення та зменшення думок досвідчених фахівців - експертів (у широкому сенсі термін «експерт» у перекладі з латинського означає «дослідний»).

Однак є особливий клас методів, пов'язаних з безпосереднім опитуванням експертів, який називають методом експертних оцінок, тому був запропонований [19] узагальнюючий термін, винесений в назву параграфа. Цей термін, хоч і дещо громіздкий, більшою мірою, ніж інші, відображає сутність методів, до яких вдаються фахівці в тих випадках, коли не має можливості відразу описати аналізовану проблемну ситуацію аналітичними залежностями або вибрати той чи інший метод формалізованого подання для формування моделі прийняття відповідних рішень.

*Методи типу «мозковий штурм»* або колективної генерації ідей. Концепція мозкової атаки або мозкового штурму набула широкого поширення з початку 1950-х років. як «метод систематичного тренування творчого мислення», спрямований на «відкриття нових ідей та досягнення згоди групи людей на основі інтуїтивного мислення». Мозкова атака ґрунтується на гіпотезі, що серед великої кількості ідей є щонайменше кілька добрих, корисних для вирішення проблеми, які потрібно виявити. Методи цього відомі також можна зустріти під назвою: колективної генерації ідей, конференцій ідей, метод обміну думками.

Залежно від прийнятих правил і жорсткості їхнього виконання розрізняють *пряму мозкову атаку, метод обміну думками, методи за типами комісій, судів* (в останньому випадку створюються дві групи: одна група вносить якнайбільше пропозицій, а друга намагається максимально їх розкритикувати). Мозкову атаку можна проводити у різних форматах – форма ділової гри із застосуванням тренувальної методики «стимулювання

спостереження». Відповідно до цієї гри група формує уявлення про проблемну ситуацію, а експерту пропонується знайти найбільш логічні способи вирішення проблеми.

*Методи типу «сценарії».* Методи підготовки та узгодження уявлень про проблему або аналізований об'єкт, викладених письмово, отримали назву сценаріїв. Спочатку цей метод передбачав підготовку тексту, що містить логічну послідовність подій або можливі варіанти вирішення проблеми, розгорнуті у часі. Однак пізніше обов'язкову вимогу тимчасових координат було знято, і сценарієм став називатися будь-який документ, що містить аналіз проблеми і пропозиції щодо її вирішення або розвитку системи, незалежно від того, в якій формі він представлений. Як правило, на практиці пропозиції для підготовки подібних документів пишуться експертами спочатку індивідуально, а потім формується узгоджений текст.

Сценарій передбачає не лише змістовні міркування, що допомагають не упустити деталі, які неможливо врахувати у формальній моделі (у цьому, власне, і полягає основна роль сценарію), а й містить результати кількісного техніко-економічного чи статистичного аналізу з попередніми висновками. Група експертів, яка готує сценарій, зазвичай користується правом отримання необхідних відомостей від підприємств та організацій, необхідних консультацій.

І тоді визначаються передові ролі фахівців із системного аналізу при підготовці певних сценаріїв – допомога залученим провідним спеціалістам відповідних галузей знань виявити загальні закономірності розвитку системи: проаналізувати зовнішні та внутрішні чинники, що впливають на її розвиток та чітко формулювання цілей: провести аналіз висловлювань провідних фахівців у періодичній пресі, наукових публікаціях; створити допоміжні інформаційні фонди, які б вирішенню відповідної проблеми.

*Методи структуризації.* Структурні уявлення різного роду дозволяють розділити складну проблему з великою невизначеністю на більш дрібні, які більш ретельно та краще піддаються дослідженню, що саме собою можна вже представити як окремий метод дослідження іменованій іноді системно-структурним. І тут ми вже отримуємо види структур, одержувані шляхом розчленування системи у часі (мережеві структури) чи просторі (ієрархічні структури різного роду, матричні структури). Зазначимо що методи структуризації є основою будь-якої методики системного аналізу, будь-якого складного алгоритму організації, проектування, ухвалення відповідних управлінського рішення.

*Методи типу «дерева цілей».* Ідея методу дерева цілей вперше була запропонована У. Черчмен у зв'язку з проблемами прийняття рішень у промисловості. Термін «дерево» передбачає використання ієрархічної структури, одержуваної шляхом розчленування загальної мети на підцілі, які, в свою чергу, розкладаються, на детальніші складові, які в мають назву – підцілі нижчих рівнів, напрямків, проблем, а починаючи вони з певного рівня – аналізу функцій.

При використанні методу "дерева цілей", як засіб прийняття рішень, часто застосовують термін "дерево рішень". При застосуванні методу для виявлення та уточнення функцій системи управління говорять про «дерево цілей та функцій». При структуризації тематики науково-дослідної організації користуються терміном «дерево проблеми», а розробки прогнозів – «дерево напрямів розвитку (прогнозування розвитку)» чи «прогнозний граф».

*Методи експертних оцінок.* Вивченню особливостей та можливостей застосування експертних оцінок присвячено багато науково методичних робіт. Основні питання які в них розглядаються:

- проблеми формування експертних груп, включаючи вимоги до експертів, розміри групи, питання тренування експертів, оцінки їхньої компетентності;
- форми експертного опитування (різного роду анкетування, інтерв'ю, змішані форми опитування) та методики організації опитування (у тому числі методики анкетування, мозкова атака, ділові ігри тощо);
- підходи до оцінювання (ранжування, нормування, різні види впорядкування, у тому числі методи переваг. попарних порівнянь та ін.);
- методи обробки експертних оцінок;
- способи визначення узгодженості думок експертів, достовірності експертних оцінок (у тому числі статистичні методи оцінки дисперсії, оцінки ймовірності для заданого діапазону змін оцінок, оцінки рангової кореляції Кендалла Спірмена, коефіцієнта конкордатні та ін.), методи підвищення узгодженості оцінок шляхом відповідних способів.

*Методи типу «дельфі».* Метод «Дельфі» або метод «дельфійського оракула» спочатку був запропонований О. Хелмером і його колегами як ітеративна процедура при проведенні мозкової атаки, яка сприяла б зниженню впливу психологічних факторів при проведенні засідань і підвищенню об'єктивності результатів при об'єктивних результатах порівняльного аналізу у складових «дерев цілей» при розробці «сценаріїв».

У конкретних методиках, що реалізують процедуру «Дельфі», ця ідея використовується різною мірою. Так, у спрощеному вигляді організується послідовність ітеративних циклів мозкової атаки. У складнішому варіанті розробляється програма послідовних індивідуальних опитувань з використанням методів анкетування, що виключають контакти між експертами, але передбачають ознайомлення їх з думками між опитуваннями.

*Методи організації складних експертиз.* Розглянуті вище недоліки експертних оцінок призвели до необхідності створення методів, що підвищують об'єктивність отримання оцінок шляхом розчленування великої первісної невизначеності проблеми, запропонованої експерту для оцінки, на менші, дрібні для кращого аналізу та осмислення.

## 7.4. Моделі систем

Найбільш просте визначення системи зазначає – це засіб досягнення мети, а мета - це суб'єктивний абстрактний образ ще не існуючого, але бажаного стану середовища або навколишнього світу.

Прикладом системи, що визначає час у будь-який момент, є годинник, або міський транспорт, мета якого забезпечити пересування людей у масштабах міста.

Аналізуючи ці приклади, легко показати, що цілі систем можуть бути неоднозначними, що ту саму систему можна використовувати для різних цілей в різні проміжки часу.

Дати опис та сформулювати мету навіть існуючих систем дуже складно. Для полегшення розв'язання задачі створення систем необхідно застосування сучасних методів моделювання, тому що для визначень та точної характеристики будь-якої системи слід удосконалювати та розвивати її модель.

*Модель чорного ящика.* Найпростішою моделлю предствалення та опису системи є модель «чорного ящика» (рис. 7.2). Ідея використання «чорної скриньки» виникла від недостатності внутрішньої будови (складу) самої системи, тому її зображаємо у вигляді непрозорого чорного ящика, який має певні властивості: цілісність та відокремленість від середовища.

Перше властивість свідчить про те що ящик відокремлений, тобто виділений із середовища, але не є повністю ізольованим, а пов'язаний із зовнішнім середовищем зв'язками. Відповідно -він має вплив на середовище та зовнішнє середовище має вплив на систему.

Поруч із очевидністю простоти моделі «чорного ящика», яка (простота) дуже оманлива стоїть складність реальної системи. При описі будь-якої реальної системи часто ми стикаємося з труднощами у визначенні всіх входів та виходів системи яка описується чи аналізується. Використання моделі «чорного ящика» дає результати, які визначаються основною метою системи. Вибір входів «чорної скриньки» є протилежним завданням.

Неврахування деяких вхідних або вихідних параметрів системи часто призводить до поганого досягнення поставленої мети, а систему треба аналізувати спочатку та проводити додатковий аналіз компонентів.

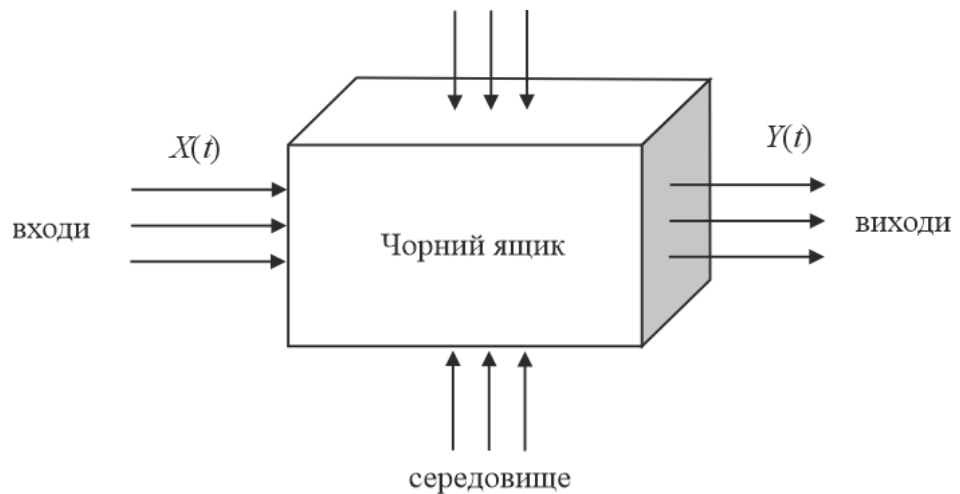


Рис. 7.2. Схема моделі чорного ящика

*Модель складу системи.* Модель чорного ящика не розглядає внутрішній склад системи. Для розвитку моделювання і деталізації опису складу системи потрібно виконати створення моделі складу системи (рис. 7.3).

Ця модель визначає та описує основні складові системи, розглядає елементи системи як неподільні частини та підсистеми, тобто модель складу системи ілюструє ієрархію складових елементів системи.

На перший погляд ця модель здається простою, але якщо дати завдання експертам визначити склад однієї системи, то у кожного експерта буде своя модель, що відрізняється від інших моделей описаних раніше або представлених іншим.

З чого це виходить?

1. Поняття елемента у всіх буде різним.
2. Оскільки модель складу системи є дійовою, то для різних цілей потрібні різні складові частини системи що розглядається.
3. Поділ всієї системи на підсистеми є відносним чи умовним, оскільки межі цих підсистем, кількість цих підсистем та його склад будуть різними, отже й моделі будуть представлені різними.

*Модель структури системи.* Досягнення практичних цілей для моделей «чорного ящика» чи моделі складу системи недостатньо, потрібно ще правильно встановити зв'язки між елементами системи, а найголовніше визначити чи є відносини між елементами та їх показники.

Сукупність необхідних та достатніх для досягнення цілей відносин між елементами називається *структурою системи*.

Нескінченність природи будь-якої системи породжує велику кількість відповідних відносин. Проте при аналізі чи побудові системи ми розглядаємо відповідну сукупність важливих відносин.

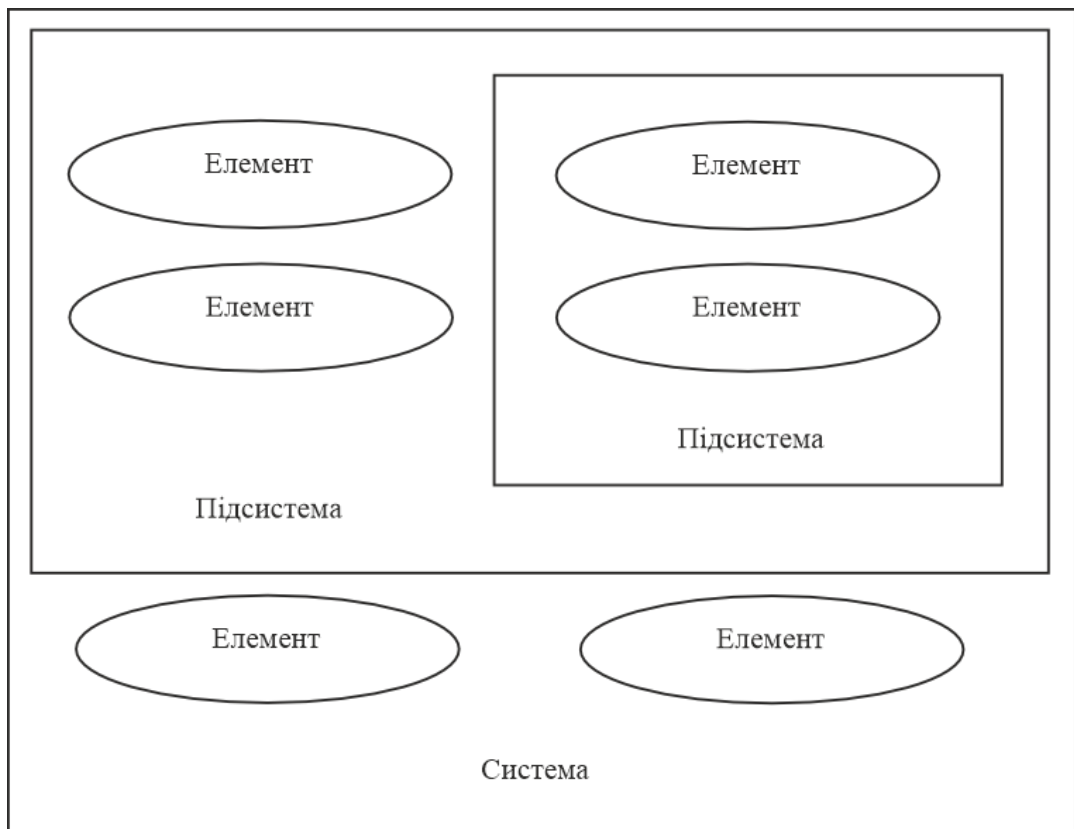


Рис. 7.3. Схема моделі складу систем

Відповідно до опису можна виділити глибше визначення системи – сукупність взаємозалежних елементів, відокремлених від середовища при взаємодії з нею як одне ціле.

У цьому випадку модель охоплює «чорну скриньку», модель складу системи та модель структури системи. Разом вони утворюють ще одну модель, яка називається *структурною схемою системи*.

Часто, структурна схема описується з допомогою математичної моделі, але на сьогодні використовують опис системи за допомогою схеми, що складалася з окремих елементів системи та зв'язків між ними. Така схема називається графом та представлена на рисунку 7.3.

У графах елементи називаються вершинами, а зв'язок між ними називаються ребрами, вершини позначаються кружками, а зв'язки між ними мають вигляд ліній. Якщо не зазначено напрями зв'язку, то такий граф називається неорієнтованим.

Вершини можуть з'єднуватися між собою будь-якою кількістю ребер (ліній), а сама вершина може бути пов'язана сама з собою, тоді ребро називається петлею.

Якщо зв'язок між вершинами мають розмірну природу, всі вони позначаються різними лініями і мають різні ваги, а графи називаються зваженими.

Теорія графів має численні програми. Особливе місце в теорії систем займають (системи) структури зі зворотними зв'язками, які відповідають кільцевим шляхам в орієнтованих графах. Часто структура інформації, що

міститься в графах, для низки досліджень недостатня, тому методи теорії графів стають допоміжними, а головними є функціональні зв'язки між входами і виходами або внутрішніми зв'язками системи (рис. 7.4).

Якщо структура систем, взаємозв'язки всередині системи змінюються у часі – то такі системи називаються *динамічними*. Тому завжди для опису реальних систем потрібно створювати моделі, які описують динаміку зміни системи для використання функціональних моделей, що відображають види зв'язку, його зміну в часі, та є найбільш адекватною моделлю опису будь-яких сучасних об'єктів.

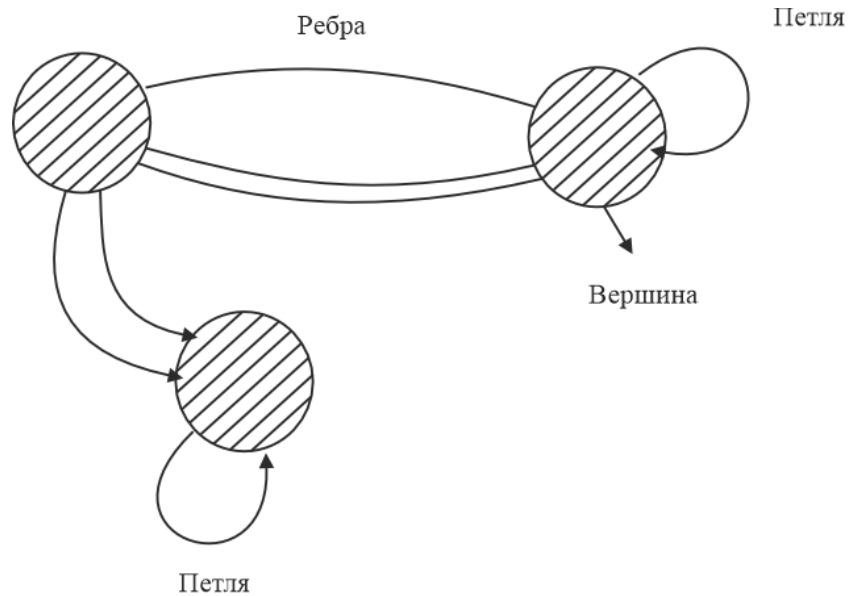



Рис. 7.3. Схема елементів графа:  
 – незв'язана вершина.

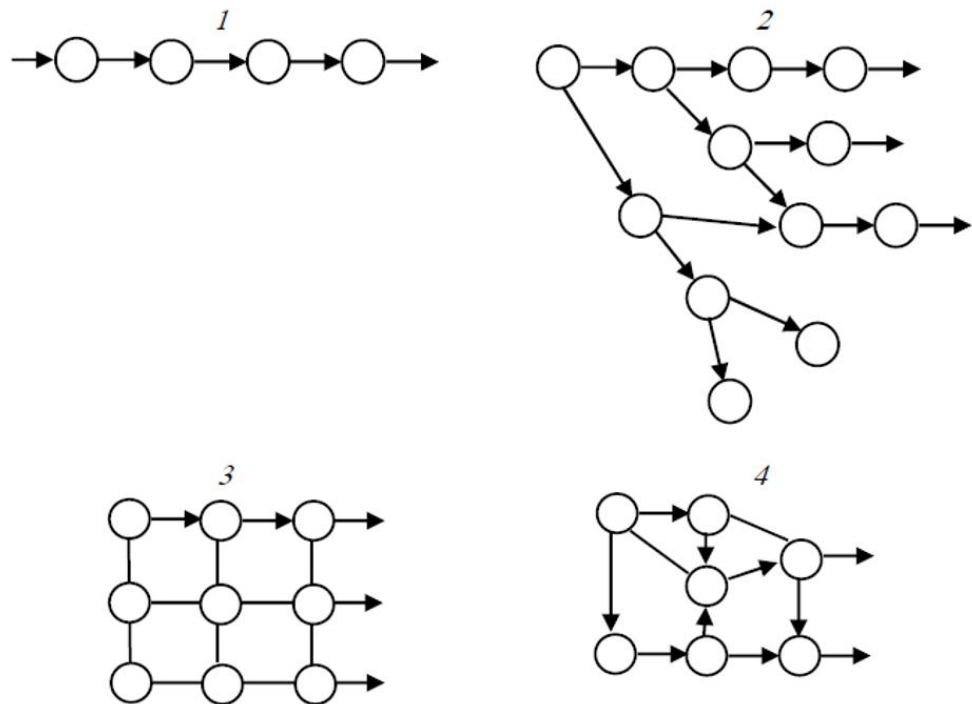


Рис. 7.4. Структури графів:

1 – лінійна; 2 – деревоподібна; 3 – матрична; 4 - мережеві.

## 7.5. Теорія інформаційного поля

### 7.5.1 Адекватність відображення.

З позицій матеріалізму сутність природи становить матерія, дана нам у відчуттях об'єктивна реальність, яка існує незалежно від наших відчуття. Це означає, що наші органи чуття дають нам інформацію які є копією матерії, що відображається. Оскільки відчуття є джерелом інформації про навколишній світ то матеріальні об'єкти сприймаються людиною з певною мірою інформації [20].

Тим не менш, оскільки в загальному випадку відображення не повністю адекватно відповідають об'єкту що відображається, має сенс говорити про інформацію як «результат» відображення, або як атрибут самої матерії.

Оскільки матерія існує у просторі, вона тим самим має свою структуру. Саме структура як розподіл матерії у просторі характеризується кількісно та є у собі. Відтворення структури матерії на якісно інших носіях або в нашій свідомості є інформація для нас.

Між цими інформаціями немає ніякої якісної відмінності, але є кількісна різниця яку можна відобразити наступним чином:

$$J_n = R_k (J_c) \quad J_c = R_k (M) \quad M \quad (7.1)$$

або в лінійному наближенні

$$J_n = R_k J_c = R_k M$$

де  $M$  - вимірюване матеріальне властивість (маса, колір, заряд тощо), що створює  $J_c$ ;  $J_n$  - чуттєва інформація (інформація для нас) або інформація сприйняття, яку надалі для стислості будемо використовувати без індексу:  $R_k$  - відносна інформаційна проникність середовища.

Співвідношення (7.1) реалізує перший із прийнятих постулатів про адекватність відображення матерії відповідно до якого інформація є функцією матерії, яка щонайменше для обмежених природень має характер пропорційної залежності.

Інформація може бути як позитивною так і негативною. Оскільки вона постає як міра кількості матерії, то й остання повинна мати різні знаки.

*Теорема Гауса.* Приймаючи наведену вище точку зору, приходимо до висновку, що об'єкти та явища природи не тільки містять певну інформацію, а й безперервно випускають її в навколишній простір незалежно від того, чи є навкруги відповідні об'єкти, здатні сприймати цю інформацію.

Оскільки чуттєве відображення протікає в часі та в просторі, то інформація є сумою потоків інформації від окремих частин матеріального об'єкта або від сукупності матеріальних об'єктів, що формують інформаційне поле навколо вимірювача, що його здатне сприймати.

Якщо говорити про відображення матеріального об'єкта або поля якоюсь довільною замкненою навколо нього поверхнею, то повна інформація складається із суми потоків інформації, що припадають на одиницю  $dS$  площі цієї поверхні, тобто з  $O = dJ / dS$ .

У такому разі повинна мати місце теорема Гауса, яка є математичним виразом філософського положення про пізнання світу:

$$M = \oint_S O dS \text{ або } J_c = \oint_S O dS \quad (7.2)$$

де  $O$  - вектор інтенсивності потоку існування (відображення): інтеграл береться по замкнутій поверхні  $S$  та охоплює відповідне явище, що вивчається, або об'єкт аналізу.

Співвідношення (7.2) означає, що будь-яка інформація в собі створює поле існування, сумарний потік якого адекватний цій інформації, тобто матерії, що служить джерелом поля є самим джерелом інформації.

З урахуванням (7.1) теорему Гауса можна подати у формі:

$$J_n = \oint_S R_k O dS = \oint_S O_n dS \quad (7.3)$$

де  $O_n = R_k O$  – вектор інтенсивності відбиття.

#### *Поле руху матерії. Інформаційний струм*

Будь-які процеси, доступні нашому спостереженню, супроводжуються обміном інформацією між системами, що беруть участь у них, та зовнішнім середовищем. Спостереження за цими процесами передбачає сприйняття суб'єктом відповідних потоків інформації. Зазначимо що одні потоки за обмежений час приносять більшу кількість інформації, інші – меншу. Зручною для зіставлення інформаційних потоків мірою служить інформаційний струм  $I$ , який природно визначити як інформацію, що проходить потоком у кожному секунду часу та визначається:

$$I = dJ / dt \quad (7.4)$$

Можна також ввести вектор густини інформаційного струму  $j$ , який визначимо як:

$$j = dI / dS \quad (7.5)$$

Розглянемо докладніше із чого складається інформаційний струм, тобто, як забезпечується перенесення інформації в системах.

Досить легко дійти висновку, що з найпоширеніших способів передачі інформації – це перенесення її разом із самими носіями інформації. За цими діями можна розглянути два види інформаційних струмів, що розрізняються джерелами енергії та інформацією яка витрачається на перенесення носіїв.

Для прикладу наведемо процес розповсюдження або поширення запахів в кімнаті. Інформація у вигляді запаху про смачну їжу або газову небезпеку здійснюється за рахунок теплової дифузії молекул у повітрі, а також за рахунок енергії потоків повітря, які не залежать від адресата чи кореспондента. При цьому напрямок передачі повідомлень не завжди узгоджується з вектором логічних зв'язків інформаційного поля, подібно до того, як в електричних полях струм перенесення (конвекції) не завжди узгоджується з вектором напруженості поля і може перетикати навіть назустріч полю. Це дає підставу називати надалі такого роду інформаційні струми *струмами перенесення*.

Згідно з визначенням, вектор щільності інформаційного струму перенесення має вигляд:

$$j_{\text{п}} = \rho v$$

де  $v$  – швидкість перенесення інформації.

Крім інформаційних струмів перенесення можна виявити також струми, що виникають під керуючим впливом інформаційного поля та згодних з ним.

Назвемо цей, завжди узгоджений з інформаційним полем струм інформації, який утворений носіями, здатними чуттєво сприймати поле і запасом енергії, що володіють, для переміщень – *чуттєвим струмом*.

## 7.5.2 Дискретні інформаційні моделі.

Хоча системні дослідження привернули особливу увагу фахівців різних областей знань лише у другій половині двадцятого століття, «математизація» надає системології особливий статус. Не можна ігнорувати, по-перше, той факт, що системологія займається не якоюсь загальнонауковою, а філософською проблемою, вивчаючи структуру відображення матерії в нашій свідомості, і, по-друге, ця особливість, мабуть, минула, оскільки раніше чи пізніше й інші розділи філософії будуть озброєні математикою з метою надання їм дієвості та прив'язки до конкретної соціальної практики.

Поняття системи стосовно наших знань про світ у цілому або про окремі аспекти буття має на увазі якусь сукупність частин, елементів, дисциплін, наук, точок зору і т.д.

Відповідно з цієї точки зору, *система* – це категорія відображення, форма уявлення матерії доступними розумінню засобами.

Особливе місце займають матеріальні продукти людської праці – машини, прилади, технічні комплекси, які збираються з деталей, вузлів і т.д. Ці машини є системи деталей і вузлів, оскільки є продуктом нашої свідомості і втілюють у собі спосіб відображення можливостей об'єктивної реальності у здійсненні тих чи інших функцій, тобто. втілюють нашу дискретну логіку.

Отже, матеріальні продукти свідомої людської діяльності, з одного боку то є системи, але, з іншого боку, вони є континуальною цілісністю. якщо розглядати їх онтологічно у зв'язку з уявленнями їх творців.

Отже, ця точка зору дає уявлення що система - це спосіб відтворення та відображення континуальної цілісності засобами нашої свідомості, нашої логіки. Інакше кажучи, система - це дискретна модель безперервного буття.

Як і будь-яка модель, система може бути: фізичною моделлю, коли вона чуттєво (за даними наших органів чуття та даними вимірювальних засобів) подібна до об'єкта, що моделюється; або логічної (у тому числі математичної) моделлю, коли її логіка подібна до логіки модельованого об'єкта; або, нарешті, імітаційною (прагматичною) моделлю, коли тільки її цілісна поведінка (вихід) аналогічно моделюванням об'єкту.

Імітаційні системи є зазвичай приватними моделями, які не претендують на адекватність вихідному об'єкту в усіх відношеннях. Фізичні і логічні моделі, навпаки, претендують на адекватність відображення вихідного об'єкта як в цілому, так і в деталях, і навіть приписують свою системну структуру об'єкту, що моделюється, та є поширеною помилкою.

Інакше кажучи, система – це діалектичний синтез взаємно виключених вимог – точності та доступності, а завданням прикладної системології та системного аналізу є вироблення засобів досягнення компромісу високої точності системного моделювання актуальних завдань практичної діяльності.

*Основні поняття дискретних інформаційних моделей.* У гносеології для відображення всіх способів отримання інформації прийнятий узагальнюючий термін «віддзеркалення», а для позначення всіх первинних джерел інформації, що утворюють існуючу поза нас об'єктивну реальність, прийнятий узагальнюючий термін «матерія». Таким чином, замість перерахування різноманітних джерел інформації та способів її отримання можна просто говорити про відображення матеріального світу (матерії) у нашій свідомості, яке завжди відбувається за допомогою наших органів чуття, тобто– зору, слуху, нюху, дотику, смаку (або технічних доповнень - вимірювальних приладів, що збільшують роздільну здатність органів чуття та доступних джерел інформації).

Згідно з формулою пізнання – «Від живого споглядання до абстрактного мислення, і від нього – до практики», можна виділити три етапи відображення дійсності: два пасивні – чуттєве та логічне відображення, і один активний – етап прагматичного відображення.

Відповідно продуктами цих етапів є чуттєва, логічна і прагматична інформація.

Явна інформація  $J$  вводиться як міра відбитої у свідомості елементної бази системи у формі:

$$J = A/\Delta A \quad (7.6)$$

де  $A$  - загальна кількість будь-яких знаків, що сприймаються вимірювальними приладами або нашими органами почуттів;  $\Delta A$  - «квант», з точністю до якого нас цікавить інформація, що сприймається, або роздільна здатність приладу.

Тут необхідно пояснити принципову різницю між  $A$  і  $J$ . Якщо  $A$  завжди прийнято виражати числом, що є класичним математичним об'єктом та задовольняє закону тотожності  $A \equiv A$  то про інформацію цього сказати не можна.

Таким чином, у прагматичному аспекті інформація завжди несе в собі вельми значний елемент суб'єктивності яка є різною для різних людей за одного і того ж значення  $A$ . Навіть за фіксованого  $\Delta A$  інформація, строго кажучи, не є числом, оскільки в межах більш менш обмежених  $\Delta A$  може мати будь-яке значення.

З іншого боку, два або кілька однакових вимірювальних приладів при вимірюванні однієї і тієї ж величини, в рамках їхньої здатності, здатні дати різну інформацію, але з однаковою достовірністю. Це означає, що інформація не задовольняє логічному закону виключеного третього, що не допускає існування кількох суперечливих, але однаково істинних величин, зате задовольняє діалектичному закону єдності і боротьби протилежностей.

Отже, інформація - це поняття, яке не піддається аналізу засобами формальної логіки і потребує застосування до нього діалектичної логіки, що забезпечує можливість аналізу не лише абсолютно, а й щодо справжніх висловлювань. З цієї точки зору  $J$  аналогічна висловлюванням природної мови, які завжди мають розмитий і відносно правдивий характер. Однак, зважаючи на дуальну природу  $J$  (число і не число) інформація на відміну від вербальних форм піддається деяким (не всім) математичним операціям.

Логічна інформація (сутність)  $H$  на відміну від  $J$  завжди належить до конкретних об'єктів, чи властивостей, яка характеризує цілий клас однорідних, у певному відношенні, об'єктів чи властивостей, будучи семантичним синтезом законів логіки.

Відповідно до основного закону класичної логіки Аристотеля, власна сутність (суть) системи зворотна обсягу поняття і про неї, тобто:

$$H = J / n \quad (7.7)$$

Обсяг поняття залежить від аспекту розгляду системи (елемента) і зазвичай передбачає їх родову приналежність.

*Інформація і ентропія.* Ентропія характеризує недостатню інформацію. При аналізі будь-яких систем людина запроваджує людський чинник, що

призводить до помилок. Для вимірювання, визначення цих помилок існує поняття *невизначеності*.

Сучасне поняття інформації та яку роль вона грає у системах склалося не відразу. Це було отримано з знань різних галузей науки. Так, наприклад, Больцман у 1877 р. упустив таку фразу, що: «Ентропія характеризує недостатню інформацію», але на цю фразу тоді ніхто не звернув уваги. Пізніше 1948 р. Шелон вивів формулу для інформаційної ентропії, яка виявилася тотожною формулі, яку вивів Больцман для термодинамічної ентропії. Вийшло, що у всіх галузях були окремо сформульовані поняття інформації та її недостатності та недотворності. Сучасна інформація встановила нерозривний зв'язок між цими явищами. Філософська теорія теж дійшла до цієї точки зору, у філософії початковий зміст інформації сприймається як знання, звістка, повідомлення тобто щось властиве людській свідомості та спілкуванню.

Однак, для відображення реального світу цього було недостатньо і було доведено, що відображення реального світу є загальною властивістю матерії, а свідомість людини є найвищою специфічною формою відображення цієї інформації.

Властивість будь-якого об'єкта (його стан) знаходиться завжди відповідно до стану іншого або інших об'єктів, іншими словами, один об'єкт містить інформацію про інше. У сучасній філософії інформація розглядається як фундаментальна властивість матерії. Тому роль інформації у сучасних системах величезна. Поняття інформації має загальність та має сенс філософської категорії. Для кібернетики та теорії систем поняття інформації так само фундаментальне і дуже важливе. Колмогоров (родоначальник кібернетики) сказав, що кібернетика - це наука, яка займається вивченням систем будь-якої природи, здатних сприймати, зберігати та опрацювати інформацію та використовувати її для управління чи регулювання систем.

Для того, щоб два об'єкти містили інформацію один про одного, необхідно, щоб між їхніми станами існувала певна відповідність. Така відповідність може встановити лише в результаті фізичної взаємодії між об'єктами, або через проміжні об'єкти при передачі певних сигналів.

*Сигнал* - це матеріальний носій інформації, засіб перенесення в часі та просторі, тобто це відповідний стан об'єкта, а будь-який стан системи має сигнальні властивості.

У штучно створених інформаційних системах сигнали організують спеціально, їх виробляють, тобто створюється сигнальний стан, який називається *кодом*. На отримання відповідного актуального сигналу в інформаційних системах при взаємодії між об'єктами, через передачу цих сигналів впливають перешкоди – шуми, тому відбувається порушення цих станів через неузгодженість кодів, якими взаємодіють об'єкти між собою. Іноді спеціально ці перешкоди створюються. Яскравим прикладом цього явища є криптографія – спеціальне розсекречення кодів.

*Типи сигналів.* Оскільки сигнали служать для перенесення інформації в просторі та часі, то вони повинні бути стійкі як до часу, так і до зміни положення в просторі. Вони поділяються на:

- статичні;
- динамічні.

У першому випадку сигнали є стабільними в часі або не змінюються, хоча б у певний моменті часу.

До другого типу відносяться сигнали, які використовуються для опису динамічних полів, наприклад, звук (застосовується модуляція та демодуляція).

Зрозуміло, що динамічні сигнали переважно використовуються для передачі сигналів, а статичні - для зберігання.

Сигнали грають у системах особливо важливу роль. Якщо енергетичні та/або речові потоки живлять систему, то потоки інформації, що переносяться сигналами, організують її функціонування та керують її роботою

Віннер сказав, що суспільство простягається до тих меж, до яких поширюється інформація, це стосується будь-якої системи. Якщо ми беремо систему суспільства, вона розвивається, управляється за допомогою інформаційних потоків., а де інформація переривається, то ви вже поза суспільством.

*Ентропія і її властивості.* Першим чи специфічним поняттям теорії інформації є поняття невизначеності випадкового об'єкта. Для кількісної оцінки цієї невизначеності було запроваджено поняття ентропії.

*Ентропія* – це кількісний захід невизначеності. Почнемо з найпростішого варіанта події, нехай деяка подія може статися з ймовірністю 0,99 та статися з ймовірністю 0,01, а інша подія має можливість відповідно 0,5 та 0,5. Очевидно, що в першому випадку результатом досвіду, експерименту майже напевно є настання події, а в другому невизначеність така велика, що від прогнозу слід утриматися.

Як міру невизначеності інформації було запроваджено поняття, зване ентропією випадкового об'єкта чи системи.

Якщо будь-який об'єкт  $A$  має стан  $A_1 \dots A_n$  а ймовірність кожного з цих станів  $p_1, \dots p_n$  то ентропія цієї події:

$$H(A) = - \sum_{k=1}^n p_k \log p_k \quad (7.8)$$

Розглянемо властивості цієї ентропії:

1) Якщо ймовірність наступу однієї з  $n$ -подій = 1, то ентропія цього стану = 0

$$H(p_1, \dots p_n) = 0, \text{ за умови } p_i = 1.$$

2) Ентропія сягає свого невеликого значення у разі, якщо ймовірності  $p_1, \dots p_n$  рівні між собою, тобто  $H(p_1, \dots p_n) = \max$  якщо:

$$p_1 = p_2 = p_3 = \dots = p_n = 1/n;$$

$$\sum p_i = 1, p_i = 1/n.$$

3) Якщо об'єкти  $A$  і  $B$  незалежні, їх ентропія дорівнює сумі ентропії кожного об'єкта

$$H(A \cap B) = H\{p_k\} + H\{q_m\} = H(A) + H(B).$$

4) Якщо об'єкти  $A$  і  $B$  залежні, то їх ентропія

$$H(A \cap B) = H(A) + H(B/A),$$

Тобто, за умови настання події  $A$ .

5) Ентропія події  $A >$  ентропії  $H(A/B)$  події  $A$  за події  $B$

$$H(A) > H(A/B).$$

Тобто, інформація про об'єкт завжди зменшує невизначеність події  $A$ . Якщо  $A$  та  $B$  залежні, і не змінюються, та якщо події  $A$  та  $B$  незалежні.

Цей апарат описує дискретні події, пов'язані з описом перервних процесів, але на практиці часто доводиться стикатися з описом невизначеності безперервних випадкових процесів і виникає ряд складності. Однак не вдаючись у подробиці, а зробивши таку заміну –

$$\sum_{k=1}^n p_k \log p_k \rightarrow \int_x p(x) \log p(x) dx,$$

функціонал можна записати у вигляді:

$$h(X) = - \int_x p(x) \log p(x) dx. \quad (7.9)$$

$p(x)$  – щільність ймовірності настання події, виражена ентропією, називається диференціал.

Процес отримання будь-якої інформації можна інтерпретувати як зміну невизначеності внаслідок передачі сигналів. При цьому корисний сигнал, що відправляється, є послідовністю незалежних символів з ймовірністю  $p(x_i)$ , сигнал, що приймається, є також набором символів  $y_k$  того ж кодування (алфавіту), і якщо у нас відсутній шум який має вплив на цю передачу, приймаються і відправляються сигнали  $Y_k = X_m$ , але оскільки за будь-якої передачі ми маємо перешкоди, йде спотворення сигналу, то на приймальній стороні системи з'явиться невизначеність, вона може бути на стороні передачі інформації. При цьому, на стороні що передала  $H(X)$  називаємо – апіорна ентропія, а на приймальній стороні  $H(X/Y)$  має назву - апостеріорна.

Щоб оцінити кількість інформації, яка була передана від одного об'єкта до іншого, береться різниця апіорної та апостеріорної інформації. Ця кількість інформації в цьому випадку - різниця між ентропією.

$$I(X, Y) = H(X) - H(X/Y), \quad (7.10)$$

Наведемо основні властивості кількості інформації:

1) Кількість інформації у випадковому об'єкті  $A$  щодо об'єкта  $Y$  дорівнює кількості інформації в  $Y$  щодо  $X$

$$I(X, Y) = H(X).$$

2) Кількість інформації завжди невід'ємно

$$I(X, Y) = I(Y, X) \geq 0.$$

3) Для дискретних об'єктів АС справедлива рівність

$$I(X, X) = I(X, X).$$

Таким чином, кількість інформації вимагає одиниці виміру, за одиницю ентропії приймають невизначеність випадкового об'єкта, у якого його ентропія дорівнює :

$$H(X) = - \sum_{i=1}^k p_k \log p_k = 1.$$

Для конкретизації беремо  $k = 2$  та в основі  $\log m = 2$ , тоді виходить тотожність:

$$-p_1 \log_2 p_1 - p_2 \log_2 p_2 = 1$$

Рішенням цієї тотожності є окремий випадок

$$p_1 = p_2 = 1/2$$

За одиницю інформації прийнято визначати величину, звану бітом. При цьому зазначимо, якщо ми беремо за основу  $\log e(ln)$  (натуральний  $\log$ ) то одиниця інформації - «БІТ».

## ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ

1. Назвіть і коротко охарактеризуйте основні класи моделей у системному аналізі (детерміновані/стохастичні, дискретні/неперервні, аналітичні /імітаційні). На ваш погляд коли та який клас доцільно застосовувати?

2. Опишіть концептуальну модель відділу розробки ПЗ: які основні сутності, потоки та події треба врахувати при розробці ПЗ (ресурси, завдання, черги, помилки, релізи)? Які припущення узагальнять модель?

3. Наведіть приклад статистичного (емпіричного) методу для оцінки часу виконання задач у команді розробки. Як побудувати розподіл інтервальних часів на основі даних?

4. Запропонуйте модель (аналітичну або імітаційну) для оцінки впливу зміни середньої кількості баг-репортів на пропускну спроможність відділу; які параметри та метрики ви б вимірювали?

5. Що таке «теорія інформаційного поля» у контексті моделювання систем? Наведіть ідею застосування інформаційних полів для представлення інформаційних потоків у команді розробки.

## Тема 8. Системний аналіз об'єктів та процесів методом імітаційного моделювання

### 8.1. Імітаційні моделі, сфера застосування та основні визначення.

Імітаційне моделювання (ІМ) є потужним інструментом системного аналізу, який дозволяє вивчати поведінку складних систем у реальному часі

або в умовах, в яких неможливо або економічно недоцільно проводити експерименти на реальних умовах. Імітація дозволяє відтворювати реальні процеси, оцінюючи їхню ефективність, варіативність та стійкість до змін у зовнішньому середовищі [18].

Наведемо загальну класифікацію основних видів моделювання:

- концептуальне моделювання – представлення системи за допомогою спеціальних знаків, символів, операцій над ними або за допомогою природних чи штучних мов;

- фізичне моделювання – моделюється об'єкт або процес відтворюється виходячи із співвідношення подібності, що впливає зі схожості фізичних явищ;

- структурно - функціональне - моделями є схеми (блок-схеми), графіки, діаграми, таблиці, малюнки зі спеціальними правилами їх об'єднання та перетворення:

- математичне (логіко-математичне) моделювання - побудова моделі здійснюється засобами математики та логіки;

- імітаційне (програмне) моделювання - при якому логіко-математична модель досліджуваної системи є алгоритмом функціонування системи, програмно реалізований на комп'ютері.

Домінантною тенденцією сьогодні є взаємопроникнення всіх видів моделювання, та різних інформаційних технологій у сукупність галузі моделювання. Так наприклад, імітаційне моделювання включає концептуальне моделювання (на ранніх етапах формування імітаційної моделі) і логіко-математичне (включаючи методи штучного інтелекту) - для цілей опису окремих підсистем і процесів моделі, а також у процедурах обробки та аналізу результатів обчислювального експерименту та прийняття рішень.

Технологію проведення та планування обчислювального експерименту з відповідними математичними методами привнесена до імітаційного моделювання з фізичного (натурного) моделювання. Зрештою, структурно-функціональне моделювання використовується із створенні та формування різних діаграмних уявлень під час створення імітаційних моделей.

Суть імітаційного моделювання є створення моделі, яка описує та відтворює діяльність самої системи.

До цього методу звертаються у випадках:

- коли аналітична модель не може бути створена або вона не здатна повністю відображати поведінку системи. Наприклад, за наявності параметра - часу, або послідовностей чи причинно-наслідкових зв'язків;

- коли неможливо або дуже дорого експериментувати на реальних складових системи чи самої системі загалом.

Перевагою імітаційних моделей є можливість управління часом у моделі: так час у моделі системи з повільною мінливістю можна прискорити, тоді як для систем, у яких процеси відбуваються дуже швидко, час можна уповільнити, щоб детально проаналізувати зміни, що відбуваються в системі.

ІМ будується на основі переходу системи з одного стану в інший. Передбачається, що відомий набір правил, якими визначається наступне стан системи з урахуванням поточного. У кожному стані системи виходить набір вихідних параметрів.

ІМ дозволяє анімувати динаміку системи, що наочніше будь-яких таблиць та графіків. Крім того, в ході імітаційного моделювання можна набути значення будь-якої величини в системі в будь-який момент модельного часу. ІМ має дуже широкий спектр застосування: від моделей з високим рівнем абстракції, в яких деталі системи не важливі і метою моделювання є стратегічний розвиток системи (наприклад, моделювання соціальних систем, екосистем тощо). до моделей з детальним опрацюванням окремих об'єктів системи (наприклад, рух пішоходів, військові ігри, склади).

ІМ застосовується в багатьох сферах, таких як бізнес, економіка, виробництво, ІТ, транспорт, охорона здоров'я та інші. Цей метод є особливо корисним для вивчення дискретних процесів та для оптимізації прийняття рішень у ситуаціях, де прямі експерименти є складними або неможливими.

Метод ІМ дозволяє вирішувати завдання високої складності, забезпечує імітацію будь-яких складних процесів з великою кількістю елементів, окремі функціональні залежності в таких моделях можуть описуватися складними математичними співвідношеннями. Тому імітаційне моделювання ефективно використовується у завданнях дослідження систем зі складною структурою з метою вирішення конкретних проблем.

## **8.2. Структура процесу імітаційного моделювання**

Процес розробки імітаційної моделі починається зі створення простої моделі, яка потім поступово ускладнюється відповідно до вимог, що пред'являються вирішуваною проблемою [18]. У процесі ІМ можна назвати такі основні етапи (рис. 8.1):

1. Формулювання проблеми та визначення пелен імітаційного дослідження. Документованим результатом на цьому етапі є складений змістовний опис об'єкта моделювання:

2. Розробка концептуального опису. Результатом діяльності системного аналітика є концептуальна модель (або вербальний опис) та вибір способу формалізації для заданого об'єкта моделювання.

3. Формалізація ІМ. Складається формальний опис об'єкта моделювання.

4. Програмування ІМ (розробка програми-імітатора). На етапі здійснюється вибір засобів автоматизації моделювання, алгоритмізація, програмування та налагодження імітаційної моделі.

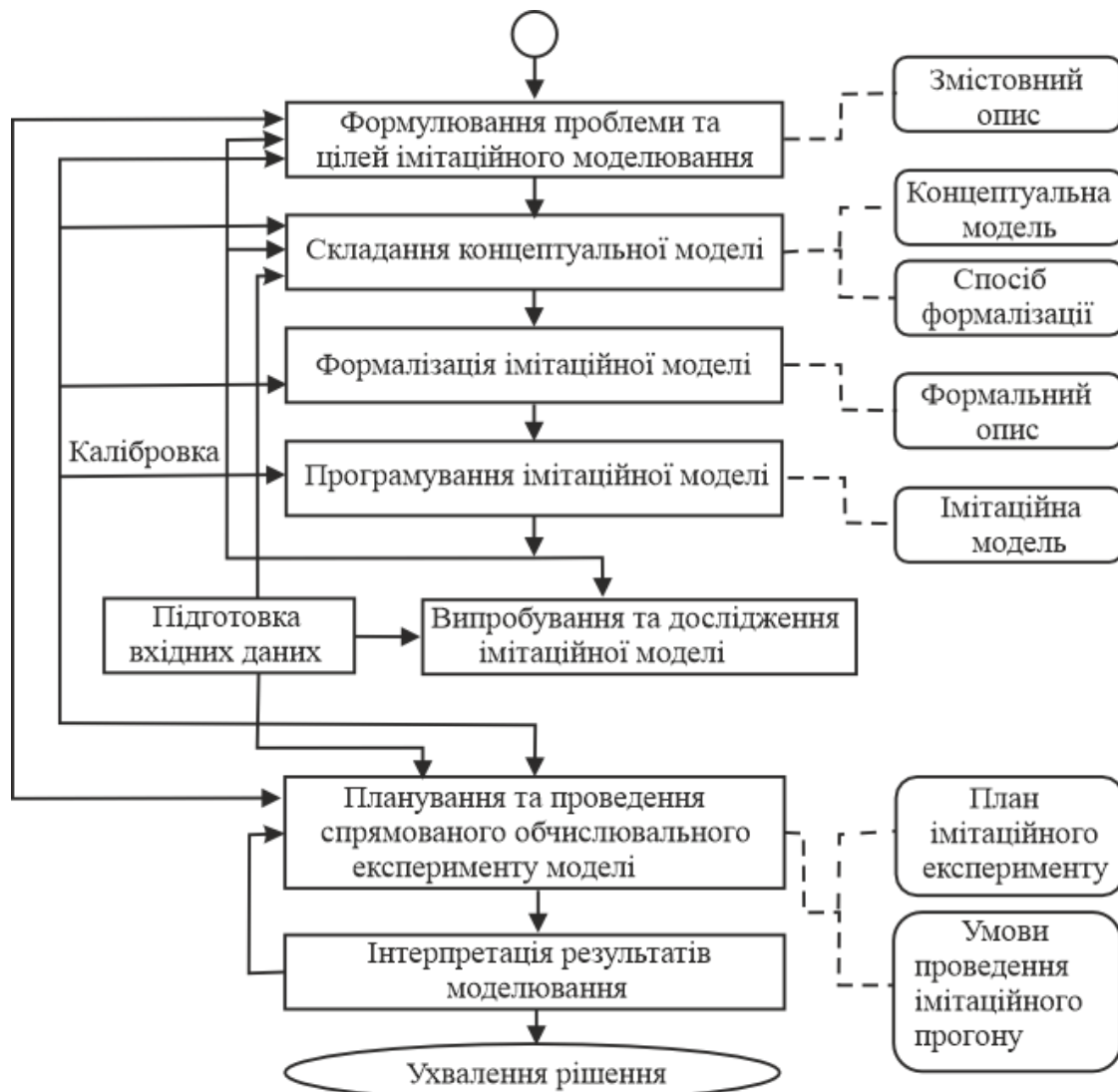


Рис. 8.1. Технологічні етапи імітаційного моделювання.

5. Випробування та дослідження моделі, перевірка моделі. Проводиться верифікація моделі, оцінка адекватності, вивчення властивостей ІМ та інші процедури комплексного тестування розробленої моделі.

6. Планування та проведення імітаційного експерименту на даному технологічному етапі здійснюється стратегічне та тактичне планування імітаційного експерименту. Результатом є складений та реалізований план експерименту, задані умови імітаційного прогону для обраного плану.

7. Аналіз результатів моделювання. Дослідник проводить інтерпретацію результатів моделювання та їх використання – власне прийняття рішень.

Першим завданням імітаційного дослідження є точне визначення проблеми та детальне формулювання цілей дослідження. Потім починається етап побудови моделі досліджуваної системи. Модель включає статистичний та динамічний опис системи. У статистичному описі визначаються елементи системи та його характеристики, а динамічному - взаємодії елементів системи, у яких відбуваються зміни її стану у часі.

Модель має бути простою для розуміння і водночас досить складною, щоб реалістична відобразити характерні риси реальної системи. Найбільш важливими є прийняті розробником рішення щодо того, чи вірні прийняті спрощення та припущення, які елементи та взаємодії між ними мають бути включені до моделі. Рівень деталізації моделі залежить від мети її створення. Необхідно розглядати лише ті елементи, які мають важливе значення для вирішення досліджуваної проблеми. Як на етапі формулювання проблеми, так і на етапі моделювання необхідна тісна взаємодія між розробниками моделі та її користувачами.

На етапі розробки моделі визначаються вимоги до вхідних даних. Деякі з цих даних можуть бути в розпорядженні розробника моделі, тоді як для збору інших знадобляться час і зусилля. Зазвичай, значення таких вхідних даних задаються на основі деяких гіпотез або попереднього аналізу. У деяких випадках значення одного (і більше) вхідних параметрів надають невеликий вплив на результати прогонів моделі. Чутливість одержуваних результатів зміни вхідних даних може бути оцінена шляхом проведення серії імітаційних прогонів для різних значень вхідних параметрів. Отже, імітаційна модель може використовуватися для зменшення витрат часу та коштів на уточнення вхідних даних.

Формулювання проблеми та визначення цілей імітаційного дослідження.

На першому етапі сформулюється проблема, що стоїть перед дослідником, приймається рішення щодо доцільності застосування методу ІМ. Потім визначаються цілі, які мають бути досягнуті в результаті імітації. Від формулювання цілей значною мірою залежить вибір типу імітаційної моделі та характер подальшого імітаційного дослідження на імітаційній моделі.

На цьому етапі визначається та детально вивчається об'єкт моделювання, ті сторони його функціонування, які становлять інтерес для дослідження. Результатом робіт на даному етапі є змістовний опис об'єкта моделювання із зазначенням цілей імітації та тих аспектів функціонування об'єкта моделювання, які необхідно вивчити на імітаційній моделі. Змістовний опис складається в термінології реальної системи, мовою предметної області, зрозумілою замовнику.

У ході складання змістовного опису об'єкта моделювання встановлюються межі вивчення об'єкта, що моделюється, дається опис зовнішнього середовища, з яким він взаємодіє. Формулюється також основні критерії ефективності, за якими передбачається проводити порівняння на моделі різних варіантів рішень, проводиться генерація та опис альтернатив.

Загального рецепта складання змістовного опису немає. Успіх залежить від інтуїції розробника та знання реальної системи.

Загальна технологія або послідовність дій на цьому етапі: збір даних про об'єкт моделювання та складання змістовного опису об'єкта моделювання: далі слід: вивчення проблемної ситуації; уточнення цілей

моделювання; обґрунтовується необхідність моделювання та здійснюється вибір методу моделювання. На цьому етапі чітко формулюється цілі моделювання. Цілі моделювання визначають загальний задум моделі та пронизують усі наступні етапи імітаційного моделювання: далі здійснюється формування концептуальної моделі досліджуваного об'єкта.

### 8.3. Розробка концептуальної моделі об'єкта моделювання

Концептуальна модель логіко-математичний опис моделюванні системи відповідно до формулювання проблеми [5]. Основний зміст цього етапу – формулювання загального задуму моделі, перехід від реальної системи до логічної схеми її функціонування (схематично загальний зміст цього технологічного переходу демонструється рис. 8.2). Тут наводиться опис об'єкта у термінах математичних понять та алгоритмізація функціонування її компонентів. Концептуальний опис є спрощеним алгоритмічним відображенням реальної системи.

При розробці концептуальної моделі здійснюється встановлення основної структури моделі, що включає статичний і динамічний опис системи. Визначаються межі системи, наводиться опис довілля, виділяються суттєві елементи та дається їх опис, формуються змінні, параметри, функціональні залежності як окремих елементів і процесів, так всієї системи, обмеження, цільові функції (критерії).

Результат роботи на цьому етапі - документований концептуальний опис плюс обраний спосіб формалізації системи, що моделюється. При створенні невеликих моделей етап поєднується з етапом складання змістовного опису моделі.

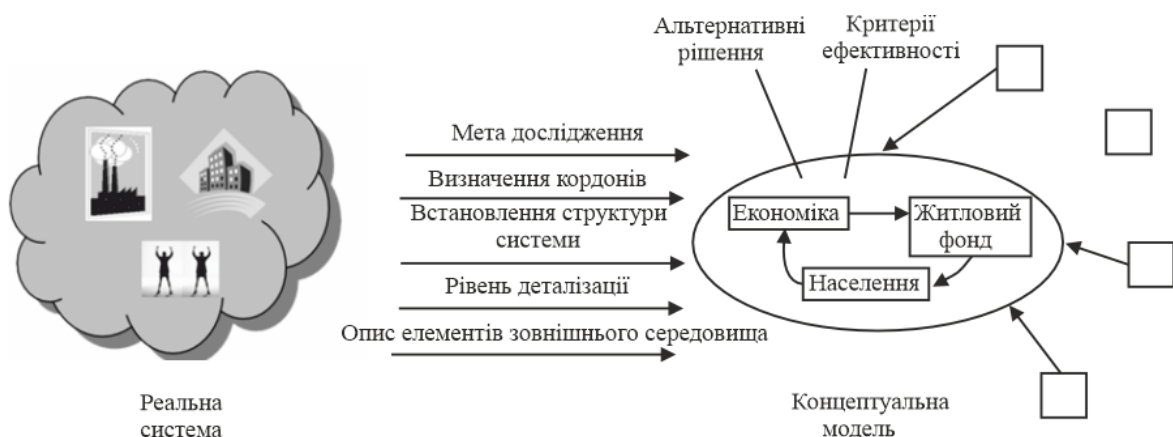


Рис. 8.2. Перехід від реальної системи до логічної схеми її функціонування

Побудова концептуальної моделі починається з того, що на основі мети моделювання встановлюються межі системи, що моделюється, визначаються впливи зовнішнього середовища. Висуваються гіпотези та фіксуються всі

припущення, необхідні для побудови імітаційної моделі. Обговорюється рівень деталізації моделювання процесів.

Нагадаємо що система є сукупність взаємозалежних елементів. Визначення системи завжди суб'єктивне, залежить від мети моделювання та від того, хто саме визначає систему. На цьому етапі здійснюється декомпозиція системи. Визначаються найбільш суттєві у сенсі сформульованої проблеми елементи системи (виконується структурний аналіз моделювання системи) та взаємодії між ними, виявляються основні аспекти функціонування моделювання систем (складається функціональна модель), наводиться опис зовнішнього середовища [21]. Декомпозиція системи (об'єкта моделювання) або виділення підсистем є операція аналізу. Елементи імітаційної моделі повинні відповідати реально існуючим фрагментам у системі. Складна система розбивається на частини, зберігаючи у своїй зв'язку, які забезпечують взаємодія. Важливо визначити, які компоненти будуть включені в модель, які будуть винесені в зовнішнє середовище і які взаємозв'язки будуть встановлені між ними.

Опис довкілля виконується з тих міркувань, що елементи довкілля надають певний вплив на елементи системи, проте вплив самої системи на них, як правило, небагато.

При обговоренні рівня деталізації моделі важливо розуміти, що в основі будь-якої декомпозиції лежать два суперечливі принципи: повнота і простота. Зазвичай на початкових етапах складання моделі спостерігається тенденція до обліку надмірно великої кількості компонентів та змінних. Однак хороша модель - проста, адже відомо, що ступінь розуміння явища обернено пропорційна числу змінних, що фігурують у його описі. Модель, перевантажена деталями, може стати складною і важкою.

Компромід між цими двома полюсами полягає у тому, що в модель включаються лише суттєві компоненти, при цьому суттєві вони є по відношенню до мети аналізу.

В опис системи мають бути включені критерії ефективності функціонування системи та оцінювані альтернативні рішення, останні можуть розглядатися як керуючі або сценарні параметри моделі системи. При алгоритмізації моделювання процесів уточнюються також основні змінні моделі, що у її описі.

Кожна модель є деякою комбінацією таких складових, як компоненти, змінні, параметри, функціональні залежності, обмеження, цільові функції (критерії).

*Формалізація та програмування імітаційної моделі.* На цьому етапі імітаційного дослідження здійснюється формалізація об'єкта моделювання. Процес формалізації складної системи включає вибір способу формалізації та складання формального опису системи.

У процесі побудови моделі можна виділити 3 рівні її уявлення:

- неформалізований - концептуальна модель;
- формалізований - формальна модель:

- програмний - імітаційна модель.

Кожен рівень відрізняється від попереднього ступенем деталізації моделюванні системи та способами опису її структури та процесу функціонування. У цьому рівень абстрагування зростає.

Концептуальна модель це систематизований змістовний опис моделі, що моделюється (або проблемної ситуації) неформальною мовою. Неформалізований опис імітаційної моделі, що розробляється, включає визначення основних елементів моделюванні системи, їх характеристики та взаємодії між елементами власною мовою, можуть використовуватися таблиці, графіки, діаграми і т.д. Неформалізоване опис моделі необхідне як самим розробникам (під час перевірки адекватності моделі, її модифікації тощо.), так взаєморозуміння з фахівцями інших профілів.

Далі формалізований опис перетворюється на програму-імітатор відповідно до деякої методики (технології програмування).

Може виявитися, що інформації, що є в концептуальному описі, недостатньо для формалізації об'єкта моделювання. У цьому випадку необхідно повернутися до етапу складання змістовного опису та доповнити його даними, необхідність яких виявляється при формалізації об'єкта моделювання. Насправді таких повернень може бути кілька.

Спостерігається різноманітність формальних схем та концепцій структуризації (алгоритмічних моделей), що знайшли застосування в імітаційному моделюванні. Схеми формалізації орієнтуються на різні математичні теорії і виходять з різних уявлень про досліджувані процеси.

Найбільш відомі та широко використовуються на практиці концепції формалізації: агрегатовані системи та автомати; мережі Петрі та їх розширення; методи та звичайно-різнісні рівняння системної динаміки, які широко використовуються в практиці аналізу та опису динаміки складних систем та побудови імітаційних моделей.

Концепція структуризації (схема представлення алгоритмічних моделей) більш-менш явно лежить в основі всіх мов і систем моделювання і підтримується спеціально розробленими прийомами і технологією програмування. інші елементи систем масового обслуговування. В рамках однієї формальної схеми можуть бути реалізовані різноманітні алгоритмічні моделі.

Формалізація моделі може виконуватися як поза системою моделювання, так і засобами системи моделювання. У ряді сучасних систем моделювання, поряд з апаратом, який підтримує ту чи іншу концепцію структуризації, є спеціальні засоби, що забезпечують застосування у системі певної концепції формалізації.

*Програмування імітаційної моделі.* Концептуальний або формальний опис моделі складної системи перетворюється на програму-імітатор відповідно до деякої методики та техніки програмування, із застосуванням мов і систем моделювання. Важливим моментом є коректний вибір інструментального засобу для реалізації імітаційної моделі.

*Збір та аналіз вихідних даних.* Не завжди цей етап виділяється як самостійний, проте робота, що виконується на цьому етапі, виключно важлива, трудомістка. Якщо програмування та трасування імітаційної моделі можна виконувати на гіпотетичних даних, то майбутнє експериментальне дослідження необхідно виконувати реальному потоці даних. Від цього залежить багато в чому точність результатів моделювання.

Основні методи отримання вихідних даних:

– з існуючої документації на систему (дані офіційних та інших звітів, статистичні збірники);

– фізичне експериментування. Часто для завдання вихідної інформації необхідно провести натурні експерименти на системі, що моделюється, або її прототипах. Такий підхід застосовується для космічних, військових досліджень, в авіації. У простіших випадках можна проводити вимірювання, наприклад, хронометраж при виконанні виробничих операцій;

– попередній, апріорний синтез даних. Іноді вихідні дані можуть не існувати, і сама природа моделі, що моделюється, виключає можливість фізичного експериментування (наприклад, проєктовані системи, прогнозування в соціальних і політичних дослідженнях). І тут пропонують різні прийоми попереднього синтезу даних. Наприклад, при моделюванні інформаційних систем тривалість виконання інформаційної вимоги оцінюється на підставі трудомісткості алгоритмів, що реалізуються на ЕОМ. До цих методів відносять різні процедури, що ґрунтуються на загальному аналізі проблематики, анкетуванні, інтерв'юванні, широкому застосуванні методів експертного оцінювання.

Наступне питання пов'язане з проблемою ідентифікації вхідних даних для стохастичних систем, що моделюються є таким. Вхідні (і вихідні) змінні стохастичної моделі зазвичай випадкові величини, вектори, функції, випадкові процеси. Тому виникають додаткові труднощі, пов'язані із синтезом рівнянь щодо невідомих законів розподілу та визначенням імовірнісних характеристик (середніх значень, дисперсії, кореляційних функцій тощо) для аналізованих процесів та їх параметрів. Необхідність статистичного аналізу при збиранні та аналізі вхідних даних пов'язана із завданнями визначення виду функціональних залежностей, що описують вхідні дані моделі, оцінкою конкретних значень параметрів цих залежностей, а також перевіркою значущості параметрів. Для підбору теоретичних розподілів випадкових величин застосовують відомі методи математичної статистики, що ґрунтуються на визначенні параметрів емпіричних розподілів та перевірці статистичних гіпотез з використанням критеріїв згоди про те, чи узгоджуються наявні емпіричні дані з відомими законами розподілу (на статистично прийнятному довірчому рівні).

*Випробування та дослідження властивостей імітаційної моделі.* Після того, як імітаційна модель реалізована на ЕОМ, необхідно провести випробування, перевірку достовірності моделі. Це є надзвичайно важливим

моментом у імітаційному моделюванні. Перевірка, виконана не ретельно, може призвести до невідомих наслідків.

На етапі випробування та дослідження розробленої імітаційної моделі організується комплексне тестування моделі – запланований ітеративний процес, спрямований головним чином на підтримку процедур верифікації та валідації імітаційних моделей та даних.

Якщо в результаті проведених процедур модель виявиться недостатньо достовірною, може бути виконане калібрування імітаційної моделі (в моделюючий алгоритм вбудовуються калібрувальні коефіцієнти) з метою забезпечення адекватності моделі, або в більш складних випадках можливі численні ітерації на ранні етапи з метою отримання додаткової інформації про об'єкт, що моделюється, і доопрацювання імітаційної моделі. Наявність помилок у взаємодії компонент моделі повертає дослідника до початкового етапу створення імітаційної моделі.

*Спрямований обчислювальний експеримент на імітаційній моделі. Аналіз результатів моделювання та прийняття рішень.*

Організація спрямованого обчислювального експерименту на імітаційної моделі передбачає вибір та застосування різних аналітичних методів для проведення та обробки результатів імітаційного дослідження. Найбільш уживаними для цілей обчислювального експерименту аналітичними методами є методи планування обчислювального експерименту, регресійний та дисперсійний аналіз, градієнтні та інші методи оптимізації, про застосування яких йтиметься нижче. У масштабних дослідженнях організуються сценарні дослідження розробленої імітаційної моделі. Проведене дослідження повинне дозволити зробити деякі висновки щодо отриманих результатів, достатні для прийняття рішень щодо зазначених на ранніх етапах проблем та завдань.

Таким чином, імітаційне дослідження є трудомістким ітеративним процесом, який вимагає від розробників моделей певних інтелектуальних витрат та технологічних зусиль.

#### **8.4. Принципи організації імітаційного моделювання**

Система масового обслуговування (СМО) - абстрактний об'єкт, у якому виконується послідовність операцій, включає сукупність приладів обслуговування, пов'язані певним логічним порядком [22].

Структура систем масового обслуговування представлена на рисунку 8.3.

Заявка характеризується моментом появи на вході системи, статусом по відношенню до інших заявок, деякими параметрами, що визначають потреби в часових ресурсах обслуговування.

Заявки, що постійно надходять на обслуговування, утворюють потік заявок - сукупність заявок, розподілену в часі.

Потік заявок може бути однорідним (з точки зору обслуговування всі заявки є рівноправними) і неоднорідним.

Основний параметр потоку заявок – проміжок часу між моментами надходження 2-х сусідніх заявок.

Потік заявок може бути стаціонарним та нестаціонарним (наприклад, змінюватись в залежності від часу доби).

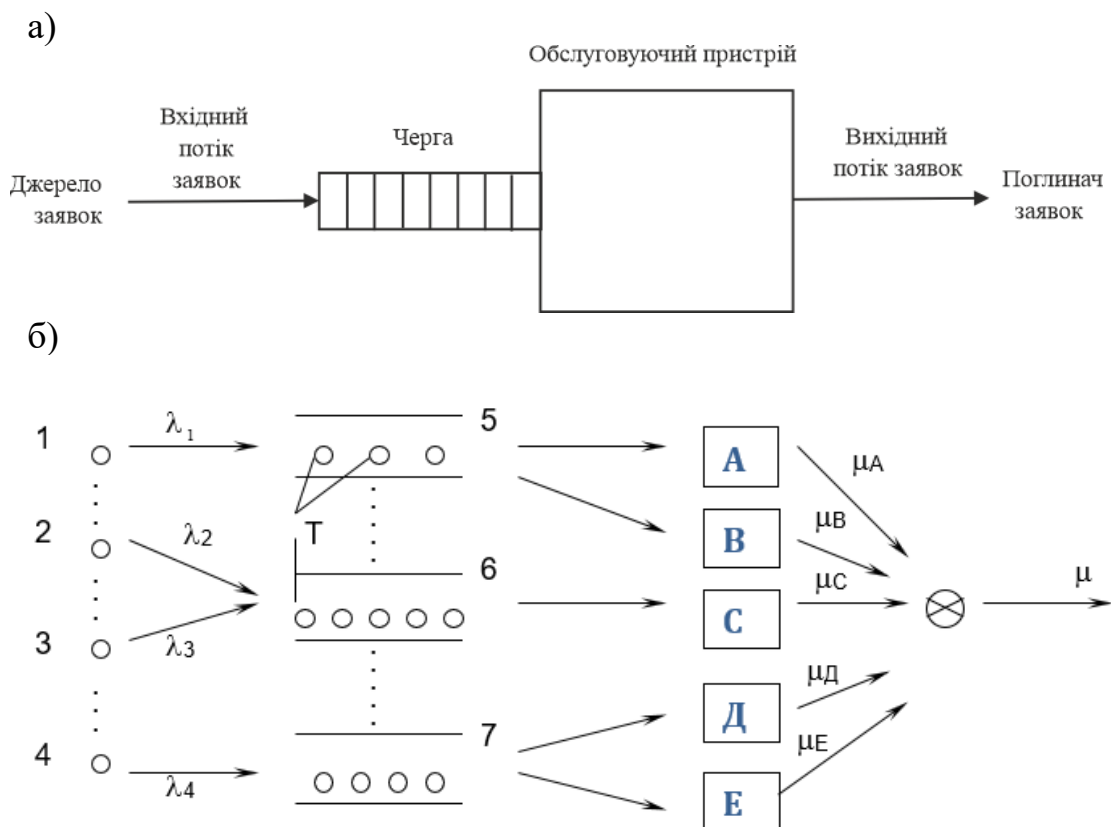


Рис. 8.3. Структура систем масового обслуговування  
 Позначення: 1, 2, 3, 4 – джерела потоку вимог; 5, 6, 7 – черги;  
 Т – вимоги; А, В, С, Д, Е – пункти або прилади обслуговування;  
 $\lambda$  – інтенсивності вхідного потоку вимог;  $\mu$  – інтенсивність обслуговування; 8 – суматор чи накопичувач обслужених вимог.

Потік заявок розглядається як випадковий процес, що характеризується функцією розподілу періоду надходження заявок (найпростіший потік, потік Ерланга).

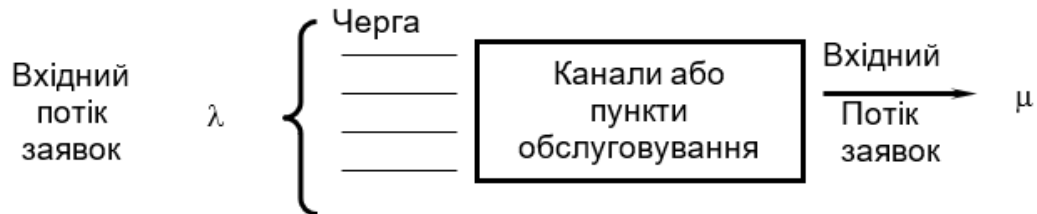
У момент виконання операцій елемент системи, в якому відбуваються операції – зайнятий, інакше – він буде вільний. Якщо канал вільний, то заявка приймається до обслуговування – знаходиться в очікуванні.

Обслуговування кожної заявки каналом означає затримку в ньому заявки на час, що дорівнює періоду обслуговування. Після обслуговування заявка залишає пристрій обслуговування (ПО). Таким чином пристрій обслуговування характеризується часом обслуговування заявки.

При випадковому характері надходження заявок утворюються черги.

Реальний процес функціонування СМО слід представляти як послідовності фаз обслуговування, виконуваних різними пристроями. Приклади багатофазного обслуговування: обслуговування покупців у магазині (прилавок. каса); виробничо-технологічний процес (обробка деталей на верстатах). Ці багатофазні системи можуть мати складну структуру (стохастичні мережі), як показано на рис 8.4.

Таким чином, загальна функціональна схема СМО матиме вигляд:



Функціональна схема СМО.

$$\lambda = \frac{N_1}{T_2} = \left[ \frac{\text{кількість заявок, що поступили}}{\text{час надходження}} \right]$$

$$\mu = \frac{N_2}{T_2} = \left[ \frac{\text{кількість обслуговуваних заявок}}{\text{час обслуговування}} \right]$$

Рис. 8.4. Стохастична мережа

Тут:  $\lambda$ - Інтенсивність вхідного потоку;  $\mu$  - Інтенсивність вихідного потоку заявок.

Розмірність  $[\lambda]$  і  $[\mu]$  буде  $\left[ \frac{1}{c} \right]$ . Якщо  $\mu \leq \lambda$ , то СМО будуть черги, таке

СМО називається з очікуванням.

Обслугована заявка звільняє прилад обслуговування та залишає систему (поглинач заявок), або рухається далі відповідно до технологічної схеми роботи системи.

Розрізняють такі типи СМО:

- одноканальні та багатоканальні - (за кількістю ПО);
- з очікуванням та без очікування (з відмовними);
- з обмеженням на довжину черги (або з обмеженим очікуванням) та без обмеження;
- з упорядкованою чергою та з неупорядкованою чергою;
- з пріоритетами та без пріоритетів;
- та ін.

Будь-яка модель будується для того, щоб оцінити якісь показники якості.

Основні показники якості обслуговування:

- загальна кількість обслужених заявок за будь-який проміжок часу;

- пропускна спроможність - середня кількість заявок, обслужених в одиницю часу;
- частка обслуговуваних заявок.
- частка заявок, які отримали відмову;
- час перебування заявки в системі (від моменту надходження заявки до системи до моменту завершення її обслуговування);
- середній час обслуговування (функція розподілу часу обслуговування);
- середня довжина черги;
- середній час очікування;
- завантаження каналів - коефіцієнт використання (як частка часу, протягом якого ПО було зайнято) - характеризує ступінь простою ПО;
- та ін.

*Класичні математичні методи дослідження СМО запропоновані теорією масового обслуговування. Аналітичні методи не завжди можуть застосовуватися для вирішення практичних завдань: наприклад, часто використовується припущення про найпростіший потік заявок (проте для різних фаз обслуговування він може бути не найпростішим), про однотипні пристрої і т.п. В імітаційному моделюванні подібні та інші обмеження знімаються: можуть застосовуватися довільні закони розподілу, різні схеми обслуговування (наприклад, порядок обслуговування заявок з черги тощо), СМО досліджується не обов'язково в стаціонарному режимі (можливе вивчення перехідного режиму, коли показники відрізняються від граничних асимптотичних).*

Сутність методу ім для СМО полягає у наступному:

Використовуються спеціальні алгоритми, що дозволяють виробляти випадкові реалізації потоків подій та моделювати процеси функціонування обслуговуючих систем. Далі здійснюється багаторазове відтворення, реалізація випадкових процесів обслуговування та на виході моделі – статистична обробка отриманих статистичних даних, оцінка показників якості обслуговування.

Розглянемо принципи імітаційного моделювання на прикладі найпростішої моделі у вигляді одноканальної системи масового обслуговування з однорідним потоком заявок, в яку надходить випадковий потік заявок з інтервалами між сусідніми заявками, розподіленими за законом  $A(\tau)$ , а тривалість обслуговування заявок у приладі розподілена згідно із законом  $B(\tau)$ .

Процес функціонування такої системи може бути представлений у вигляді тимчасових діаграм, на основі яких можуть бути виміряні і розраховані характеристики обслуговування заявок.

Нехай  $\epsilon$  генератори випадкових чисел, що формують значення відповідних випадкових величин із заданими законами розподілів  $A(\tau)$  та  $B(\tau)$ . Тоді можна побудувати часові діаграми, що відображають процес функціонування системи, що розглядається.

На рис. 8.5 представлені чотири діаграми, що відображають:

1) «процес надходження заявок» як моментів  $t_i$  надходження заявок у систему, сформованих за правилом:

$$t_i = t_{i-1} + \tau_{ai}(t_0 = 0),$$

де  $\tau_{ai}(i = 1, 2, \dots)$  – інтервали між заявками, що надходять до системи, значення яких виробляються за допомогою генератора випадкових величин  $A(\tau)$ ;

2) «процес обслуговування в приладі», представлений у вигляді тривалостей обслуговування  $\tau_{bi}$ , які виробляються за допомогою генератора випадкових величин  $B(\tau)$  і моментів завершення обслуговування  $t'_i$  заявок в приладі, які визначаються по правилу:

$t'_i = t_i + \tau_{bi}$ , якщо на момент надходження  $i$ -ї заявки обслуговуючий прилад буде вільний;

$t'_i = t_{i-1} + \tau_{bi}$ , якщо на момент надходження  $i$ -ї заявки обслуговуючий прилад буде зайнятий обслуговуванням попередньої заявки ( $i = 1, 2, \dots; t'_0 = 0$ );

3) «модельний або реальний час», що показує дискретну (стрибкоподібну) зміну часу в реальній системі, кожен момент якого відповідає одному з наступних подій: надходження заявки в систему або завершення обслуговування заявки в приладі: відзначимо, що в ці моменти часу відбувається зміна стану системи, що описується числом заявок, що знаходяться в системі:

4) «число заявок у системі», описує стан дискретної системи та змінюється за правилом: збільшення на 1 у моменту надходження заявки в систему та зменшення на 1 у момент завершення обслуговування.

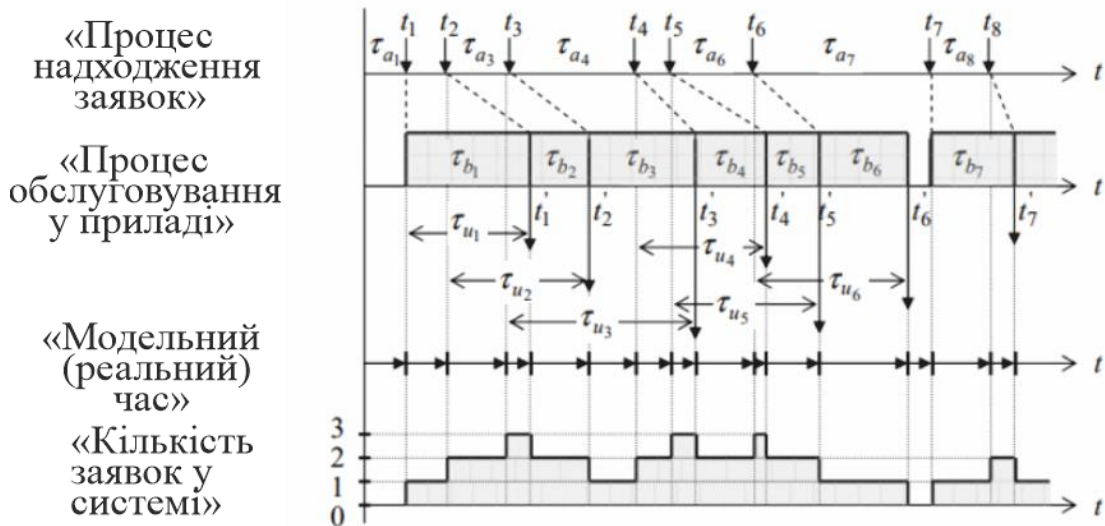


Рис 8.5. Діаграми функціонування одно каналної СМО

При дотриманні обраного тимчасового масштабу представлені діаграми дозволяють шляхом вимірювання визначити значення ймовірно-часових характеристик функціонування системи, що моделюється, зокрема, як показано на другій діаграмі, час знаходження кожної заявки в системі  $\tau_{ui}$ .

Час перебування заявок у системі – величина випадкова. Застосовуючи методи математичної статистики, можна розрахувати два перші моменти розподілу часу перебування:

- математичне очікування:

$$u = \frac{1}{N} \sum_{i=1}^N \tau_{u_i};$$

- другий початковий момент

$$u^{(2)} = \frac{1}{N-1} \sum_{i=1}^N \tau_{u_i}^2;$$

де  $N$  - кількість значень часу перебування заявок, отриманих на діаграмі, тобто кількість заявок, відображених на діаграмі як ті, що пройшли через систему і покинули її.

Звідси можуть бути отримані значення дисперсії, середньоквадратичного відхилення та коефіцієнта варіації часу перебування заявок у системі.

На основі отриманих за допомогою тимчасових діаграм значень часу перебування заявок у системі можна побудувати гістограму функції густини розподілу часу перебування.

Таким чином, імітаційна модель є алгоритмом реалізації тимчасової діаграми функціонування досліджуваної системи. Наявність вбудованих більшість алгоритмічних мов генераторів випадкових чисел значно спрощує процес реалізації імітаційної моделі на ЕОМ. Однак при цьому залишається низка проблем, які потребують вирішення. Одна з них полягає в принципі реалізації тимчасової діаграми та пов'язаного з нею завданням зміни часу в імітаційній моделі.

У найпростішому випадку тимчасова діаграма може бути реалізована наступним чином: спочатку формуються моменти надходження всіх заявок до системи, а потім кожної заявки визначаються тривалості обслуговування в приладі і формуються моменти завершення обслуговування (виходу заявок із системи). Такий підхід неприйнятний, тому що навіть для дуже простої системи доведеться зберігати в пам'яті ЕОМ одночасно мільйони значень моментів надходження та завершення обслуговування заявок, а також інших змінних, причому зі збільшенням кількості класів заявок та кількості приладів, що обслуговують, це число збільшиться у багато разів.

Другий підхід, який може бути запропонований для реалізації тимчасової діаграми - покрокова побудова діаграми. Для цього слід сформулювати змінну для модельного часу і вибрати крок  $\Delta t$  його зміни. У кожний такий час необхідно перевіряти яка подія (надходження до системи або завершення обслуговування заявки) відбулася в системі за попередній інтервал  $\Delta t$ .

Цей підхід значно скорочує потребу у пам'яті оскільки у цьому випадку у кожний момент часу необхідно зберігати в пам'яті ЕОМ значення параметрів (моментів надходження та завершення обслуговування) тільки тих заявок, які знаходяться в системі на даний момент часу.

Недолік такого підходу полягають у проблемі вибору довжини інтервалу  $\Delta t$ . З одного боку, цей інтервал повинен бути якнайменше для зменшення похибки моделювання, з іншого боку, він повинен бути якнайбільше для зменшення часу моделювання.

Найбільш ефективним визнано підхід із змінним кроком просування модельного часу, який реалізується відповідно до принципу «до найближчої події». Принцип «просування модельного часу до найближчої події» ось у чому. За всіма процесами, що паралельно протікають у досліджуваній системі, у кожний момент часу формуються моменти настання «найближчої події в майбутньому». Потім модельний час просувається до моменту настання найближчої з усіх можливих подій. Залежно від того, яка подія виявилася найближчою, викручуються ті чи інші дії. Якщо найближчою подією є надходження заявки до системи, то виконуються дії, пов'язані із заняттям приладу за умови, що він вільний, і занесення заявки на чергу, якщо прилад зайнятий. Якщо ж найближчою подією є завершення обслуговування заявки в приладі, то виконуються дії, пов'язані зі звільненням приладу та вибором обслуговування нової заявки з черги, якщо вона не порожня. Потім формується новий момент настання цієї події. На третій діаграмі «Модельний (реальний) час» просування часу відповідно до цього принципу показано у вигляді стрілок.

Для того щоб забезпечити правильну тимчасову послідовність подій в імітаційній моделі, використовується системний годинник, що зберігає значення поточного модельного часу. Зміна значення модельного часу здійснюється відповідно до принципу «перерахунку часу до найближчої події». Наприклад, якщо поточне значення модельного часу дорівнює 25, а чергові події повинні наступити в моменти часу 31, 44 і 56, значення модельного часу збільшується відразу 7 на 6 одиниць і «просувається» до значення 31. Основною одиницею часу в моделі можна вибрати будь-яку одиницю, яка дозволить отримати необхідну точність моделювання. Одиниці часу вибираються з вимог користувача до точності моделювання. Вибрана одиниця виміру повинна незмінно використовуватися у всій моделі.

Крім розглянутої служби часу, в імітаційній моделі необхідно реалізувати процедури, пов'язані з формуванням потоків заявок та імітацією обслуговування, з організацією черг заявок, з організацією збору та статистичної обробки результатів моделювання.

Таким чином, імітаційне моделювання дискретних систем зі стохастичним характером функціонування, таких як системи та мережі масового обслуговування, передбачає використання ряду типових процедур, що забезпечують реалізацію відповідних імітаційних моделей. До таких процедур насамперед належать такі процедури:

- вироблення (генерування) випадкових величин:
- формування потоків заявок та імітація обслуговування:
- організація черг заявок:
- організація служби часу:

– збір та статистична обробка результатів моделювання.

## 8.5. Моделювання в сфері ІТ

Імітаційне моделювання мережевих систем також широко використовують для прогнозування поведінки комп'ютерних мереж, виявлення можливих "вузьких місць" при тестуванні різних стратегій розподілу трафіку [23].

В даний час ринок програмного забезпечення представлений багатьма популярними системами ІМ різного класу: від простих програм, призначених для встановлення на базові ПК, до могутніх систем, що включають бібліотеки комунікаційних пристроїв, що дозволяють в значній мірі автоматизувати дослідження мережі. Але кожна із них має свої недоліки, від дуже високих цін, до вкрай нерозвиненого інтерактивного графічного інтерфейсу або потреби користувача вивчати мову написання TCL-скриптів. Це створює не тільки незручності для користувача, але й вимагає значних витрат часу на освоєння нового синтаксису. Проведемо опис та аналіз деякі системи *ІМ мереж*.

На сьогоднішній день найбільш потужною є система імітаційного моделювання мереж COMNET III, яка дозволяє точно передбачати продуктивність локальних, глобальних і корпоративних мереж. Система COMNET III працює в середовищі Windows та Unix і пропонує конструювати модель мережі за допомогою готових базових блоків. Користувач застосовує техніку drag-and-drop для графічного зображення модельованої мережі, складеної з бібліотечних елементів. Потім система COMNET III виконує детальне моделювання отриманої мережі і динамічно відображає результати у вигляді наочної мультиплікації результуючого трафіка. Після закінчення моделювання визначаються такі характеристики продуктивності мережі як:

- прогнозовані затримки між кінцевими і проміжними вузлами мережі, пропускні спроможності каналів, коефіцієнти використання сегментів буферів і процесорів;
- злети і спади трафіка як функції часу, а не як усереднені значення;
- джерела затримок і вузьких місць мережі.

*ІМ баз даних*: розробка та тестування моделей розподілу даних, їх резервного копіювання, відновлення після відмов.

На рис. 8.6 представлений приклад ходу ІМ. Вузли 0,3,4 виступають у ролі джерел повідомлень, які мають бути доставлені до вузла 2 через 1 вузол. Вузол 1 не встигає обробляти всі пакети, в результаті чого у нього сформувалася велика черга (наступним буде зелений, червоний, зелений і т. д.), а деякі повідомлення через наповненість буфера просто відкидаються.

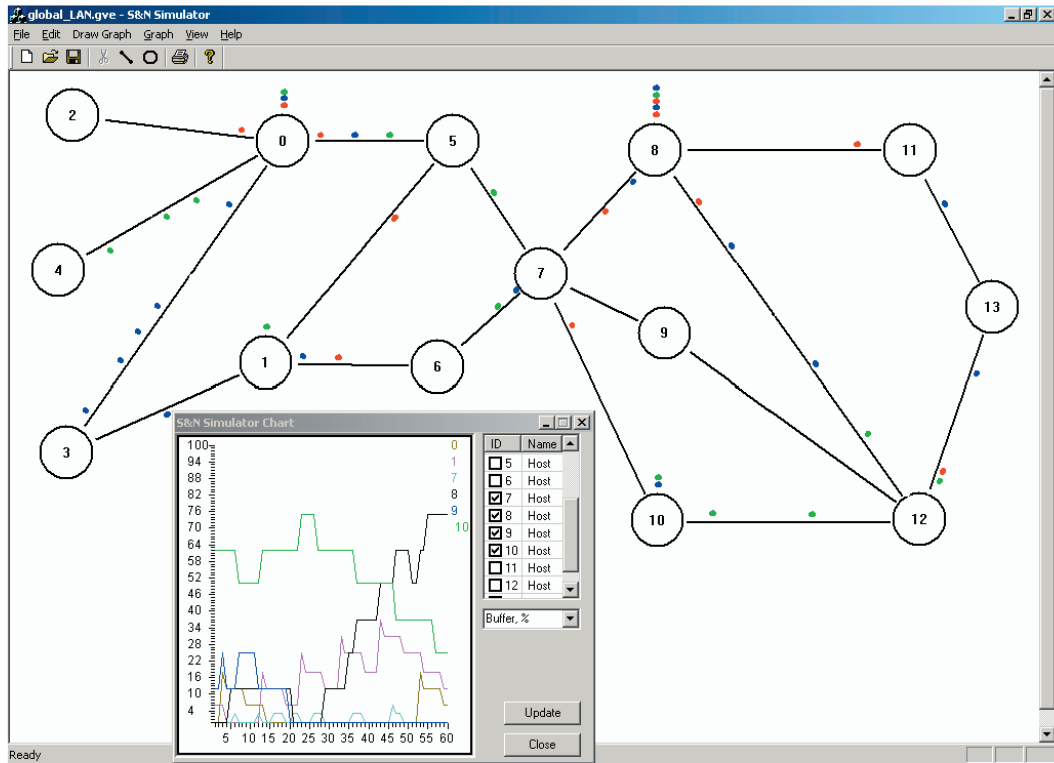


Рис. 8.6. Модель представлення комп'ютерної мережі

Параметри вузлів та ліній зв'язку задаються за допомогою діалогових вікон (рис. 8.7), які викликаються при виборі того чи іншого елемента мережі. Система дозволяє вказувати для вузла:

- ім'я вузла (елемента комп'ютерної мережі);
- час обробки пакета;
- паузу між надсиланнями пакетів;
- об'єм буфера для зберігання пакетів;
- частоту генерування повідомлень на вузлів;
- максимальний розмір повідомлення;
- колір пакета (при використанні анімації).

А для каналів зв'язку:

- ім'я каналу зв'язку;
- швидкість передачі даних;
- довжину лінії.

Зміна параметрів одного елемента мережі, впливає на функціонування системи в цілому, наприклад, зміна часу обробки пакетів у вузлах впливає на загальний час перебування пакетів в мережі.

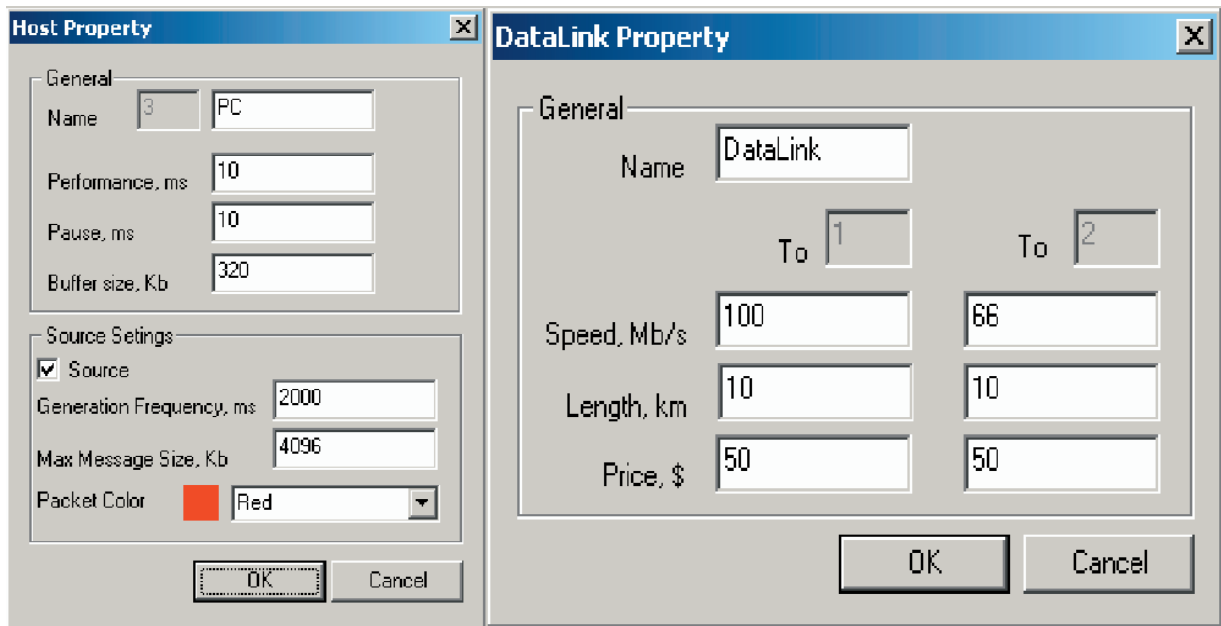


Рис. 8.7. Налаштування елементів мережі.

Дане програмне забезпечення дає можливість побачити ці зміни.

- S&N Simulator призначений для моделювання мереж з пакетною комутацією і різними методами маршрутизації пакетів, він дозволяє моделювати поведінку мережі, змінюючи:

- топологію мережі;
- спосіб маршрутизації пакетів;
- пропускні спроможності будь-якого каналу мережі;
- навантаження на мережу (інтенсивності вхідних потоків);
- довжини пакетів і розподіл числа пакетів в одному повідомленні;
- розміри пам'яті на вузлах комутації;
- обмеження на максимальний час перебування повідомлень в мережі;
- пріоритети різних повідомлень.

Система дозволяє застосовувати такі методи маршрутизації, як фіксована, лавинна, випадкова та адаптивна маршрутизація, метод Дейкстри, метод Беллмана-Форда. Більшість методів реалізовано в не рандомізованій і рандомізованій модифікаціях. Система використовує принцип розділення повідомлень на типи, що відрізняються довжинами і пріоритетами пакетів.

Анімація дає можливість спостерігати за ходом прийняття рішень вузлами під час виконання маршрутизації. Режим інтенсивного моделювання дає можливість за декілька хвилин побачити роботу мережі, яка працює на значно більшому відрізку часу.

В результаті роботи моделі одержуємо таку інформацію:

- середні затримки повідомлень різних типів;
- гістограми й функції розподілу затримки (часу доставки) повідомлень;
- гістограми щільності й функції розподілу зайняті пам'яті у вузлах комутації;

- кількість повідомлень різних типів, що дійшли до адресата;
- кількість відмов у доставці повідомлень з різних причин (брак пам'яті, перевищення допустимого часу перебування в мережі і т.д.);
- наповненість буфера прийому та кількість відкинутих пакетів. Дані опції можна побачити у вікні, представленому на рис.8.8, в якому можна вибирати для перегляду статистичної інформації елементи мережи, робота яких нас цікавить підчас моделювання.

На рис. 8.8 відображено динаміку зміни наповненості буферів елементів мережі та кількість відкинутих пакетів в часі. При побудові графіків параметри функціонування мережі знімали щосекунди.

Система дозволяє змінювати частоту опитування параметрів.

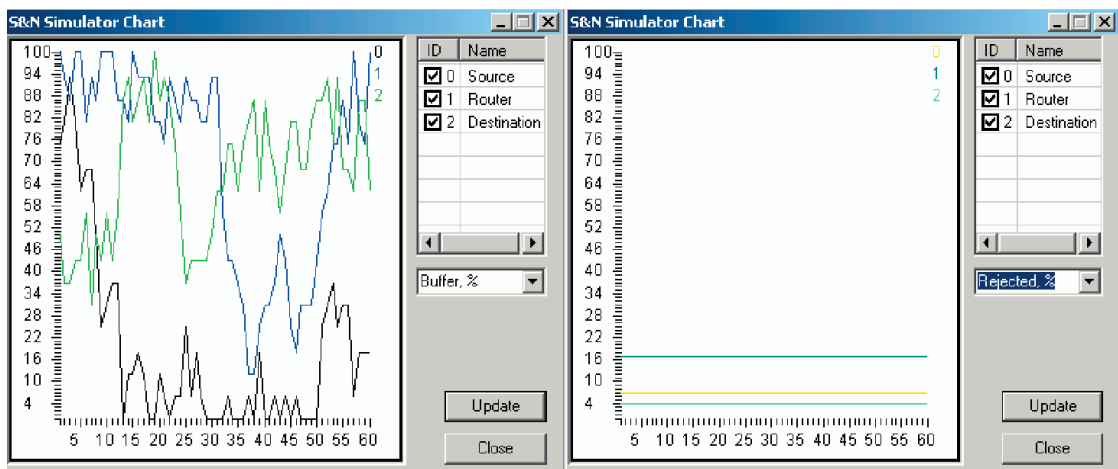


Рис. 8.8. Параметри функціонування мережі

В процесі моделювання користувач може ініціювати заповнення “журналу реєстрації подій мережі” та використовувати його для подальшого статистичного аналізу.

*Архітектура розподілених баз даних.* Архітектура системи визначає її структуру. Це означає, що компоненти системи визначені, функція кожного компонента визначена, а також визначені взаємозв'язки та взаємодії між цими компонентами. Специфікація архітектури системи вимагає ідентифікації різних модулів з їхніми інтерфейсами та взаємозв'язками з точки зору потоку даних і керування через систему.

Існує чотири «еталонні» архітектури для розподіленої СУБД: клієнт/сервер, однорангова мережа, мультибаза даних і хмарові обчислення.

*Клієнт/сервер архітектура.* Загальна ідея дуже проста: відрізнити функціональні можливості, які потрібно надати на сервері, від тих, які потрібно надати на клієнті. Це забезпечує дворівневу архітектуру, яка полегшує керування складністю сучасних СУБД і складністю розподілу.

У реляційних клієнт/серверних СУБД сервер виконує більшу частину роботи з керування даними. Це означає, що вся обробка та оптимізація запитів, керування транзакціями та керування сховищами виконуються на сервері.

Хмарні обчислення, як правило, багатовузлові (рис. 8.10), тобто складається з кількох територіально розподілених вузлів (або центрів обробки даних), кожен зі своїми ресурсами та даними.

Архітектура хмарного вузла (центру обробки даних) зазвичай є трирівневою [24].

Перший рівень складається з веб-клієнтів, які отримують доступ до хмарних веб-серверів, як правило, через маршрутизатор або балансувальник навантаження на хмарному сайті.

Другий рівень складається з веб-серверів/серверів додатків, які підтримують клієнтів і забезпечують бізнес-логіку.

Третій рівень складається з серверів баз даних. Можуть існувати інші типи серверів, наприклад, сервери кешу між серверами додатків і серверами баз даних.

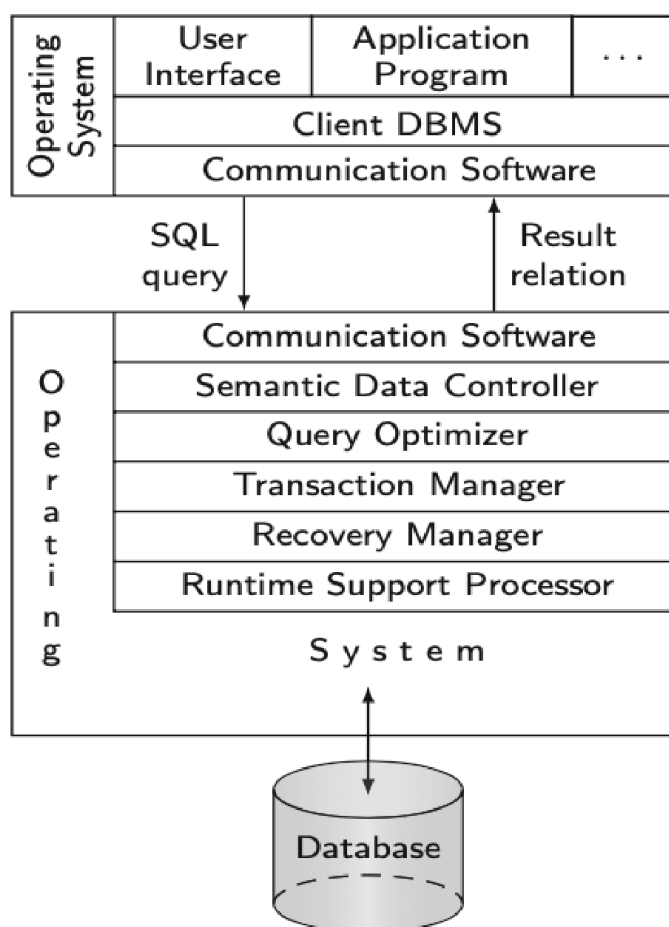


Рис. 8.9. Клієнт/сервер архітектура.

Таким чином, хмарна архітектура передбачає два рівні розподілу: географічний розподіл між вузлами, що використовують глобальну мережу та всередині сайту, розподіл між серверами, як правило, у комп'ютерному кластері.

Техніки, що використовуються на першому рівні, належать до територіально розподілених СУБД, тоді як методи, що використовуються на другому рівні, належать до паралельних СУБД.

Розглянемо розподілені бази даних систем керування комп'ютерно-інтегрованим виробництвом, у яких є множина користувачів, що відправляють свої запити для пошуку інформації. Кожен запит потрапляє до загальної черги очікування. Відомі інтенсивності надходження запитів до локальних баз та пропускна здатність каналів зв'язку. Будемо вважати, що всі запити приблизно мають однакову довжину. Час передачі запиту складає  $\tau \pm \Delta\tau$  одиниць часу.

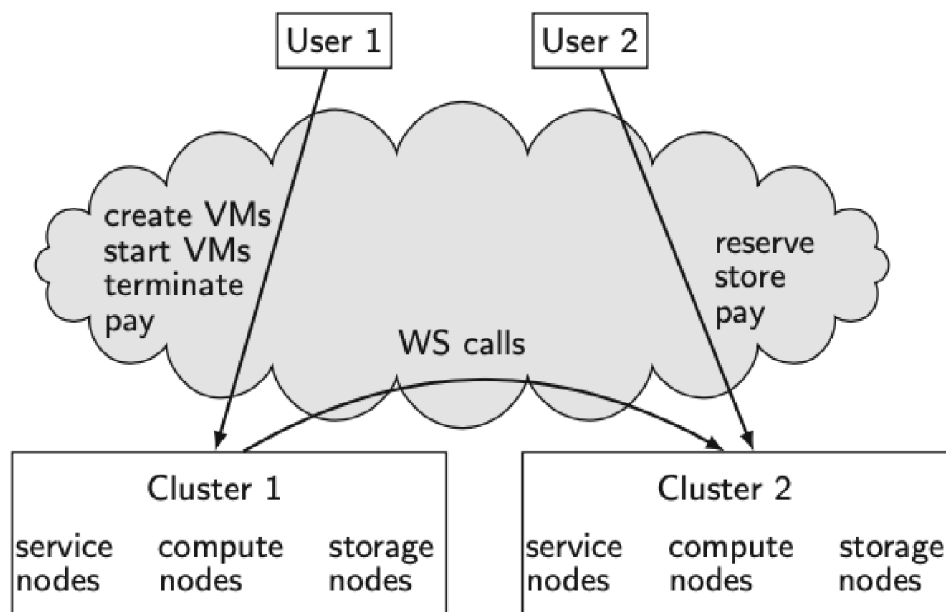


Рис. 8.10. Архітектура хмарних обчислень

Після очікування у черзі, до кожної локальної бази даних надходять запити від користувачів. Після надходження запит реєструється та очікує у черзі, до конкретної локальної бази даних. Після очікування запит надходить до локальної бази даних на обробку.

Під час обробки запиту з заданою ймовірністю, виникають ситуації, коли не вистачає даних для повної відповіді на запит, тому відбувається перенаправлення запиту до інших локальних баз даних з метою пошуку більш детальної інформації (рис. 8.11). Пошук додаткової інформації відбувається одночасно в усіх локальних базах даних. Після перенаправлення відбувається повторна обробка запиту. Якщо запит оброблено від залишає систему.

Розглядається задача оцінки часу доступу до інформаційних ресурсів (ІР), що розв'язується на етапі оптимізації РБД в процесі її проектування (реінжинірингу).

Задані:

- множина корпоративних користувачів РБД, що розміщуються у  $n$  вузлах комп'ютерної мережі; кількість ІР локальних баз  $m$  і місця їх розміщення у вузлах мережі;

- розподіл IP по вузлах мережі  $x = [x_{ij}]$  (де  $x_{ij}$  – булева змінна:  $x_{ij} = 1$ , якщо в  $i$ -му вузлу зберігається  $j$ -й IP  $t_{ij}^{qp}$ ,  $x_{ij} = 0$  – в іншому випадку;
- інтенсивності (закони розподілу та параметри) надходження запитів від користувачів до IP  $\lambda_{ij}$ ,  $i = 1, n$ ,  $j = 1, m$ ; час обробки (закони розподілу та параметри) запитів до IP  $t_{ij}$ ;
- пропускні здатності каналів мережі  $h = [h_{ij}]$ ,  $M = 1, n$ ;
- обсяги запитів користувачів до IP баз  $a_{ij}$ ,  $i = 1, n$ ,  $j = 1, m$  і відповідей на запити  $b_{ij}$ ,  $i = 1, n$ ,  $j = 1, m$ ).

Для заданого інтервалу моделювання необхідно визначити оцінки кількості запитів, що були оброблені та необроблені, та часу доступу користувачів до IP РБД  $t(x)$ . Вимоги до розроблюваного засобу моделювання розподіленої бази даних системи керування комп'ютерно-інтегрованим виробництвом можна подати у такому вигляді.

*Концептуальна модель розподіленої бази даних.* Створення розподіленої бази даних – це процес, який починається з набору вимог і призводить до визначення схеми, яка визначає набір зв'язків.

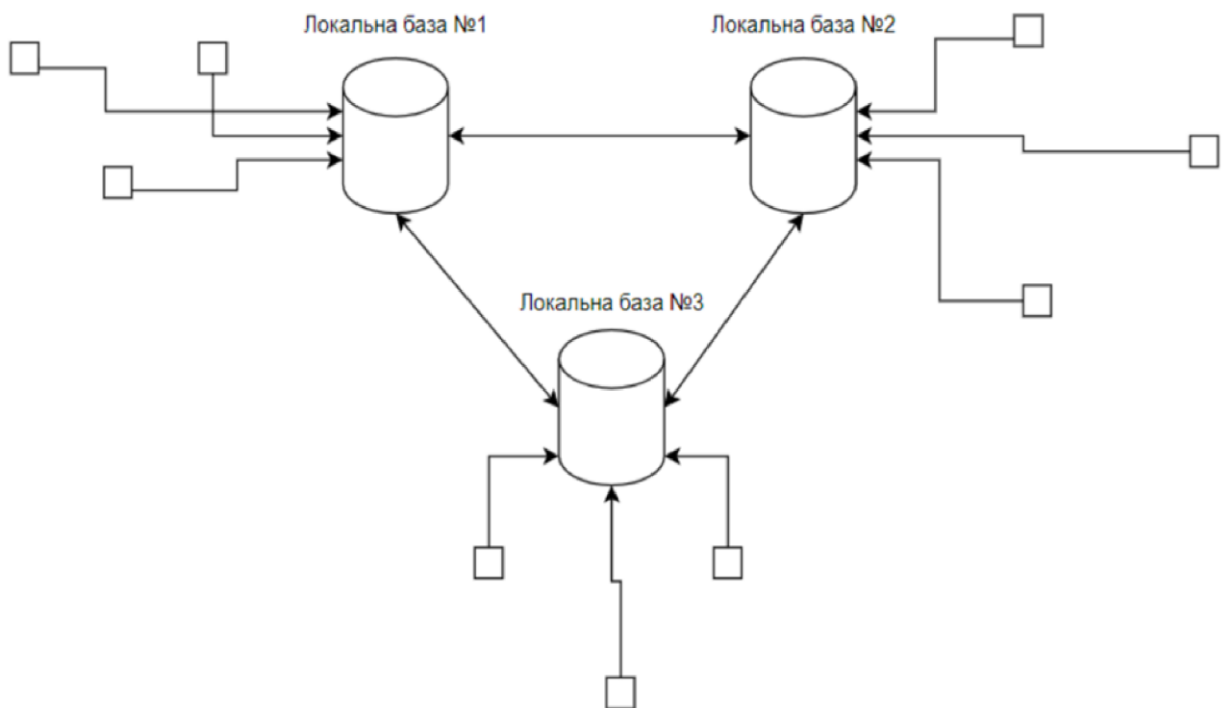


Рис. 8.11. Концептуальна модель РБД для трьох локальних баз даних

Дизайн розподілу починається з концептуальної схеми і виконує два завдання: розділення (фрагментація) і розподіл. Деякі методи поєднують ці два завдання в одному алгоритмі, а інші реалізують їх у двох окремих алгоритмах.

Існує дві основні причини та цілі фрагментації в розподілених СУБД. У першому випадку основною причиною є локальність даних. Необхідно щоб запити отримували доступ до даних на одному вузлі, аби уникнути дорогого віддаленого доступу до даних. Друга основна причина полягає в тому, що

фрагментація дає змогу одночасному виконанню кількох запитів (через паралелізм між запитами).

Фрагментація зв'язків також призводить до паралельного виконання одного запиту шляхом поділу його на набір підзапитів, які працюють над фрагментами.

Таким чином, у розподілених СУБД фрагментація може потенційно зменшити дорогий віддалений доступ до даних і підвищити паралелізм між і всередині запитів.

Фрагментація важлива для продуктивності системи, але вона також створює труднощі в розподілених СУБД. Не завжди можливо повністю локалізувати запити та транзакції для доступу до даних лише на одному вузлі – це називається розподіленими запитами та розподіленими транзакціями.

Їх обробка призводить до зниження продуктивності через, наприклад, необхідність виконання розподілених об'єднань і вартість зобов'язань розподілених транзакцій. Один із способів подолати це зниження продуктивності для запитів реалізувати тиражування даних на кількох вузлах, але це ще більше посилює накладні витрати на розподілені транзакції. Друга проблема пов'язана з семантичним контролем даних, зокрема з перевіркою цілісності. В результаті фрагментації атрибути, що беруть участь в обмеженні, можуть бути розкладені на різні фрагменти, які розподіляються на різні вузли. У цьому випадку сама перевірка цілісності передбачає розподілене виконання, що є дорогим.

Таким чином, завдання полягає в тому, щоб розділити і виділити дані таким чином, щоб більшість запитів і транзакцій користувачів були локальними для одного вузла, мінімізуючи розподілені запити та транзакції.

На рисунку 8.11 представлена концептуальна модель розподіленої бази даних для трьох локальних баз даних.

Розглянемо процес функціонування розподіленої бази даних комп'ютерно-інтегрованого виробництва. Система складається з заданої множини локальних БД. Під час роботи системи користувач надсилає запит до необхідної локальної бази даних (це відображено на рисунку 8.11). Кожен запит, що надійшов до окремої локальної бази даних обробляється, але може виникнути ситуація, що даних для відповіді не вистачає, тому відбувається обмін з іншими локальними базами даних для повної відповіді на запит.

Час обробки кожного запиту в локальній базі даних при первинній та повторній обробці визначається складністю запиту та кількістю запитів, що надійшли до системи.

*Структурна схема моделі та її опис.* Для того щоб описати процеси системи керування розподіленої бази даних використовують структурні схеми. Структурна схема є для кожного проєктованого об'єкта основним проєктним документом.

У структурній схемі управління та контролю відображаються особливості технологічного характеру даного виробництва, а також технічні

засоби, що використовуються при створенні локальних систем контролю та автоматизації.

Структурна схема моделювання розподіленої бази даних представлена на рисунку 8.12.

Об'єктом дослідження є імітаційна модель розподіленої бази даних системи керування комп'ютерно-інтегрованим виробництвом, на якій задана множина локальних баз даних та час роботи системи.

Є множина користувачів, які відправляють свої запити до системи. Кожен запит потрапляє до загальної черги очікування. Відомі інтенсивності надходження запитів до локальних баз та пропускна здатність каналів зв'язку. Будемо вважати, що всі запити приблизно мають однакову довжину. Час передачі запиту складає  $\tau \pm \Delta\tau$  одиниць часу (рис. 8.12).

Після очікування у черзі, до кожної локальної бази даних надходять запити від користувачів. Через задану певну кількість секунд, після надходження, запит реєструється та очікує у черзі, до конкретної локальної бази даних. Після очікування запит надходить до локальної бази даних на обробку.

Під час обробки запиту з заданою ймовірністю, виникають ситуації, коли не вистачає даних для повної відповіді на запит, тому відбувається перенаправлення запиту до інших локальних баз даних для пошуку більш детальної інформації. Пошук додаткової інформації відбувається одночасно в усіх локальних базах даних. Після перенаправлення відбувається обробка запиту, але вже повторна. Якщо запит оброблено від залишає систему.

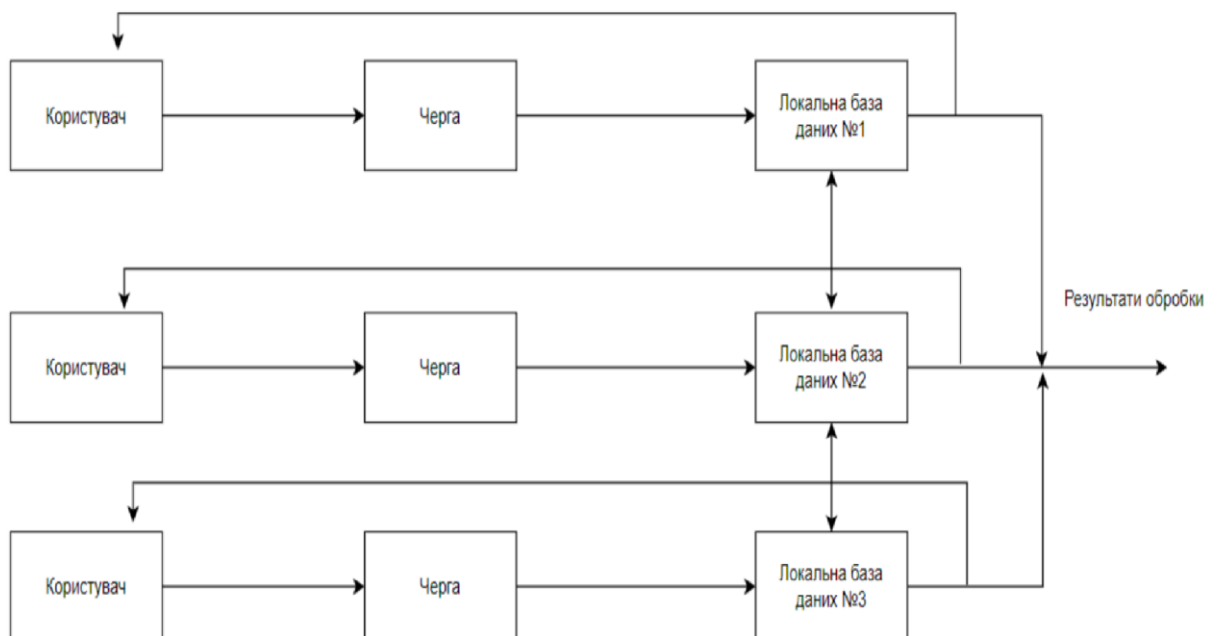


Рис. 8.12. Структурна схема РБД з трьома локальними базами

*Опис моделювального алгоритму розподіленої бази даних. Аналіз та побудова моделювального алгоритму дозволяє розв'язувати дуже велику кількість складних задач на моделювання розподіленої бази даних та*

допомагає знизити фінансові витрати та час. Моделювальний алгоритм – це ефективний, дієвий і найкращий метод, який можна використовувати для вираження рішення будь-якої проблеми в межах обмеженого простору та часу на чітко визначеній формальній мові. Для того щоб продемонструвати алгоритм можна використати блок-схеми. Починаючи з початкового стану, інструкції описують процес моделювання розподіленої бази даних, який під час виконання проходить через кінцеву кількість чітко визначених послідовних станів, зрештою створюючи «вихід» і завершуючи в кінцевому стані.

*Властивості алгоритму.* Простого написання послідовності інструкцій як алгоритму недостатньо для виконання певного завдання. Необхідно мати такі властивості, пов'язані з моделювальним алгоритмом.

*Відсутність двозначності.* Кожен крок в алгоритмі має бути однозначним. Це означає, що кожна інструкція має бути чіткою та точною. Інструкція в будь-якому алгоритмі не повинна означати будь-якого конфліктного значення. Ця властивість також свідчить про ефективність алгоритму.

*Діапазон введення.* Необхідно вказати діапазон введення. Це пояснюється тим, що зазвичай алгоритм керується введенням, і якщо діапазон введення не вказано, алгоритм може перейти в нескінченний стан.

*Кратність.* Один і той самий алгоритм можна представити кількома різними способами. Це означає, що не можна написати послідовність інструкцій простою мовою, але ми можемо написати її у формі псевдокоду. Аналогічно, для розв'язання однієї задачі ми можемо написати кілька різних алгоритмів.

*Швидкість.* Алгоритм, написаний з використанням певних ідей. Такий моделювальний алгоритм повинен бути ефективним і виробляти вихід з високою швидкістю.

*Скінченність.* Алгоритм повинен бути кінцевим. Це означає, що після виконання необхідних операцій його слід припинити.

Схема моделювального алгоритму розподіленої бази даних для трьох локальних баз даних представлена на рисунку 8.13.

Опис алгоритму роботи програми з трьома локальними базами даних/Робота кожної локальної бази даних ідентична, тому у описі моделювального алгоритму описано роботу однієї локальної бази даних [24].

1. Відбувається надходження запитів від розробника до локальної бази даних системи розподіленої бази даних.

2. Запити, що надходять до системи, потрапляють до черги, де очікують обробки. Кожна локальна база даних має свою чергу очікування.

3. Запит потрапляє до локальної бази даних для обробки.

4. У разі успішної обробки запиту локальною базою даних, до якої надійшов запит, він залишає систему РБД.

5. Оброблена відповідь запиту надходить до розробника. Також після закінчення часу роботи системи формується звіт про результати обробки.

6. Якщо локальна база даних, до якої надійшов запит, не може дати повну відповідь, відбувається обмін запиту між іншими локальними базами даних для повної відповіді на запит.

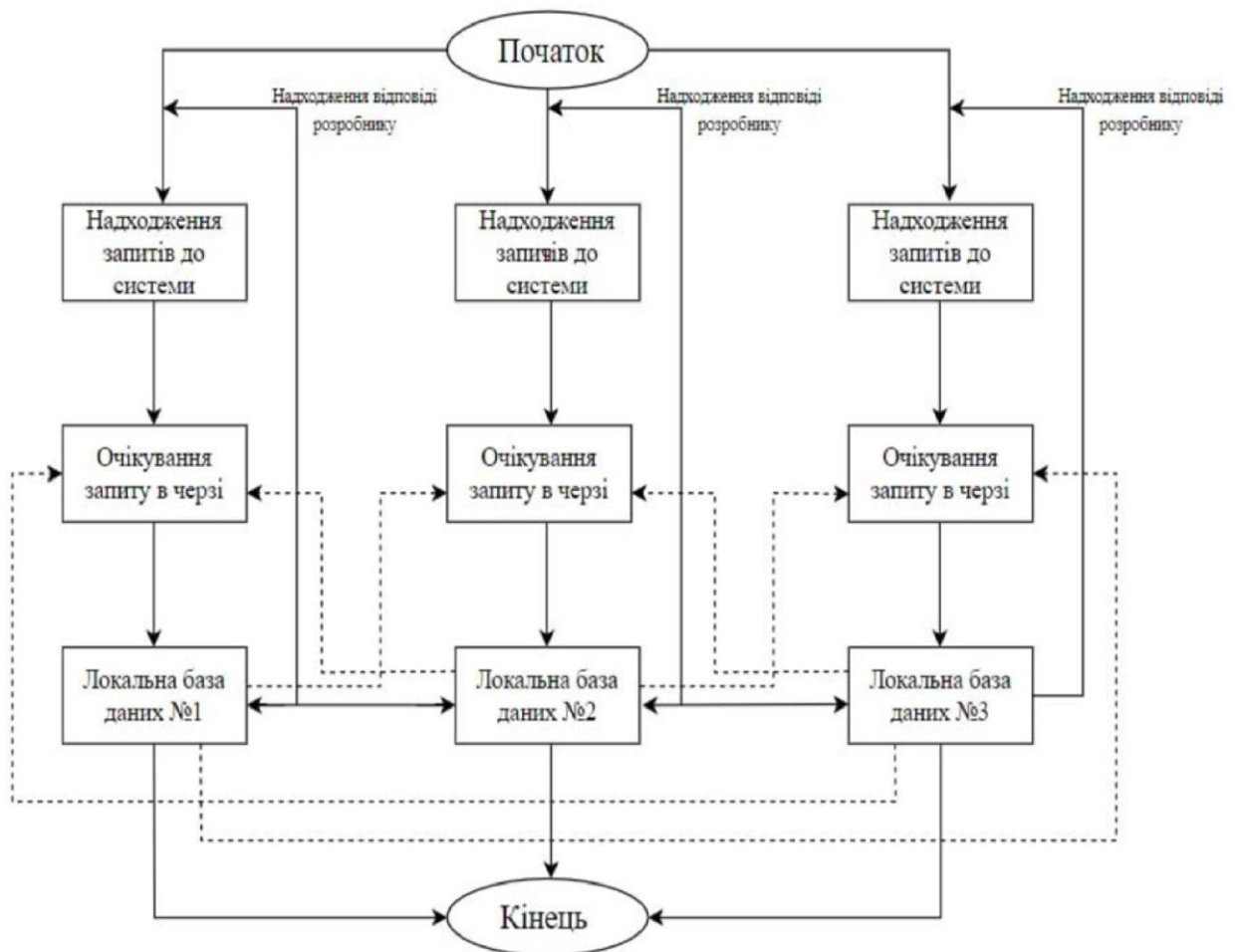


Рис. 8.13. Схема моделювального алгоритму РБД для трьох локальних баз

7. Запит, якому необхідно здійснити обмін, надходить до черги очікування у іншій локальній БД.

8. Після обслуговування запиту, для якого відбувся обмін залишає систему РБД, а відповідь так само надходить до розробника і формується звіт по закінченню часу роботи системи.

На основі аналізу процесу розподілу запитів в РБД сформульовано постановку задачі її моделювання. Виходячи з дискретності та стохастичності процесів функціонування РБД, запропоновано розглядати їх як багатофазні системи масового обслуговування. Це дозволило обрати для їх моделювання дискретно-стохастичний підхід.

Таким чином, імітаційне моделювання є потужним інструментом для аналізу, розробки та оптимізації складних систем у різних сферах діяльності. Використання різних методів моделювання, таких як дискретне, континуальне та агентне, дозволяє з'ясувати, як система функціонує і як вона

реагує на зміни. Завдяки моделюванню можна не тільки покращити систему, але й передбачити можливі проблеми, оптимізувати ресурси і знизити ризики.

### **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Перелічіть послідовні етапи процесу імітаційного моделювання - від формулювання задачі до аналізу результатів. Які завдання вирішують верифікація і валідація моделі?

2. Що таке концептуальна модель об'єкта моделювання? Назвіть ключові елементи концептуальної моделі (суб'єкти, ресурси, події, правила переходів).

3. Опишіть принципи організації імітаційного експерименту: warm-up, число реплікацій, довжина прогонів, управління випадковістю. Як визначити достатню кількість прогонів?

4. Які особливості має імітаційне моделювання у сфері ІТ (моделювання мереж, серверів, навантаження, виявлення «вузьких місць»)? Наведіть приклад сценарію експерименту.

5. Спроектуйте експеримент (план експерименту) для перевірки гіпотези «збільшення кількості серверів на 20% знижує середній час відгуку на 15%» у моделі мережевого сервісу; вкажіть змінні, фактори, кількість прогонів і статистичний критерій перевірки

## **Тема 9. Оцінка якості програмних систем**

### **9.1. Оцінка якості програмних систем: метрики якості ПЗ**

Оцінка якості програмного забезпечення (ПЗ) є важливим етапом у розробці та підтримці програмних продуктів. Вона дозволяє визначити, наскільки система відповідає вимогам замовника, наскільки ефективно працює в реальних умовах, та наскільки вона стабільна та надійна.

Процес оцінки якості нерозривно пов'язаний з визначенням параметрів, які піддаються виміру, у яких зацікавлений споживач. Сучасні методи оцінки якості мають у своєму розпорядженні широкий арсенал засобів вимірювання, серед яких особливо виділяють прикладний статистичний аналіз даних, методи експертних оцінок, а також складніші методи інтелектуального аналізу даних, таких як нейронні мережі, методи машинного навчання тощо.

В галузі аналізу якості програмного забезпечення особливе місце займає спеціальний науковий напрямок – програмометрика, що має основою метричну теорію програм [25].

Найбільш характерними рисами математичного апарату, що використовується для цих цілей, є наступні:

- Завдання вимірів спираються на потреби предметної області, а чи не власне математики.

- Використовуваному математичному апарату притаманні різні припущення, що часом розходяться зі строгим апаратом математичної статистики.
- Значна увага приділяється збору даних для аналізу та передобробки даних, що надходять (з метою виявлення порушень форматів даних, аномальних значень, пропусків тощо).

У галузі метрик оцінки якості програмного забезпечення також виділяють такі підходи, як:

- Метрики оцінки безпосередньо ПЗ – на основі підходів, що відображають специфічні характеристики програм.
- Метрики аналізу технології розробки ПЗ – на основі підходів до оцінки параметрів розробки програм та їх технічних характеристик.

Існують також метрики, що використовуються при оцінці якості ПЗ виходячи з виду інформації, що надходить:

- Метрики відповідності розробленого програмного забезпечення задалегідь визначеним вимогам.
- Метрики, що дозволяють аналізувати відхилення норми параметрів первинних проектних матеріалів (визначають повноту технічних параметрів вихідного коду).
- Метрики, що дозволяють прогнозувати якість реалізованого програмного забезпечення.

Застосування різних засобів оцінки якості має особливості. Зокрема, використання математичних методів націлене на те, щоб підвищити об'єктивність в оцінці якості, уникаючи інтуїтивності. Однак практика показує, що ефективність оцінки якості зростає при використанні творчої, інтуїтивної діяльності окремих експертів або колективів експертів, за допомогою яких досягається побудова найбільш адекватних і повних моделей процесів, якість яких вимірюється.

Експертними називають загальну сукупність методів, що ґрунтуються на використанні кваліфікації, досвіду, творчого потенціалу одного або кількох фахівців вимірюваного процесу.

На реалізацію методів експертного оцінювання істотно впливають такі фактори:

- Фахівці для проведення експертизи повинні мати достатню професійну кваліфікацію та бути відповідним чином підібрані.
- Експерти повинні визначати свої оцінки незалежно від зовнішніх впливів та один від одного.
- Число залучених експертів має бути обґрунтованим: збільшення кількості недостатньо компетентних фахівців не призведе до підвищення якості прийнятого рішення, зокрема відсутність кваліфікації не можна компенсувати кількістю залучених людей.

- Зменшення числа експертів у групі може призвести до збільшення кількості випадкових факторів, упередженості, суб'єктивних оцінок.

Вимірювальні шкали. Необхідність в об'єктивності та відтворюваності вимірювань та кількісної оцінки різних характеристик програмного забезпечення потребує використання певної системи вимірювань та методів оцінки.

Системою вимірювань характеристик ПЗ називають сукупність характеристик, які піддаються вимірюванню, вимірювальних шкал, характерних для них, одиниць їх вимірювання та визначених між ними зв'язків.

Існують різні шкали вимірювань реєстрованих характеристик об'єктів. Розглянемо найпоширеніші їх, представлені рисунку 9.1:



Рис 9.1. Шкали вимірювань

З характеристик і специфіки використовуваних метрик їм застосовують такі шкали:

- номінальна шкала (шкала найменувань), у разі якої всі об'єкти поділяються на класи за тією самою ознакою, а об'єкти з різних класів мають різні позначення. Номінальна шкала характеризується обмеженими можливостями статистичної обробки змінних, багато операцій якої (визначення середнього арифметичного, медіани) для неї не мають сенсу.
- порядкова (рангова) шкала, у разі якої якісні змінні розбиваються на класи, та й самі класи впорядковуються.

Важливою характеристикою порядкової шкали є те, що емпірична значимість змінних, виміряних у ній, пов'язані з різницею між сусідніми значеннями, мають чисельні уявлення. Іншими словами, незважаючи на те, що різниця між «помірним» використанням процедур у програмі та відсутністю їх дорівнює одиниці, так само, як і різниця між «частим» та «помірним» їх використанням, не можна фіксувати, що різниця між ними однакова. Для змінних даної шкали неприпустимі арифметичні операції, проте можливе використання медіани та моди, будь-яких монотонних перетворень, процедур перевірки гіпотези рівності медіани заданому значенню, гіпотези перевірки рівності двох медіан, методів рангової кореляції Кендалла та Спірмена;

- інтервальна шкала, коли він емпіричну значимість мають як абсолютні значення показників, а й різниця (інтервал) з-поміж них.
- шкала відносин – шкала, у якій чисельні значення числової системи визначаються з точністю до перетворення подоби (у разі інтервальної шкали – з точністю до лінійних перетворень). Емпіричну значимість тут мають відносини між значеннями.

Властивості шкали інтервалів тут зберігаються. До цієї шкали належать всі інтервальні змінні, котрим фіксована абсолютна нульова точка. Практика показує, що різниця між змінними, що належать до шкали відносин та інтервальної шкали, зазвичай невелика; змінні, визначені до шкали інтервалів, мають також шкалу відносин.

*Категорії показників якості програмного забезпечення.* В даний час існує велика кількість метрик, які застосовуються для оцінки програмного забезпечення. Усі дані метрики можуть бути класифіковані з різних підстав. Запропонована нижче класифікація виконана за напрямками якісної оцінки метрик:

- Аналіз надійності ПЗ, що дає можливість аналізувати ситуації відмов та прогнозувати їх.
- Аналіз мовних засобів, їх рівня та використання.
- Аналіз продуктивності ПЗ, у тому числі шляхом визначення помилок реалізації, з подальшим підвищенням її ефективності.
- Аналіз складності ПЗ, у тому числі, інформаційної та топологічної.
- Аналіз складності сприйняття ПЗ, у тому числі, когнітивної ергономічності, що має в основі психологічні особливості сприйняття інформації людиною, результати якої мають цінність на етапі розробки, проектування, а також внесення змін та супроводу ПЗ.
- Аналіз праці розробника, необхідний для техніко-економічного обґрунтування проекту з розробки програмного забезпечення, а також для прогнозування термінів всього проекту та окремо складових його етапів.

З точки зору можливості вимірювання характеристик та їх точності всі атрибути якості можуть бути розбиті на 3 категорії показників, що мають свої особливості. Зведена таблиця категорій показників з відповідними шкалами вимірювань і прикладами представлена нижче в таблиці 9.1.

Для кількісних показників більш характерним є використання інтервальної або відносної шкал, оскільки вони можуть бути об'єктивно вимірними та зіставленими з вимогами у числовому еквіваленті. Категорійні дають інформацію про сукупність властивостей та загальних ознак об'єкта, тим самим описуються номінальною шкалою категорій-властивостей. Якісні показники, як правило, визначаються та оцінюються суб'єктивно із залученням експертів та містять окремі ознаки – категорії, для опису яких

застосовують порядкову чи номінальну шкалу категорій. Частина атрибутів якісних показників можуть бути описані кількісно, проте частіше формуються та аналізуються експертно.

Кожна математична модель метрики спрямована на кількісний аналіз того чи іншого параметра (характеристики, субхарактеристики) ПЗ, сукупність яких може бути класифікована за типом одержуваної інформації про метрики і за типом використовуваної інформації про ПЗ:

- за типом отримуваної інформації про метрики ПЗ: визначальні відхилення від заданої норми показників;
- прогножуючі значення показників;
- визначальні факт відповідності ПЗ сформованим вимогам;
- за типом використовуваної інформації про:
  - метрики, що мають в основі лексичний аналіз ПЗ (метрики Холстеда, Джилба, Чепіна і т.д.);
  - метрики, що базуються на оцінці потоків керування (метрика Маккейба);
  - метрики, що оцінюють міжмодульні та внутрішньомодульні зв'язки (найважливіші характеристики складності ПЗ на етапі проєктування);
  - метрики, що аналізують потоки даних.

Таблиця 9.1 - Категорії показників

	Категорії показників		
	Кількісні	Якісні	Категорійні
Шкала вимірів	Інтервальна відносна	Порядкова номінальна	Номінальна шкала категорій- властивостей
Приклади показників якості	Надійність, ефективність комплексів програм	Мобільність, супроводжуваність, трудомісткість, практичність	Функціональна придатність програмного засобу

Основною ідеєю всіх перерахованих 4 груп є визначення коректності і складності ПЗ, що розробляється, а також прогнозування параметрів і характеристик майбутнього ПЗ.

## 9.2. Основні показники якості програмного забезпечення

*Поняття алгоритмічної складності.* Складність відіграє у технічних науках, у своїй має часом різні значення. З точки зору математики, складність безпосередньо пов'язана з текстом опису об'єкта і визначається правилом «чим довший текст опису об'єкта, тим він складніший». Очевидно, що за такої постановки ключову роль грає вибір способу опису об'єкта [26].

Виходячи з підходу до розгляду об'єктів як деяких послідовностей символів даного алфавіту, найкращим з погляду економії буде алгоритмічний спосіб їхнього опису. Тоді якщо позначити як деяку частково-рекурсивну функцію, отримаємо міру складності послідовності  $x$ :

$$\min l(p): \varphi(p) = x$$

$$K_{\varphi} = \infty, \text{ якщо } \forall p \in s \varphi(p) \neq x,$$

де  $p$  - код, яким  $\varphi$  відновлює послідовність  $x$ ;  $l(p)$  - довжина коду (кількість двійкових розрядів), а  $s$  - набір всіх допустимих програм.

У програмометриці широко поширена формулювання міри складності деякої послідовності символів з певного алфавіту як довжини (числа двійкових розрядів) найбільш короткої програми, яка генерує таку послідовність.

Розглянемо основні властивості міри складності:  $K_{\varphi} \leq l(x)$ , тобто складність  $x$  не перевищує довжини послідовності  $x$ :

2.  $\lim_{x \rightarrow \infty} K_{\varphi} = \infty$  тобто. складність послідовності символів  $x$  зростає із зростанням довжини послідовності необмежено;

3. майже всі послідовності випадкові, тобто.

4. алгоритмічна складність довільної послідовності символів не перевищує ентропії послідовності:

$$K_{\varphi} \leq -N \sum_{i=1}^s P_i \log P_i$$

Ентропія послідовності може бути прорахована шляхом обчислення ймовірностей-частот символів у даній послідовності:

$$P_1 = m_1/N; P_2 = m_2/N;$$

В уточненому вигляді нерівність алгоритмічної складності даної послідовності символів може бути подана як:

$$K_{\varphi} < s \log 2s.$$

*Імовірнісна модель тексту програми.* Дослідження квантитативної лінгвістики дозволили виявити низку закономірностей, зокрема, емпіричних, широко відомих нині. Наприклад, закон Ципфа встановлює співвідношення між частотою появи слів у тексті (виділених із словника цього тексту) і довжиною цього тексту. Подібні результати були отримані М.М. Холстедом під час аналізу текстів програм, реалізованих кількох алгоритмічних мовами. Їм було виявлено емпіричне співвідношення загальної кількості слів програми та його величини у словнику, надалі теоретично обґрунтовані з властивостей складності алгоритмічної теорії.

Основні умови, яким мають задовольняти тексти програм, словник яких складається лише з операндів та імен операторів, наведено нижче:

– виникнення операнда чи імені оператора поспіль багато разів мало ймовірно;

– повторення будь-якої групи операндів чи операторів багато разів виключається шляхом циклічної організації програм;

– при використанні процедур та функцій у тексті програми присутні лише імена процедур та функцій без безпосереднього повторення цілих блоків програм періодично;

– якщо ім'я операнда оголошено у програмі, хоча б раз воно в ній з'явиться.

Тим самим, якщо розглядати будь-яку програму як результат стиснення розгортки (шляхом використання процедур і функцій, циклів), можна дійти невтішного висновку у тому, що довжина програми є мірою складності її розгортки, і всі властивості розгортки істотно впливають на скорочення тексту програми.

*Математичне очікування та дисперсія довжини тексту програм.* Якщо текст програми розглядати без урахування його семантики, лише як випадкову послідовність операндів та операторів, які утворюють словник програми, актуальною стає завдання використання певного генератора випадкових послідовностей для написання тексту цієї програми. Ця мета може бути описана як формування вибірки з цієї генеральної сукупності із поверненням. Якщо прийняти за  $L_r$  кількість витягів із генеральної сукупності, що йдуть за фіксацією  $r$ -ого імені оператора до вибору наступного  $(r + 1)$ -ого включно, то має місце співвідношення, що дає інформацію про вибірку, яка вичерпала генеральну сукупність (у нашому випадку – словник програми):

$$Q_1 = 1 + L_1 + L_2 \dots$$

Розподіл  $L_r$  відповідатиме розподілу номера першого успіху з ймовірністю, що дорівнює:

$$P = \eta - r / \eta$$

виходячи з чого математичне очікування представляється у вигляді:

$$M(L_r) = \eta / \eta - r$$

Використовуючи теорію складання математичних очікувань, властивостей сум гармонійних рядів та переходячи до двійкових логарифмів, маємо:

$$M(Q) \approx \ln 2 \cdot \eta \log 2 \eta \approx \eta \log \eta$$

Це співвідношення є співвідношенням Холстеда, яке дає інформацію про математичне очікування числа слів деякої програми (її довжини), якщо словник програми становлять слів (операндів і операторів).

Шляхом багатовимірного статистичного аналізу це співвідношення було уточнено, і було визначено залежність із рівнем значущості:  $\pm 3 \div 6\%$ :

$$N = 0,91 \eta \log \eta$$

Тим самим, вже на стадії визначення завдань можуть бути оцінені кількість імен вхідних змінних, кількість імен вихідних змінних, операції з якими виконуватиме проектоване програмне забезпечення.

Внаслідок чого на стадії визначення завдань може бути прорахована та оцінена довжина (кількість слів) проектованого ПЗ, що безпосередньо пов'язане з етапом визначення трудомісткості проекту та деяких додаткових характеристик.

Якщо припустити, що програму можна розбити на модулі, всі з яких у програмі рівні, то кількість слів одного модуля у словнику визначатиметься як  $\eta / k$ , а для довжини  $N_M$  та дисперсії  $D(Q_M)$  модуля виведені співвідношення:

$$N_M = \eta / k \log \eta / k; D(Q_M) = \pi^2 \eta^2 / 6k^2;$$

Використовуючи теорему додавання, для цілої програми отримуємо:

$$N = kN_M = \eta \log \eta / k; D(Q) = kD(Q_M) = \pi^2 \eta^2 / 6k^2;$$

Оцінити точність розглянутого співвідношення Холстеда можна шляхом розрахунку відношення абсолютного розкиду довжини програми до її математичного очікування, а саме:

$$\delta = \frac{\sqrt{D(Q)}}{M(Q)} = \frac{\sqrt{\pi^2 \eta^2}}{6k}; \eta \log \frac{\eta}{k} = \frac{\pi}{\sqrt{6k}} \frac{1}{\log \frac{\eta}{k}}.$$

Тим самим можна зробити висновок про те, що співвідношення Холстеда буде обчислюватися точніше зі зростанням довжини програми.

*Метричні характеристики програм.* Характеристика «довжина програми» безпосередньо з визначенням інших найважливіших її характеристик.

Одна з них – обсяг програми, для визначення якої важлива кількість двійкових розрядів, а не слів. Якщо словник складається з  $\eta$  слів, то для вирішення проблеми завдання номера кожного зі слів необхідно мінімум  $\log \eta$  біт. Тоді обсяг програми може бути розрахований як:

$$V = N \log \eta = \eta \log^2 \eta$$

Далі, уточнимо отримані вище співвідношення шляхом поділу кількості операторів та операндів, позначивши їх як  $\eta_1$  і  $\eta_2$  відповідно. Тоді співвідношення Холстеда можна переписати у вигляді:

$$N = \eta_1 \log \eta + \eta_2 \log \eta \approx \eta_1 \log \eta_1 + \eta_2 \log \eta_2 = N_1 + N_2$$

Зважаючи на приблизну взаємно-однозначну відповідність між кількістю операторів і операндів (оскільки кожен операнд з'являється в тексті програми хоча б з одним операндом), приходимо до важливого для практики висновку:

$$N \approx 2N_2 = 2 \eta_2 \log \eta_2$$

Якщо ввести додатковий параметр  $\eta_2^*$  – розмір генеральної сукупності імен вхідних-вихідних змінних, отримуємо:

$$\eta_2 \approx \eta_2^* \log \eta_2^*$$

Для мінімально можливого обсягу програми виведено співвідношення:

$$V^* = (\eta_2^* + 2) \log (\eta_2^* + 2)$$

Виходячи з нього, може бути розрахований рівень реалізації програми:

$$L = V^*/V$$

Цей важливий показник дає інформацію про економічність використання засобів мовлення, виявляє ступінь компактності програми. Чим ближче значення  $L$  до одиниці, тим досконалішою вважається програма.

Дані показники можуть бути уточнені при вирішенні проблем оптимізації кількості модулів та їх довжини у програмі. Завдання структуризації програмного засобу на етапі проектування носить змістовний

характер і може бути формалізована, тим щонайменше, розрахунок цих метричних показників важливо для визначення найкращих властивостей його структури. Розрахунок числа та довжини модулів та інших подібних характеристик дозволяє визначити оптимальні з погляду якості проєкту параметри структури проєктованого програмного засобу.

*Кількісна оцінка програмування.* Об'єктивною оцінкою роботи програмування може стати число порівнянь вибірок описаної вище моделі цього процесу. Відповідно до теорії сортування, мінімальна кількість порівнянь у процесі пошуку елемента масиву буває при дихотомічній вибірці (шкала кодується 2 значеннями, 0 або 1, що взаємно виключають) і дорівнює  $\log m$ .

Нехай  $N$  – довжина програми,  $\eta$  – її словник, тоді робота програмування ( $E$ , загальна кількість вибірок) дорівнюватиме  $N \log \eta$ . При цьому враховується рівень реалізації програми  $L$ : число вибірок збільшується  $1/L$  разів.

Тоді робота програмування може бути розрахована за такою формулою:

$$E = N \log \eta / L = V / L = V^2 / V^*$$

оскільки

$$L = V^* / V$$

Скориставшись визначенням «числа Страуда» (що характеризує кількість уявних порівнянь, які виробляє індивід у секунду) і визначенням кваліфікаційного часу програмування ( $T$ ) як відношення роботи до Страуда, отримуємо:

$$L = E / S = V^2 / SV^*$$

Для розрахунку календарного часу програмування ( $T_k$ ) скористаємося формулою розрахунку кількості команд ( $C$ ):

$$C = 3/8 N,$$

де  $3/8$  – коефіцієнт перерахунку Батіга.

Тоді:  $T_k = 3N / 8vn$  (днів)

Додаткові відомості щодо роботи програмування може дати введена Холстедом метрика оцінки рівня мови програмування:  $\lambda = L_2 V$

У таблиці 9.1 наведено значення рівня мови програмування для деяких широко використовуваних мов, представлених у таблиці 9.2:

Таблиця 9.2 – Рівні мов програмування

Мова	$\lambda$
Бейсик	1,22
Паскаль	1,25
Асемблер	0,88
Фортран	1,14
Природна мова	2,16

Для оцінки співвідношення двох мов програмування вводять оцінку відношення їхнього кваліфікаційного часу, яка виглядає наступним чином:

$$T_1 / T_2 = \lambda_1^2 / \lambda_2^2$$

Таким чином, на підвищення продуктивності процесу програмування може вплинути і процес підвищення рівня алгоритмічних мов.

Метрика Чепіна. Крім метрик Холстеда, розглянутих у цьому розділі, необхідно приділити також увагу метриці Чепіна – заходам визначення проблеми розуміння програм виходячи з вхідних і вихідних даних.

Базовий варіант цієї метрики, визнаний найефективнішим з погляду практичного застосування, включає у собі 4 функціональні групи числа змінних, формують список вводу-виводу:

*P* – Змінні, що використовуються для розрахунків і для виведення.

*M* – Змінні, які створюються всередині програми (модифіковані).

*Z* – керуючі змінні, призначення яких полягає у управлінні функціонування програмного модуля.

*T* – так звані «паразитні» змінні (які не використовуються в програмі). Змінні такого типу не відіграють вирішальної ролі для поставленого головного завдання, проте можуть бути використані для реалізації проміжних дій.

Характерною особливістю метрики Чепіна є необхідність урахування всіх змінних у всіх функціональних групах, оскільки та чи інша змінна може бути використана відразу для реалізації кількох функцій.

Метрика Чепіна в базовому варіанті виглядає так:

$$Q = a \cdot P + b \cdot M + c \cdot C + d \cdot T,$$

де *a*, *b*, *c*, *d* – вагові коефіцієнти.

Вагові коефіцієнти у цій формулі вводяться для нормалізації різного впливу тієї чи іншої групи змінних на складність програми.

Автор метрики надає ваговим коефіцієнтам такі значення:

$$a = 1, b = 2, c = 3, d = 0,5.$$

Найменший ваговий коефіцієнт відноситься до групи *T* не випадково: «паразитні» змінні часто ускладнюють розуміння програми, при цьому не збільшуючи складність потоку даних.

Варто згадати, що метрика Чепіна була визначена шляхом аналізу вихідних текстів програм, тим самим є ще один підхід до їх автоматизованого аналізу.

### 9.3. Методи тестування, верифікації та валідації програмних систем

Процес тестування програмного забезпечення, поряд з процесом розробки, є основним в діяльності будь-якої компанії, що веде діяльність по створенню інформаційних систем. очікування потенційних користувачів до конкретних характеристик програмної системи, тому можуть використовуватися для визначення критеріїв успішності виконання тестів. Отже, програмне рішення, що тестується, може мати атрибут якості, якому притаманний розглянутий показник [27].

Як було зазначено, з урахуванням вимог формується реалізація системи як програмних модулів (коду мовою програмування), які у подальшому необхідно протестувати. У цьому випадку під процесом тестування слід розуміти комплекс процедур, що дозволяє визначити некоректну і незаплановану поведінку системи та її компонентів. Далі, за умови успішного виконання всіх запланованих тестів, слід провести верифікацію системи. Верифікація дозволить визначити правильність роботи системи та ґрунтується на функціональних сценаріях. Якщо система виконує всі покладені неї функції правильно, отже можна говорити про успішну верифікації. Останнім етапів у процесі перевірки системи, що розробляється, є валідація. Валідація дозволяє визначити, чи система відповідає початковим вимогам потенційних користувачів (чи вдалося реалізувати те, що було потрібно). Іншими словами, чи дозволяє система досягати тих результатів, яких від неї очікують. Розглянута схема наведена на рисунку рис 9.2.

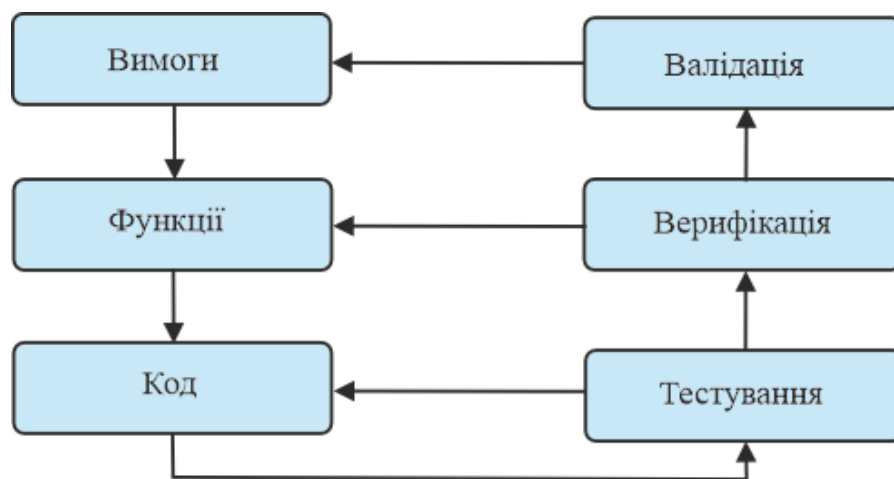


Рис 9.2. Місце тестування у процесі розробки

*Модель програмної помилки.* Будь-яку некоректну поведінку програмної системи можна описати за допомогою взаємозв'язку трьох понять: fault; error; failure.

Термін fault (несправність) означає деяку програмну невідповідність у коді, наприклад, неправильна конкатенація рядків, при якій порядок

операндів помилковий, і фінальний рядок виявляється не таким, яким його очікують бачити.

Термін Error (помилка) визначає стан програмної системи, у якому проявляється Fault. Слід зазначити, що Fault не завжди може призвести до хибного стану системи і в такому разі вважається латентною помилкою (Latent Error). Це з тим, що помилковий модуль може використовуватися у звичайних сценаріях роботи користувачів, у більшості випадків не порушує процес функціонування. Прикладом Error можна вважати виконання операції конкатенації рядків та запис результату змінну.

У свою чергу, під терміном failure (відмова) розуміється відхилення у нормальному, передбаченому режимі функціонування, що призводить до порушення очікувань користувачів. Прикладом може бути неправильний формат номера, в якому код міжнародного зв'язку знаходиться в кінці, а не на початку, як цього вимагають правила.

Модель програмної помилки показано рисунку рис 9.3.

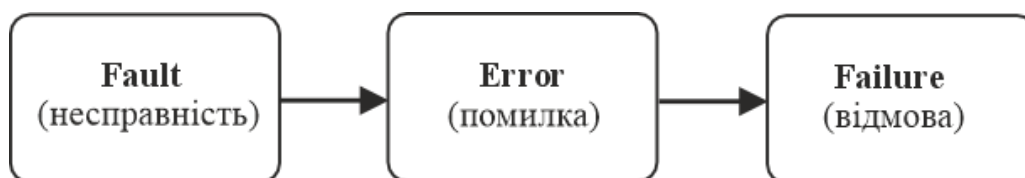


Рис 9.3. Модель програмної помилки

*Властивості тестів.* Відповідно до моделі програмної помилки необхідно коректно планувати та проводити розробку тестів. Для того, щоб тест можна було вважати правильно складеним, необхідно щоб він відповідав трьом основним властивостям:

- Reachability – тест повинен виконати місце у вихідному коді, де є програмна помилка;
- Corruption – під час помилки стан програми має зіпсуватися з появою збою;
- Propagation – збій має поширитися далі та викликати невдачу у роботі програми.

Забезпечення цих властивостей – одна із найскладніших задач тестування програмного забезпечення, заради вирішення якої великі компанії виділяють серйозні фінансові активи для їх відповідності.

*Класифікація методів тестування.* Існує велика кількість методів тестування. Класифікація основних із них наведена нижче:

За знанням осередку системи:

- чорна скринька (black box testing);
- сірий ящик (grey box testing);
- білий ящик (white box testing).

По об'єкту тестування:

- функціональне тестування (functional testing);
- тестування інтерфейсу користувача (UI Testing);
- тестування локалізації (localization testing);
- тестування швидкості та надійності (load/stress/performance testing);
- тестування безпеки (security testing);
- тестування досвіду користувача (usability testing);
- тестування сумісності (compatibility testing).

За суб'єктом тестування:

- альфа-тестувальник (alpha tester);
- бета-тестувальник (beta tester).

За часом проведення тестування:

- до передачі користувачеві - альфа-тестування (alpha testing);
- тест приймання (smoke test, sanity test чи confidence test);
- тестування нових функціональностей (new feature testing);
- регресивне тестування (regression testing);
- тест здачі (acceptance or certification test);
- після передачі користувачеві - бета-тестування (beta testing).

За критерієм "позитивності" сценаріїв:

- позитивне тестування (positive testing);
- негативне тестування (negative testing).

За ступенем ізольованості компонентів, що тестуються:

- компонентне тестування (component testing);
- інтеграційне тестування (integration testing);
- системне (або енд-ту-енд) тестування (system or end-to-end testing).
- За ступенем автоматизованості тестування:
- ручне тестування (manual testing);
- автоматизоване тестування (automated testing);
- змішане/напівавтоматизоване тестування (semi automated testing).

За рівнем підготовки до тестування:

- тестування з документації (formal/documentated testing);
- ед хок-тестування (ad hoc testing).
- ЧОРНИЙ ЯЩИК (black box)

Метод, заснований на незнанні внутрішнього об'єкта тестування, називається «чорним ящиком». Він також може називатися поведінковим, оскільки ідеї для проведення тестування ґрунтуються на передбачуваних патернах (patterns) поведінки користувачів. Об'єктом тестування для даного методу найчастіше є системні компоненти перетворення вхідних даних на фактичний результат роботи (код програми, схема бази даних).

*БІЛИЙ ЯЩИК (white box)*

На відміну від попереднього методу, тестування «білої скриньки» ґрунтується на знанні про пристрій та логіку роботи об'єкта, що тестується. Таким чином, сценарії цього методу проєктуються з метою тестування

певного компонента, а не паттерна поведінки. Також цей метод називають «скляним ящиком», «відкритим ящиком» або «ніяким ящиком». Об'єктами тестування даного методу найчастіше виступають загальновідомі технологічні рішення, наприклад, алгоритми кодування, шифрування, архівації. Слід сказати, що тестування бази даних її розробником виконується методом «білого ящика».

#### *СІРИЙ ЯЩИК (gray/grey box)*

Це підхід, що поєднує елементи двох попередніх підходів:

використання патернів поведінки користувачів;

використання знань про пристрій об'єкта, що тестується.

Мабуть, є одним із найчастіших методів, оскільки тестувальники повинні мати специфічний набір знань у кожній предметній галузі, що в більшості випадків неможливо. В результаті цього, процес тестування ґрунтується на наявних знаннях і доповнюється користувачами патернами.

#### *ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ (functional testing)*

Функціональне тестування полягає у перевірці відповідності заявленого функціоналу системи на працездатність у зазначених умовах та із зазначеними параметрами.

#### *ТЕСТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА (UI (“ю-аї”) testing)*

Це тестування, у якому перевіряються елементи інтерфейсу користувача. Перевірка відбувається виходячи з наперед визначених параметрів, у тому числі колірне рішення, розташування елементів керування, коректність їхньої роботи тощо.

#### *ТЕСТУВАННЯ ЛОКАЛІЗАЦІЇ (localization testing)*

Тестування локалізацій передбачає перевірку контенту та текстів програмного засобу на коректність перекладу на інші мови.

#### *ТЕСТУВАННЯ ШВИДКОСТІ І НАДІЙНОСТІ (load/stress/performance testing)*

Це перевірка поведінки системи (або її окремих частин) при одночасному підключенні та виконанні основних функцій системи безліччю користувачів. Подібне тестування є досить тривалим та ресурсомістким, оскільки потребує залучення апаратних засобів, а також інфраструктури для імітації користувачів.

#### *ТЕСТУВАННЯ БЕЗПЕКИ (security testing)*

Безпека тестується залежно від типу системи з використанням спеціальних механізмів та стороннього програмного забезпечення. Основним призначенням такого тестування є забезпечення безпеки даних, неможливості пошкодження внутрішніх модулів системи, а також розмежування прав доступу до системи.

#### *ТЕСТУВАННЯ ДОСВІДУ КОРИСТУВАЧА (usability testing)*

Покликано об'єктивно оцінити досвід користувача (user experience), який працюватиме з інтерфейсом, що розробляється. При юзабіліті-тестуванні також перевіряється інтуїтивність інтерфейсу. Юзабіліті-

тестування часто проводиться шляхом залучення групи потенційних користувачів з метою зібрати враження від роботи з системою.

#### ТЕСТУВАННЯ СУМІСНОСТІ (compatibility testing)

Даний тип тестування призначений для перевірки працездатності системи, що розробляється на певному типі апаратного забезпечення з певним типом програмного забезпечення.

#### Альфа-тестувальник (alpha tester)

Суб'єкт тестування, що працює в компанії розробнику, який професійно чи непрофесійно проводить тестування. Альфа-тестувальником може бути будь-який співробітник компанії, незалежно від його компетенції.

#### БЕТА-ТЕСТУВАЧ (beta tester)

Суб'єкт тестування, який не є співробітником компанії та якому надається можливість користуватися новою системою до того, як вона стане доступною решті. Основним призначенням такої людини є формування думки про систему, а також критичне формування зауважень та побажань до неї. Іноді бета-тестування оплачується.

#### НЕГАТИВНЕ ТЕСТУВАННЯ (negative testing)

Негативне тестування покликане виявити, розробити та виконати сценарії тестування, що призводять до потенційної помилки користувача (error) або потенційного дефекту в системі (failure).

#### ПОЗИТИВНЕ ТЕСТУВАННЯ (positive testing)

Позитивне тестування, на відміну негативного, покликане розробити та перевірити сценарії правильного використання системи користувачами.

#### КОМПОНЕНТНЕ ТЕСТУВАННЯ (component testing)

Компонентне тестування аналізує компоненти системи помилки функціонування, помилки розробки архітектури, інтерфейсів тощо. Більшою мірою виконується на початкових етапах тестування системи та впливає на інші етапи тестування, оскільки непрацездатність будь-якого компонента не дозволяє виконувати тестування вищого рівня, в яких задіяний цей компонент.

#### ІНТЕГРАЦІЙНЕ ТЕСТУВАННЯ (integration testing)

Інтеграційне тестування виконується після компонентного, призначене для перевірки можливості взаємодії кількох компонентів системи, у рамках загальних завдань та процесів.

#### СИСТЕМНЕ (АБО ЕНД-ТУ-ЕНД) ТЕСТУВАННЯ (system or end-to-end testing)

Системне тестування має на увазі повну та досконалу перевірку всієї системи від початку до кінця. Є наймасштабнішим і трудомістким, вимагає великої кількості ресурсів, але може бути автоматизовано за допомогою інструментів автоматичного тестування та безперервної інтеграції (continuous integration).

#### РУЧНЕ ТЕСТУВАННЯ (manual testing)

Представляє послідовне виконання тест-кейсів без допомоги будь-яких програм, що автоматизують роботу. Є найбільш витратним за часом, залежним від компетенції тестувальника та механічно монотонним.

#### АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ (automated testing)

Це окрема дисципліна тестування, значна частина ефективності роботи в рамках якої залежить від того, які завдання віддано для автоматизації та як ця автоматизація була здійснена. Автоматизація може принести величезне полегшення всім тестувальникам, так і завалити роботу всього відділу і відкласти реліз. У цьому методі тестування застосовуються спеціалізовані системи автоматизації та безперервної інтеграції.

#### ТЕСТУВАННЯ З ТЕСТ-КЕЙСІВ (documented testing)

На перших стадіях розробки тестів, що покривають розроблювану систему, відбувається створення тест-кейсів, які є сценаріями роботи системи та очікування від результатів цієї роботи (поява вікна, виведення результату, поява повідомлення і т.д.). Є одним із найпростіших методів тестування, що не висувають особливих вимог до кваліфікації тестувальника.

#### ІНТУЇТИВНЕ ТЕСТУВАННЯ (ad hoc testing)

У цьому методі пошук помилок і некоректностей у процесі роботи системи здійснюється на інтуїтивній основі без застосування конкретних методологій.

### 9.4. Оцінка якості з урахуванням специфікацій та вимог до ПЗ

Стабільні та узгоджені вимоги є фундаментом, який використовується для планування розробки рішення та його здачі в експлуатацію. внесених раніше змін.

Варто також зауважити, що правильна організація роботи з вимогами дозволяє управляти ризиками на ранніх стадіях розробки. Тобто є можливість відстежити ризик, що виникає з певної вимоги, оцінити ступінь його впливу, ймовірність появи. На підставі цього можна розробити план щодо запобігання та усунення наслідків цього ризику. Всі ці дії дозволять уникнути значних фінансових втрат та надати потенційним клієнтам певні гарантії.

Можна зробити висновок, що вимоги відіграють важливу роль у таких областях:

- Планування проєкту;
- Управління ризиками;
- приймальні тестування;
- Формування компромісів (узгоджень);
- Управління змінами.

Очевидним стає факт використання вимог у всьому життєвому циклі проєкту.

Для демонстрації взаємозв'язку стадій проєкту та процесу управління вимогами існує класична V-модель (V-model), наведена на рисунку 9.4. V-

модель візуалізує зв'язок між вимогами і методами їх тестування, і навіть відображає системну розробку в термінах рівнів (layers), де кожен рівень співвідноситься з . Варто зауважити, що основний принцип роботи з вимогами на всіх рівнях залишається постійним.

Ще однією областю, де використання вимог може принести значну користь, є зв'язок між проєктами. Більшість розробників бажають:

- максимально збільшити використання напрацьовань у різних проєктах;

- керувати сімействами подібних продуктів;

- використовувати програмне управління для узгодження дій;

- оптимізувати розробку, використовуючи досвід попередніх проєктів.

Коректно складені вимоги заінтересованих сторін (stakeholders requirements) можуть бути хорошим і лаконічним описом бажаного результату для вищого керівництва, оскільки, як було зазначено, сформулюється без використання специфічних термінів і понять.

Так само, системні вимоги (system requirements) можуть послужити чудовим технічним описом проєкту.

Обидва цих набору вимог є основою для всіх наступних дій у рамках проєкту, а досвід, який отримується при його реалізації, надалі благотворно впливає на процес складання цих типів вимог. Тривіальним фактом є можливість у майбутніх проєктах заздалегідь передбачати можливості запобігання проблемам, виявлених у межах схожих із нею проєктів. Схема отримання та поширення досвіду в організації показано на рис 9.5



Рис 9.4. Зв'язок вимог із тестуванням

Дана схема демонструє низхідний процес отримання досвіду та висхідний процес поширення досвіду. Це з тим, що досвід, представлений застосуванням специфічних технологій, промислових фреймворків,

технологічних рішень тощо. може бути отриманий виключно в процесі реалізації конкретного проєкту, тобто поетапної декомпозиції завдань та підбору відповідних способів їх вирішення. Процес поширення досвіду, з іншого боку, можливий лише за наявності будь-якої бази реалізованих проєктів, з якої можна запозичувати існуючі рішення конкретних завдань та застосовувати їх до поточних проєктів.

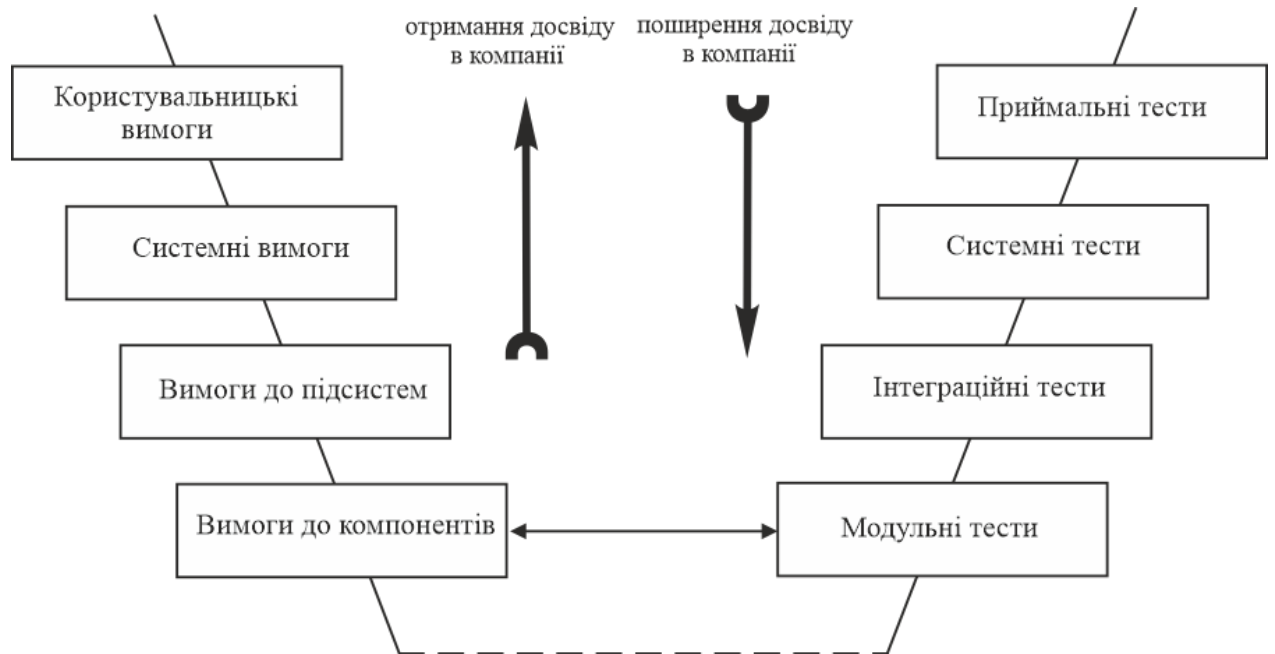


Рис. 9.5. Схема здобуття та поширення досвіду в організації

Як вже було зауважено, вимоги дуже часто змінюються і не реагувати на їх зміни не можна, оскільки це може спричинити часткове або повне незадоволення потреб користувачів, внаслідок чого продукт, що розробляється, виявиться незатребуваним на ринку, а, отже, нерентабельним. На підставі цього можна сміливо стверджувати, що розробка вимог та керування їх змінами тісно пов'язані між собою процеси.

При виникненні необхідності внесення будь-яких змін до проєкту необхідно врахувати їх вплив на такі аспекти:

- якість продукту;
- вартість продукту, що випускається;
- графік робіт із випуску.

Після проведеного аналізу процес внесення змін зводиться до кількох кроків:

- прийняти чи відхилити зміну;
- узгодити вартість зміни із замовниками/постачальниками;
- організувати роботи з переробки.

Для того щоб провести аналіз впливу змін на продукт, що розробляється, існує базова концепція встановлення і контролю зв'язків між вимогами (requirement traceability).

Основною концепцією, що дозволяє проводити аналіз такого роду впливів, є можливість встановлення та подальшого контролю зв'язків між вимогами. Таким чином, можна стверджувати, що керування змінами (change management) є важливою частиною процесу роботи з вимогами. Можливий вплив змін до вимог показано на рисунку 9.6.

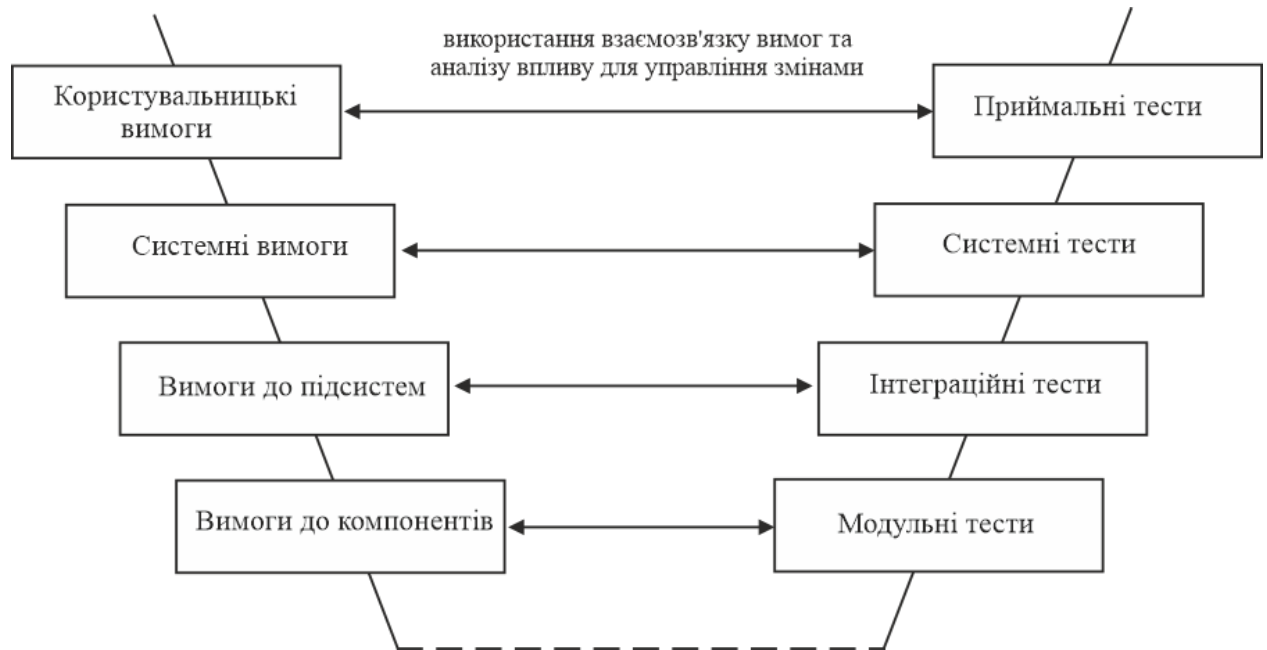


Рис. 9.6. Ризики управління змінами пов'язані із взаємопов'язаністю вимог

Очевидним стає факт залежності можливості управління проектом та процесу роботи з вимогами. Не маючи вимог, дуже важко коректно оцінити прогрес робіт, а в момент внесення змін до початкового завдання не буде можливості відстежити їх вплив на перебіг проекту.

Також варто зазначити, що коректно сформульовані на кожному рівні вимоги, надають керівнику проекту правильне уявлення про проект, про хід його виконання та прогрес реалізації, а також дають можливість ефективніше виконувати свою роль в управлінні загальним процесом розробки.

Для того, щоб чітко розуміти взаємозв'язок вимог на всіх рівнях, необхідно явним чином створювати та аналізувати зв'язки між ними. Більшою мірою це необхідно для розуміння трансляції вимог верхнього рівня (користувацьких) до вимог підсистем та їх компонентів.

У контексті розробки вимог, створення та аналіз зв'язків необхідні, в першу чергу, для розуміння того, як вимоги високого рівня – загальні цілі, завдання, побажання, очікування, потреби тощо. – трансформуються у вимоги низького рівня. Отже, в основному зв'язки потрібні між різними рівнями інформації.

Використання зв'язків може принести такі вигоди:

- велика впевненість у досягненні цілей (встановлення зв'язків та формалізація їх контролю призводить до чіткого розуміння того, як саме досягаються цілі);

- можливість оцінити вплив змін (існування зв'язків між вимогами дає можливість проводити різного роду аналіз впливу змін, що вносяться);

- можливість оцінити внесок працівників, підрядників та субпідрядників (з'являється можливість чітко оцінити ту частину роботи, яку виконують за проектом співробітники та інші організації);

- можливість контролювати хід проекту та оцінювати обсяг виконаної роботи;

- можливість зіставляти витрати та можливу вигоду (однозначне визначення зв'язку між вимогами та певними компонентами системи, дозволяє порівнювати витрати з передбачуваним позитивним ефектом від їх реалізації).

Можна виділити два основні типи зв'язків:

- зв'язки між вимогами різних рівнів (зазвичай мають тип «багато до багатьох» і одна вимога нижнього рівня може бути пов'язана з кількома вимогами вищого рівня, як, втім, і навпаки);

- зв'язки між вимогами рівня.

Стрілки на лініях зв'язків завжди проставляються від приймача інформації до джерела. Для такої угоди є дві причини:

- такий формат стрілки найчастіше відповідає хронологічному порядку появи інформації (зв'язок завжди вказує на інформацію, що з'явилася раніше);

- дуже часто це відповідає також і правам на володіння інформацією (одній людині належать вихідні з документа зв'язку, іншій - вхідні).

Приклад зв'язків між вимогами представлено рисунку 9.7.

Можна виділити три основні методи аналізу зв'язків між вимогами: аналіз впливу, аналіз наслідків та аналіз покриття [28]. Коротка характеристика цих методів наведена у таблиці 9.3

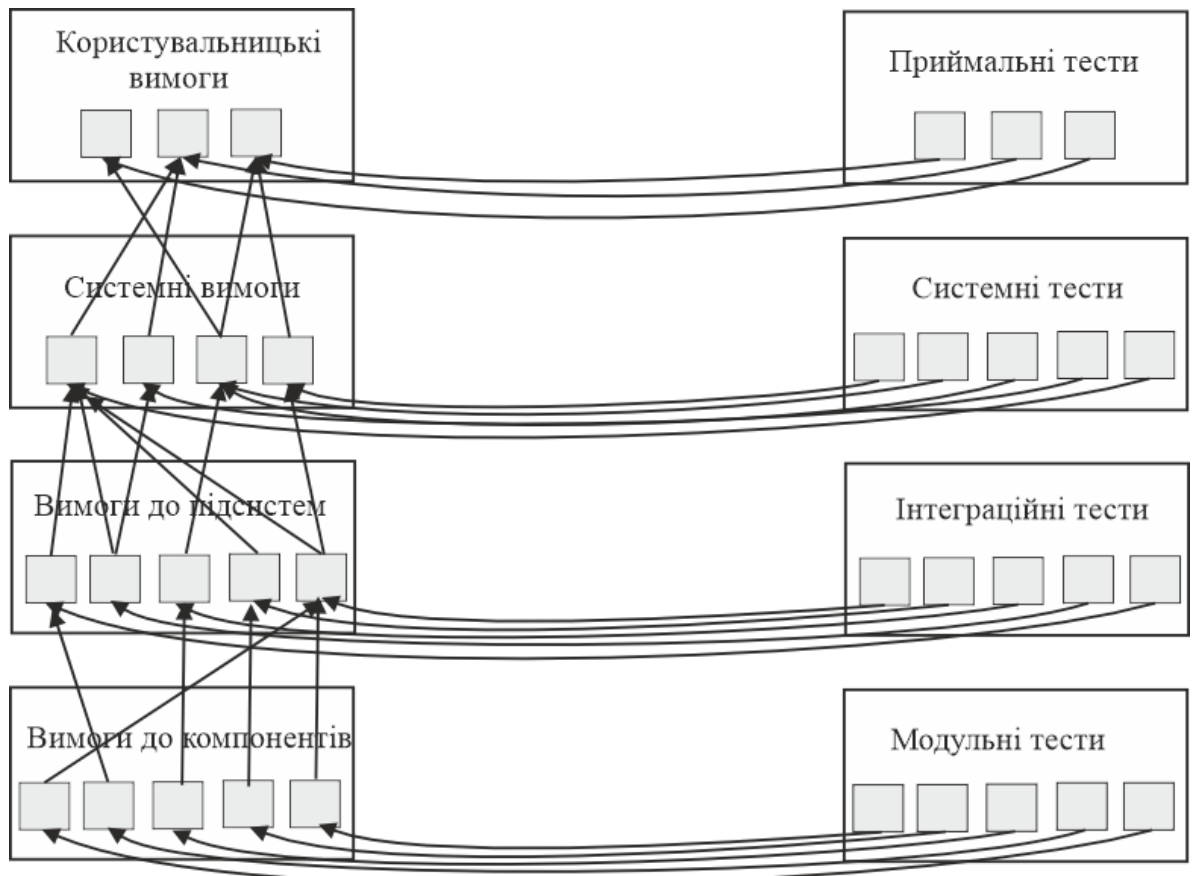


Рис. 9.7. Зв'язки між вимогами

Таблиця 9.3 – Методи аналізу зв'язків між вимогами

Метод аналізу	Опис	Підтримуваний процес
Аналіз впливу	Аналіз вхідних зв'язків з метою оцінки наслідків зміни вимог	Процес зміни
Аналіз наслідків	Аналіз вихідних зв'язків з метою оцінки доцільності зміни	Аналіз економічної доцільності
Аналіз покриття	Аналіз вхідних зв'язків з метою оцінки прогресу робіт	Проектування та звітність керівництву

Аналіз впливу дозволяє оцінити кількість елементів нижчого рівня, на які вплине зміна, що вноситься в конкретний елемент, і наскільки сильно це на них позначиться. Інакше кажучи, аналіз впливу допомагає з відповіддю: «Що буде, якщо змінити цю вимогу?».

Аналіз наслідків є повною протилежністю аналізу зв'язків. У розгляд беруться елементи нижчого рівня, та був перевіряється їх відповідність елементам вищого рівня. Якщо елементи нижніх рівнів немає зв'язків із

елементами верхніх рівнів, то, мабуть, вони лише збільшують витрати виробництва продукту і може бути виключені. Тобто аналіз наслідків відповідає питанням: «Це справді потрібно?».

Схема процесів аналізу впливу та наслідків представлена на рисунку 9.8.

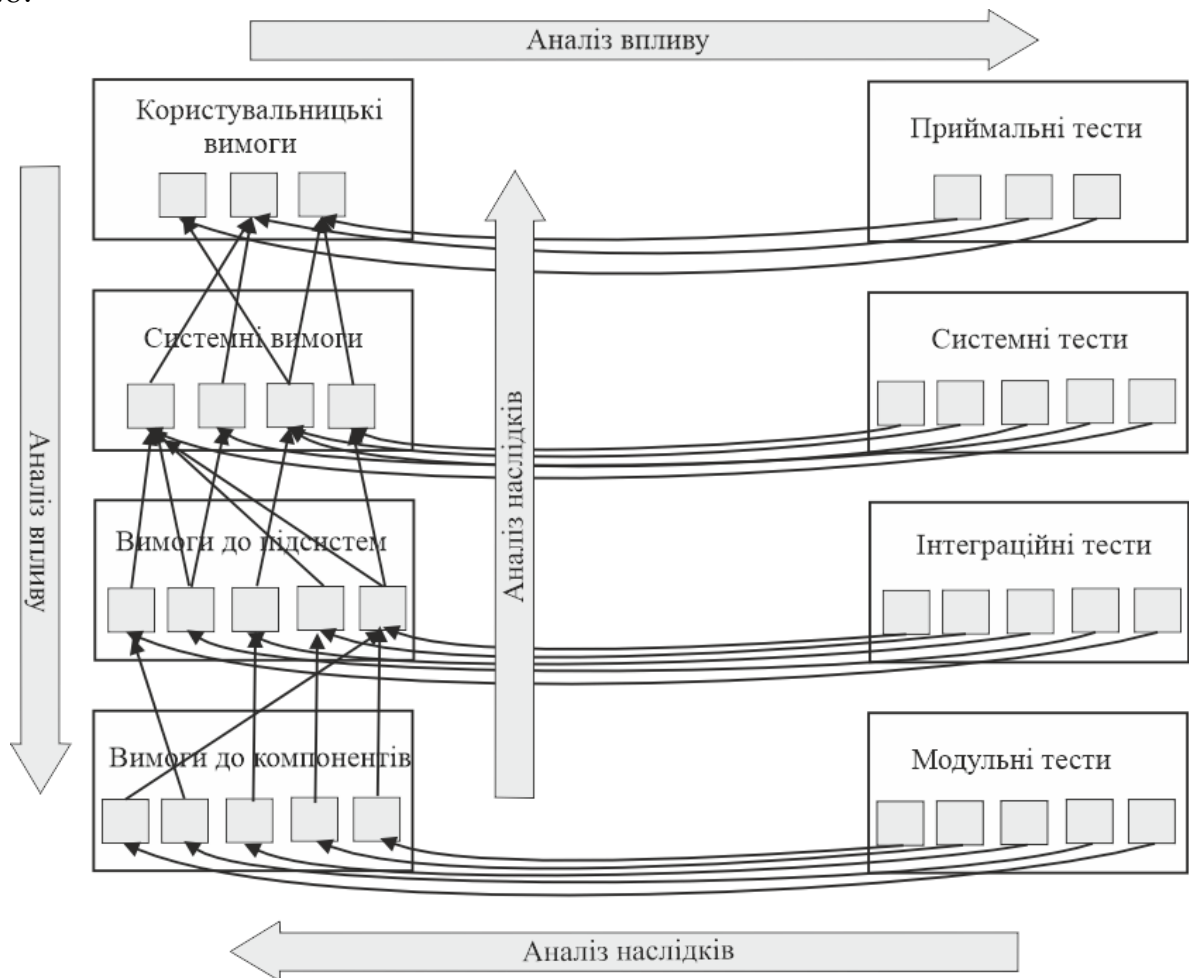


Рис. 9.8. Аналіз впливу та аналіз наслідків

Аналіз покриття, як і аналіз впливу, розглядає вхідні зв'язки. З його допомогою можна перевірити наявність зв'язків між усіма елементами вищих рівнів з елементами нижчих рівнів, і навіть з необхідними цих елементів тестами. Відсутність зв'язків між рівнями тестами свідчить про неможливість задовольнити чи протестувати конкретну вимогу. У той же час наявність подібних зв'язків не гарантує можливості задовольнити та протестувати вимогу. Аналіз покриття представлений рисунком 9.9.

Також аналіз покриття можна використовувати для вимірювання прогресу робіт (скільки споконвічно заявлених вимог задоволені та протестовані). Мається на увазі, що системні інженери розробляють системні вимоги, що відповідають вимогам користувачів, або, як ще кажуть, «покривають» їх.

У процесі розробки, кожна системна вимога пов'язана з тими вимогами користувача, для задоволення яких воно призначене. Завершеність системних

вимог на будь-якому етапі проекту може бути визначена як відсоток покриття вимог користувача системними.

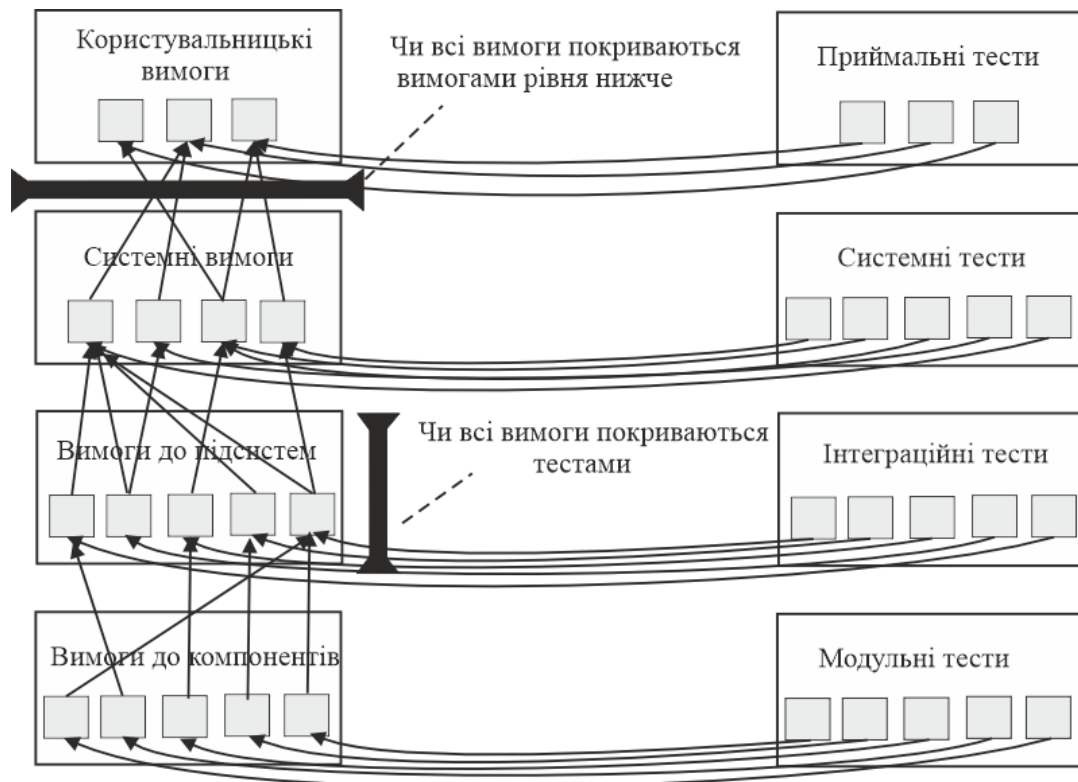


Рис. 9.9. Схема аналізу покриття

Аналогічний принцип можна застосовувати для вимірювання прогресу розробки планів тестування. Завершеність готових планів тестування окреслюється відсоток покриття вимог тестами.

Слід виділити наявність взаємозв'язку між розробкою вимог і системним моделюванням. У контексті управління вимогами системне моделювання використовується для опису та наочного відображення різних аспектів системи, причому зазвичай використовується кілька різних моделей, можливо навіть взаємопов'язаних. При цьому деякі аспекти системи можна описати вимогами (в текстуальному вигляді).

Перевагою такого підходу є те, що невеликий обсяг пов'язаної інформації, що відноситься до певного аспекту системи, може бути зібраний в одній моделі, оброблений, структурований та проаналізований методами, які найбільше підходять для роботи з такою інформацією.

Моделювання допомагає системному інженеру в аналізі вимог для:

- обговорення розроблюваної системи із замовником та покращення взаєморозуміння з колегами;
- аналізу системи з метою переконатися у наявності бажаних системних властивостей (emergent properties) та у відсутності небажаних властивостей;

– розуміння процесу перевірки реалізація вимог, за її трансформації на нижчий рівень

На різних рівнях використовують різні види моделей. Для формування користувачьких вимог (stakeholder requirements) можна використовувати сценарії користувача (stakeholder scenarios).

При переході до системних вимог можна використовувати різні типи функціональних моделей. Наприклад, UML-діаграми: діаграми класів (class diagrams), діаграми послідовності повідомлень (message sequence charts) та діаграми станів (state charts).

При переході від системних вимог до архітектури, більшого значення починають грати різні аспекти продуктивності створюваної системи. У цьому випадку кілька різних типів моделей можуть бути використані для того, щоб продемонструвати те, що обрана архітектура може задовольнити як функціональним, так і нефункціональним вимогам.

Як було зазначено, вимоги тісно пов'язані з тестуванням. Під тестуванням можна розуміти будь-яку дію, спрямовану на виявлення та запобігання дефектам у системі, де дефект – це відхилення від вимог. Таким чином, на додаток до класичних методів тестування додаються рецензування, інспектування та аналіз з використанням моделювання.

Тестування починається на етапі розробки специфікацій системи (design) і полягає у рецензуванні вимог, інспекції специфікацій та застосуванні різних форм моделювання системи.

На рисунку 9.10 продемонстровано V-модель циклу розробки вимог та стратегію перевірки системи.

Як зазначалося, системне проектування є життєво важливим процесом для бізнесу, оскільки, з його допомогою, прийняті рішення визначають можливості виходу ринку з «правильним» продуктом. Для розмежування зон відповідальності та прискорення прийняття рішень на початковому етапі розробки необхідно чіткий поділ між «областю проблем» і «областю вирішення» конкретного проекту.

До проблемної галузі слід зарахувати:

- Формулювання проблем;
- моделі використання;
- Користувачькі вимоги.

Починаючи з системних вимог, все має бути віднесено до галузі рішення. У таблиці 9.4 представлено «ідеальний» поділ між проблемною областю та областю вирішення, а також сформульовані цілі, яким мають задовольняти вимоги трьох верхніх рівнів.

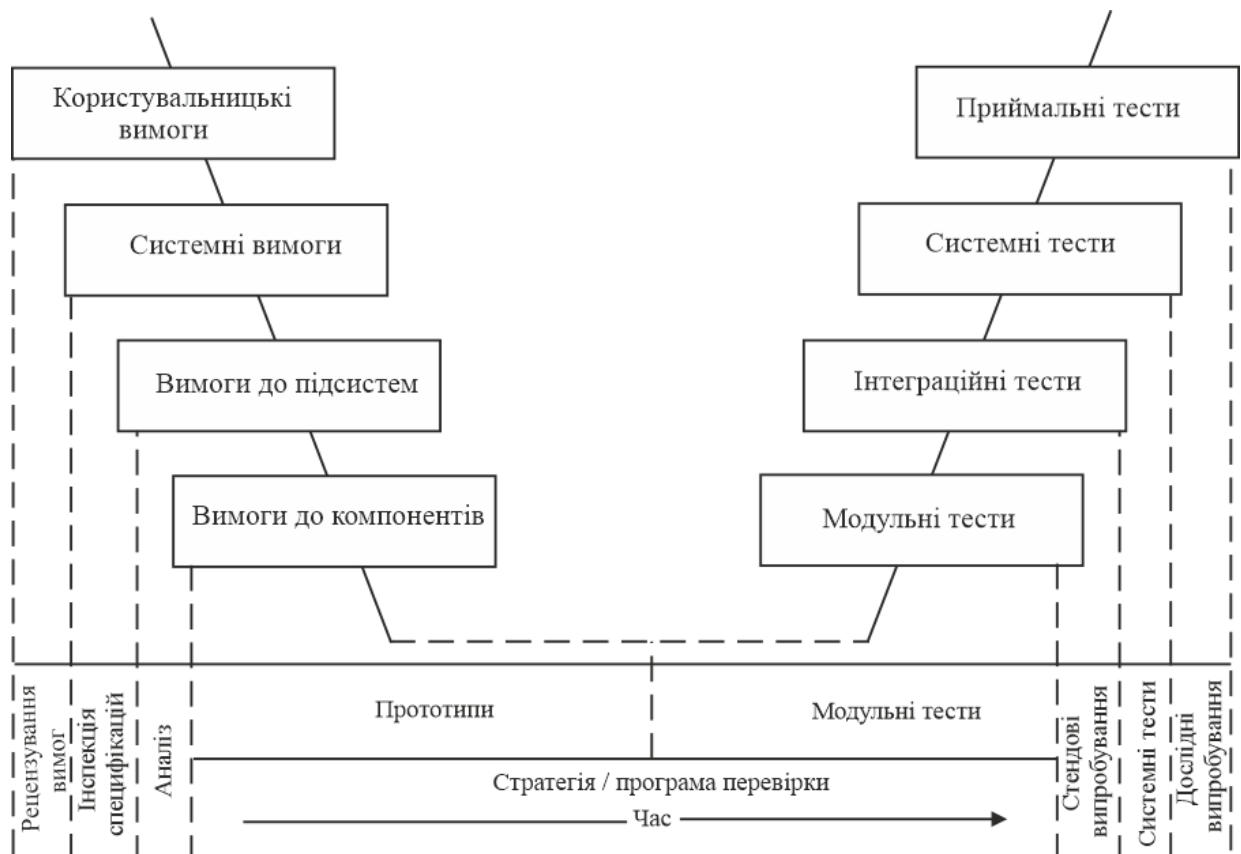


Рис. 9.10. Стратегія перевірки системи відповідно до циклу розробки

Опис області проблем має містити можливості системи і все, що необхідно для визначення проблем, при цьому не повинна містити нічого, що визначає конкретні рішення. Завдяки цьому, системні інженери не обмежуються у виборі найкращого вирішення проблеми.

Моделювання, допомагаючи переходити від рівня до рівня, також може привносити елементи конкретних рішень, навіть на верхньому рівні. Щоб уникнути ухилу в площину рішення, на ранніх етапах краще застосовувати моделювання тільки для опису системи (підсистеми).

Відсутність чіткого поділу між проблемами та рішеннями може призвести до таких негативних наслідків:

- недостатнє розуміння існуючих проблем;
- неможливість визначити межі (масштаб) системи та зрозуміти який функціонал повинен до неї входити, а який ні;
- домінування розробників та виконавців у дискусіях про систему, оскільки єдиний опис, що існує для системи, описує її у термінах реалізації, а не у формулюваннях проблем;
- неможливість знаходження найкращого рішення через обмеження свободи у виборі рішення.

Таблиця 9.4 – Короткий опис областей рішень та проблем

Рівень вимог	Область	Погляд	Ціль
Користувальницькі вимоги	Область проблем	Представники зацікавлених сторін	Визначає - що користувач хоче досягти за допомогою системи, що створюється.
Системні вимоги	Область рішень	Аналітик	Абстрактно визначає – як система задовольнятиме користувальницьким вимогам.
Системні специфікації (архітектура)	Область рішень	Архітектор	Визначає – як конкретна архітектура системи задовольнятиме системним вимогам.

У зв'язку з великою складністю та широтою галузі вимог у проєктах, необхідний процес управління ними. Зіставивши інформацію, можна виділити три стадії цього процесу (рис 9.11):

- Планування;
- Розробка;
- Управління змінами.

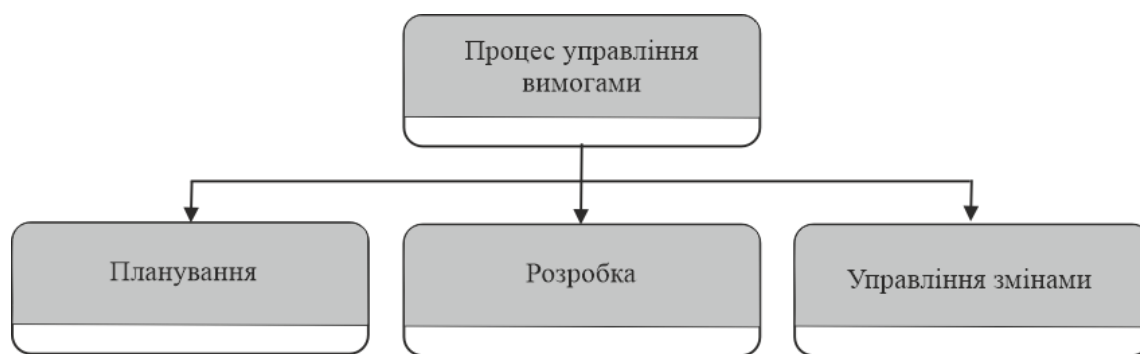


Рис. 9.11. Стадії процесу керування вимогами

Процес управління вимоги починається з планування. На етапі планування системний аналітик створює план керування вимогами та шаблони необхідної документації. Планування – перший крок під час роботи з вимогами, він починається етапі перед проєктного обстеження.

План управління вимогами одна із найважливіших документів у процесі управління вимогами. У цьому документі визначаються типи вимог та атрибути кожного типу, відносини між вимогами, документи, що використовуються у цьому процесі. Також системний аналітик визначає та заносить у план рішення про використання спеціального інструментального

засобу для управління вимогами. Розширений варіант плану управління вимогами може містити опис ролей, що у процесі; завдань, що виконуються кожною роллю, та іншу службову інформацію.

Сучасні інструментальні засоби дозволяють створювати автоматичні звіти із необхідною інформацією. Якщо ухвалено рішення про те, що документація буде створюватися автоматично з використанням звітів, на етапі планування необхідно створити шаблони таких звітів. Шаблони звітів, як і шаблони документів, мають бути розроблені з урахуванням вимог державних стандартів.

Як уже говорилося раніше, планування є початковим етапом у роботі з вимогами і є вхідною точкою в процес розробки та управління вимогами. Основні результати планування вимог наведені нижче:

- методологія та процес розробки програмного забезпечення;
- ролі учасників проєкту та завдання, що виконуються ними;
- список учасників проєкту та їх навички;
- різні регламенти робіт у частині виявлення, документування, узгодження вимог та ін;
- план керування вимогами;
- шаблони документів.

Усі результати планування необхідно узгоджувати з керівником проєкту, оскільки він має бути в курсі всіх аналітичних завдань та їхнього розподілу між членами команди.

Методологію, процеси, типи вимог, шаблони документів, регламенти перевірки вимог та управління змінами необхідно узгоджувати з процесами розробки та тестування.

Для того, щоб чітко розуміти процес розробки вимог, який повною мірою відповідатиме плану, необхідно розглянути кожен його етап. Залежно від масштабу компанії та складності завдань, які стоять під час реалізації проєктів, кількість етапів може різнитися. Процес розробки вимог містить такі етапи:

- Ідентифікація зацікавлених сторін.
- Виявлення вимог заінтересованих сторін.
- Формування вимог.
- Уточнення та переформулювання вимог.
- Аналіз вимог.
- Приведення вимог до вигляду однаково зрозумілому всім зацікавлених сторін.
- Визначення критеріїв приймання вимог.
- Визначення стратегії перевірки вимог.
- Створення тестів.
- Специфікація вимог.
- Визначення пріоритетів вимог.
- Виведення залежних вимог.
- Класифікація вимог.

- Розподіл вимог.
- Відстеження вимог.
- Тестування вимог.
- Перевірка вимог.
- Твердження вимог.

Оцінка технічного рівня (якості) програмного забезпечення.

Для оцінки ефективності програмного забезпечення пропонується використовувати набір приватних показників, наведених у таблиці 9.5:

Для спільного використання вищепоказаних показників запроваджується співвідношення з використанням вагових коефіцієнтів:

$$KK = D_1y_1 + D_2y_2 + D_3y_3.$$

Для вагових коефіцієнтів  $D$  існує прийняті значення (таблиця 9.6):

Проблеми при кількісному визначенні значень показників було скорочено з використанням бальних оцінок.

$y_{1.1} =$

- 6 – при індивідуальній підготовці
- 8 - при комбінованій підготовці
- 10 – при централізованій підготовці

$y_{1.2} =$

- 5 – при регламентації використання документа
- 10 – при регламентації використання показника

$y_{1.3} =$

- 5 - при відмові елемента АІС з порушенням
- 8 - без порушення при резервуванні функцій-персоналом
- 10 – за рахунок резервування засобами автоматизації

$y_{1.4} =$

- 3 – до 5%
- 8 – від 5 до 20%
- 10 - понад 20%

$y_{2.1} =$

- 3 – до 8 год
- 5 - від 8 до 14 год
- 8 - від 14 до 18 год
- 10 - понад 18 год

$y_{2.2} =$

- 2 - при неавтоматичному зв'язку з обчислювальним центром (ВЦ)
- 6 - при напівавтоматичному зв'язку з ВЦ
- 10 - при автоматичному зв'язку з ВЦ

Таблиця 9.5 – Набір приватних показників

Показник якості	Опис показника	Формула для розрахунку	Опис змінних
Рівень організаційного забезпечення	підготовка первинних даних; їх використання; стійкість до порушень; відношення кількості оптимізаційних завдань до сумарного числа	$y_1 = y_n \sum_{i=1}^4 d_{1i} y_{1i}$	$y_n$ залежить від методології проектування, що використовується: з використанням автоматизованого проектування $n = 1,0$ ; на основі типових проектних рішень $n = 0,8$ ; з використанням прототипів $u_n = 0,7$ ; з використанням оригінального проектування $n = 0,6$ ;
Рівень технічного забезпечення	середнє значення завантаження комп'ютера на добу; зв'язок із периферією; використовувані засоби відображення	$y_2 = y_n \sum_{i=1}^3 d_{2i} y_{2i}$	$d$ - Вагові параметри показників рівня; $y$ - Оцінка даного показника рівня
Рівень математичного забезпечення	тип комп'ютера; інформаційне забезпечення; використовувана система програмування.	$y_3 = y_n \sum_{i=1}^3 d_{3i} y_{3i}$	

Таблиця 9.6 – Прийняті значення для вагових коефіцієнтів  $D$ 

Число працівників підприємства	Тип виробництва	Вид забезпечення		
		Організаційне	Технічне	Математичне
		$D_1$	$D_2$	$D_3$
До 2000	Індивідуальне	0,6	0,3	0,1
	Серійне	0,5	0,3	0,2
	Масове	0,4	0,3	0,3
2000 - 8000	Індивідуальне	0,7	0,2	0,1
	Серійне	0,5	0,2	0,3
	Масове	0,3	0,2	0,5
Понад 8000	Індивідуальне	0,7	0,2	0,1
	Серійне	0,4	0,2	0,4
	Масове	0,1	0,2	0,7

$Y_{2.3} =$

- 3 - передбачена сигналізація при відключеннях
- 7 - при регламентованих засобах відображено
- 10 - при відповідно - запитних пристроях

$Y_{3.1} =$

- 4 - тип ЕОМ попереднього покоління
- 8-тип ЕОМ нового покоління

$Y_{3.2} =$

- 3 - при локальному вирішенні завдань
- 5 - за єдиного нормативного господарства
- 10 – з єдиною інформаційною базою

$Y_{3.3} =$

- 5 - алгоритмічні мови
- 10 – операційні системи

Таким чином, з урахуванням описаних вище метрик можна оцінювати в автоматизованому режимі ефективність та якість програмного забезпечення на різних етапах створення.

Розглянуті у цьому розділі метрики мають значення як процесу оцінки якості ПЗ, так інших напрямів інформатики. Описані гіпотези Холстеда були підтверджені великих обсягах даних і дали потужний поштовх у розвиток ідей лінгвістичного підходу у технічних науках.

Програмометрика має значне практичне значення, оскільки кількісні метрики даного наукового напрямку дають можливість на етапі проектування оригінального програмного забезпечення розробляти найкращу структуру, дотримуватись термінів розробки програмного забезпечення, оцінювати на ранніх етапах трудомісткість, надійність, ефективність програмного забезпечення, проводити процедуру техніко-економічного обґрунтування.

## **9.5. Важливість оцінки якості для забезпечення надійності програмного забезпечення**

Якість при використанні в процесі експлуатації відображає ступінь задоволеності потреб користувачів продуктом у досягненні необхідних цілей із результативністю, продуктивністю та задоволеністю в заданих умовах.

Під результативністю розуміється точність і повнота досягнення заданих цілей користувачами при застосуванні програмного засобу. складова частина інтегральної оцінки функціонування та використання програмних засобів.

Якість використання в процесі експлуатації – це об'єднаний ефект функціональних та конструктивних характеристик якості програмного засобу для користувача.

Зв'язок із зовнішнім та внутрішнім аспектами якості програмного засобу залежить від типу та завдань користувачів:

- для кінцевого користувача якість у процесі експлуатації програмного засобу визначається характеристиками функціональних можливостей, надійності, практичності та ефективності;
- для персоналу супроводу якість у процесі експлуатації програмного засобу характеризується безпосередньою супровідністю;
- для персоналу по впровадженню якість у процесі експлуатації програмного засобу визначається насамперед мобільністю.

Результати оцінювання цього аспекту якості можуть бути різними для сценаріїв і завдань окремих користувачів. Для великих програмних засобів практично неможливо виміряти всі внутрішні і зовнішні характеристики якості з усіма атрибутами, аналогічно не оцінюються і всі можливі сценарії. Відбувається ранжування сценаріїв за пріоритетами, та був проводиться оцінювання показників, з урахуванням оптимального розподілу наявних ресурсів.

Якість при використанні в процесі експлуатації характеризується ефектом та складністю використання програмного засобу, що описуються трудомісткістю з необхідною результативністю.

Оцінка якості програмних систем є необхідним етапом для забезпечення ефективної та безпечної роботи програмного забезпечення. Застосування відповідних метрик якості, методів тестування, верифікації та валідації дозволяє не тільки виявити дефекти, але й запобігти серйозним проблемам у процесі експлуатації. Збереження високих стандартів якості є ключовим для досягнення стабільності, надійності та конкурентоспроможності програмних продуктів.

## **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Назвіть основні характеристики якості ПЗ (згідно ISO/IEC 25010) і для кожної запропонуйте по одній вимірній метриці.
2. Поясніть відмінності між тестуванням, верифікацією і валідацією. Наведіть приклади артефактів, що перевіряються на кожній стадії.
3. Що таке «покриття тестами» (coverage) - види покриття (statement, branch, path, MC/DC)? Як обрати цільовий рівень покриття для критичної системи?
4. Як обчислити щільність дефектів і які висновки можна зробити порівнявши щільності для двох релізів? Наведіть формулу і приклад розрахунку.
5. Опишіть процедуру оцінки якості ПЗ з урахуванням специфікацій та вимог: як сформулювати критерії прийняття, як пов'язати тест-кейси з вимогами.

## Тема 10. Експертний аналіз проблем. Обробка і використання

знань.

### 10.1 Метод експертних оцінок

У сучасному системному аналізі однією з ключових задач є ефективне прийняття рішень в умовах невизначеності та складності. Зазвичай для цього використовуються різні методи аналізу даних, моделювання, оптимізації. Проте в багатьох випадках необхідно залучити фахівців - експертів, які мають глибокі знання в певній галузі. Саме експертний аналіз дозволяє отримати знання для прийняття обґрунтованих рішень в умовах нестачі інформації або в ситуаціях, коли наявні дані є неповними або нечіткими [19].

Експертний аналіз полягає у використанні знань фахівців для оцінки складних ситуацій і пошуку оптимальних рішень. В цьому процесі важливо правильно обробляти ці знання та використовувати їх у рамках системного підходу для вирішення конкретних проблем.

Зростаюча складність управління організаціями вимагає ретельного аналізу цілей та завдань діяльності, шляхів та засобів їх досягнення, оцінки впливу різних факторів на підвищення ефективності та якості роботи. Це призводить до необхідності широкого застосування експертних оцінок у процесі формування та вибору рішень.

Експертиза як спосіб отримання інформації завжди використовувалася під час вироблення рішень. Проте наукові дослідження з її раціонального проведення було розпочато лише три десятиліття тому. Результати цих досліджень дозволяють зробити висновок про те, що в даний час експертні оцінки є в основному науковим методом, що сформувався, аналізу складних неформалізованих проблем.

Сутність методу експертних оцінок полягає в раціональній організації проведення експертами аналізу проблеми з кількісною оцінкою суджень та опрацюванням їх результатів. Узагальнена думка групи експертів сприймається як вирішення проблеми.

У процесі прийняття рішень експерти виконують інформаційну та аналітичну роботу з формування та оцінки рішень. Все різноманіття розв'язуваних ними завдань зводиться до трьох типів:

- формування об'єктів,
- оцінка характеристик,
- формування та оцінка характеристик об'єктів.

Формування об'єктів включає визначення можливих подій та явищ, побудову гіпотез, формулювання цілей, обмежень, варіантів рішень, визначення ознак та показників для опису властивостей об'єктів та їх взаємозв'язків тощо. У задачі оцінки характеристик експерти виробляють виміри достовірності подій та гіпотез, важливості цілей, значень ознак та показників, переваг рішень. У задачі формування та оцінки характеристик об'єктів здійснюється комплексне рішення перших двох типів задач. Таким

чином, експерт виконує роль генератора об'єктів (ідей, подій, рішень тощо) та вимірювача їх характеристик.

При розв'язанні розглянутих завдань усі безліч проблем можна поділити на два класи: з достатнім та недостатнім інформаційним потенціалом. Для проблем першого класу є необхідний обсяг знань та досвіду щодо їх вирішення. Тому стосовно цих проблем експерти є якісними джерелами та досить точними вимірювачами інформації. Для таких проблем узагальнена думка групи експертів визначається посередництвом їх індивідуальних суджень і є близькою до істинної.

Щодо проблем другого класу, експерти вже не можуть розглядатися як досить точні вимірювачі. Думка одного експерта може виявитися правильною, хоча вона сильно відрізняється від думки решти експертів. Обробка результатів експертизи під час вирішення проблем другого класу неспроможна ґрунтуватися на методах опосередкування.

Метод експертних оцінок застосовується для вирішення проблем прогнозування, планування та розробки програм діяльності, нормування праці, вибору перспективної техніки, оцінки якості продукції та ін.

Для застосування методу експертних оцінок у процесі прийняття рішень необхідно розглянути питання підбору експертів, проведення опитування та опрацювання його результатів. Ці питання викладаються у наступних параграфах.

## **10.2 Підбір експертів**

Залежно від масштабу розв'язуваної проблеми організацію експертизи здійснює ОПР - або призначається групою управління. Підбір кількісного та якісного складу експертів проводиться на основі аналізу широти проблеми, необхідної достовірності оцінок, характеристик експертів та витрат ресурсів.

Широта вирішуваної проблеми визначає необхідність залучення до експертизи спеціалістів різного профілю. Отже, мінімальна кількість експертів визначається кількістю різних аспектів, напрямів, які необхідно врахувати під час вирішення проблеми.

Достовірність оцінок групи експертів залежить від рівня знань окремих експертів та кількості членів. Якщо припустити, що експерти є досить точними вимірювачами, то зі збільшенням числа експертів достовірність експертизи групи зростає.

Витрати ресурсів для проведення експертизи пропорційні кількості експертів. Зі збільшенням числа експертів збільшуються тимчасові та фінансові витрати, пов'язані з формуванням групи, проведенням опитування та опрацювання його результатів. Отже, підвищення достовірності експертизи пов'язані з збільшенням витрат. Наявні фінансові ресурси обмежують максимальну кількість експертів у групі. Оцінка числа експертів

знизу та зверху дозволяє визначити межі загальної кількості експертів у групі.

Характеристики групи експертів визначаються з урахуванням індивідуальних характеристик експертів: компетентності, креативності, ставлення до експертизи, конформізму, конструктивності мислення, колективізму, самокритичності.

Нині перелічені показники переважно оцінюються якісно. Для низки показників є спроби запровадити кількісні оцінки.

**Компетентність** –ступінь кваліфікації експерта у певній галузі знань. Компетентність може бути визначена на основі аналізу плідної діяльності спеціаліста, рівня та широти знайомства з досягненнями світової науки та техніки, розуміння проблем та перспектив розвитку.

Для кількісної оцінки компетентності використовується коефіцієнт компетентності, з урахуванням якого зважується думка експерта. Коефіцієнт компетентності визначається за апріорними та апостеріорними даними. При використанні апріорних даних оцінка коефіцієнта компетентності провадиться до проведення експертизи на основі самооцінки експерта та взаємної оцінки з боку інших експертів. При використанні апостеріорних даних оцінка коефіцієнта компетентності здійснюється на основі обробки результатів експертизи.

Існує ряд методик визначення коефіцієнта компетентності за апріорними даними. Найбільш простою є методика оцінки відносних коефіцієнтів компетентності за результатами висловлювання фахівців про склад експертної групи. Сутність цієї методики ось у чому. Низці фахівців пропонується висловити судження про включення осіб до експертної групи для вирішення певної проблеми. Якщо до цього списку потрапляють особи, які не увійшли до початкового списку, їм також пропонується назвати фахівців для участі в експертизі. Провівши кілька турів такого опитування, можна скласти досить повний перелік кандидатів у експерти. За результатами проведеного опитування складається матриця, у комірках якої проставляються змінні  $x_{i,j}$ , рівні

$x_{i,j} = 1$ , якщо  $j$ -й експерт назвав  $i$ -го експерта

$x_{i,j} = 0$ , якщо  $j$ -й експерт не назвав  $i$ -го експерта.

Причому кожен експерт може включати або не включати себе до експертної групи.

За даними матриці обчислюються коефіцієнти компетентності як відносні ваги експертів за формулою:

$$k_i = \frac{\sum_{j=1}^m x_{ij}}{\sum_{i=1}^m \sum_{j=1}^m x_{ij}}, \quad i = \overline{1, m},$$

де  $k_i$  - Коефіцієнт компетентності  $i$ -го експерта,  $m$  - кількість експертів (розмірність матриці  $\| x_{i,j} \|$ ). Коефіцієнти компетентності нормовані так, що їх сума дорівнює одиниці:

$$\sum_{j=1}^m k_i = 1.$$

Змістовний зміст коефіцієнтів компетентності, обчислених за даними таблиці  $\|x_{ij}\|$ , полягає в тому, що сума підрахунку одиниць (число “голосів”), поданих за  $i$ -го експерта, і поділяється на загальну суму всіх одиниць. Таким чином, коефіцієнт компетентності визначається як відносна кількість експертів, які висловилися за включення  $i$ -го експерта до списку експертної групи.

**Креативність.** Це здатність вирішувати творчі завдання. В даний час крім якісних суджень, заснованих на вивченні діяльності експертів, немає жодних пропозицій щодо оцінки цієї характеристики.

**Конформізм.** Це схильність до впливу авторитетів. Особливо сильно конформізм може проявитися під час проведення експертизи як відкритих дискусій. Думка авторитетів придушує думку осіб, які мають високий рівень конформізму.

**Ставлення до експертизи** є дуже важливою характеристикою якості експерта під час вирішення цієї проблеми. Негативне чи пасивне ставлення фахівця до вирішення проблеми, велика зайнятість та інші чинники суттєво впливають на виконання експертами своїх функцій. Тому участь в експертизі має розглядатись як планова робота. Експерт повинен виявляти інтерес до цієї проблеми.

**Конструктивність мислення** – це прагматичний аспект мислення. Експерт повинен давати рішення, які мають властивість практичності. Облік реальних можливостей вирішення проблеми є дуже важливим при проведенні експертного оцінювання.

**Колективізм** – має враховуватися під час проведення відкритих дискусій. Етика поведінки людини у колективі у часто істотно впливає створення позитивного психологічного клімату і цим на успішність вирішення проблеми.

**Самокритичність** експерта проявляється при самооцінці ступеня своєї компетентності, а також при врахуванні думок інших експертів та ухваленні рішення щодо розглянутої проблеми.

Перелічені характеристики експерта досить повно описують необхідні якості, які впливають на результати експертизи. Однак їх аналіз вимагає дуже кропіткої та трудомісткої роботи зі збирання інформації та її вивчення. З іншого боку, зазвичай, частина показників експерта оцінюється позитивно, а частина – негативно. Виникає проблема узгодження характеристик та вибору експертів з урахуванням суперечливості їх якостей. Причому чим більше характеристик береться до уваги, тим важче прийняти рішення про те, що важливіше і що допустимо для експерта. Для усунення зазначеної проблеми необхідно сформулювати узагальнену характеристику експерта, яка враховує його найважливіші якості, з одного боку, і що допускає безпосередній її вимір, з іншого боку. Як таку характеристику можна прийняти достовірність

думок експерта, яка визначає його як “вимірювальний прилад”. Однак застосування такої узагальненої характеристики потребує інформації про минулий досвід участі експерта у вирішенні проблем. У ряді випадків такої інформації може бути. Достовірність оцінок експерта кількісно оцінюють за формулою:

$$D_i = N_i / N, (i = \overline{1, m})$$

де  $N_i$  - кількість випадків, коли  $i$ -й експерт дав рішення, прийнятність якого підтвердилася практикою,  $N$  - загальна кількість випадків участі  $i$ -го експерта у вирішенні проблем.

Вклад кожного експерта у достовірність оцінок усієї групи визначається за формулою:

$$D_i^{ог} = \frac{D_i}{\frac{1}{m} \sum_{i=1}^m D_i}, (i = \overline{1, m}),$$

де  $m$  – число експертів у групі. У знаменнику стоїть середня достовірність групи експертів.

### 10.3 Опитування експертів

Опитування експертів є заслуховування і фіксацію у змістовній і кількісній формі суджень експертів з вирішуваної проблеми. Проведення опитування є основним етапом спільної роботи груп управління та експертів. На цьому етапі виконуються такі процедури:

- організаційно-методичне забезпечення опитування;
- постановка завдання та пред'явлення питань експертам;
- інформаційне забезпечення роботи експертів

Вигляд опитування визначає різновид методу експертної оцінки. Основними видами опитування є: анкетування, інтерв'ювання, метод дельфи, мозковий штурм, дискусія.

Вибір того чи іншого виду опитування визначається цілями експертизи, сутністю вирішуваної проблеми, повнотою і достовірністю вихідної інформації, часом і витратами на проведення опитування. Розглянемо зміст та технологію проведення перелічених вище видів опитування.

**Анкетування.** Анкетування є опитуванням експертів у письмовій формі за допомогою анкет. В анкеті містяться питання, які можна класифікувати за змістом та типом. За змістом питання поділяються на три групи:

- об'єктивні дані про експерта (вік, освіту, посаду, спеціальність, стаж роботи тощо);
- основні питання щодо суті аналізованої проблеми;

- додаткові питання, що дозволяють з'ясувати джерела інформації, аргументацію відповідей, самооцінку компетентності експерта тощо.

За типом основні питання класифікуються на відкриті, закриті та з віялом відповідей.

Відкриті питання передбачають відповідь у довільній формі. Відкриті питання доцільно застосовувати у разі великої невизначеності проблеми. Цей тип питань дозволяє широко охопити проблему, виявити спектр думок експертів. Недоліком відкритих питань є можлива велика різноманітність та довільна форма відповідей, що суттєво ускладнює обробку анкет.

Закриті питання застосовують у разі розгляду чітко визначених двох альтернативних варіантів, коли потрібно по суті визначити міру більшості думок щодо цих альтернатив. Обробка закритих питань не викликає жодних труднощів.

Питання з віялом відповідей доцільно використовувати за наявності кількох досить чітко визначених альтернативних варіантів. Ці варіанти формують для орієнтації експертів за можливим колом напрямків у вирішенні проблеми. Для отримання детальнішої інформації з кожного питання можуть бути запропоновані порядкова та бальна шкали. Експерт з кожної відповіді вибирає значення порядкової та бальної оцінок. Наприклад, значеннями порядкової шкали може бути “дуже добре”, “добре”, “задовільно”, “незадовільно”, чи “значно”, “незначно”, “не впливає” тощо. Обробка анкет із питаннями цього типу за складністю займає проміжне місце між відкритими та закритими питаннями.

Якщо анкетування проводиться в кілька етапів, то доцільно при великій складності та невизначеності проблеми спочатку використовувати відкриті типи питань, а на наступних – з віялом відповідей та закриті типи.

Крім анкети експертам подається пояснювальна записка, в якій роз'яснюються цілі та завдання експертизи, надається необхідна експерту інформація, наводяться інструкції щодо заповнення анкет та необхідні організаційні відомості.

Інтерв'ювання – це усне опитування, яке проводиться у формі бесіди-інтерв'ю. Під час підготовки розмови інтерв'юер розробляє питання експерту. Характерною особливістю цих питань є можливість швидкої відповіді на них експертом, оскільки він практично не має часу на його обмірковування.

Тематика інтерв'ю може повідомлятися експерту заздалегідь, але конкретні питання ставляться безпосередньо у процесі розмови. Доцільно у зв'язку з цим готувати послідовність питань, починаючи від простого та поступово їх поглиблюючи та ускладнюючи, але водночас і конкретизуючи.

Перевагою інтерв'ю є безперервний живий контакт інтерв'юера з експертом, що дозволяє швидко отримати необхідну інформацію шляхом прямих та уточнюючих питань, залежно від відповідей експерта.

Недоліками інтерв'ю є можливість сильного впливу інтерв'юера на відповіді експерта, відсутність часу для глибокого продумування відповідей та великі витрати на опитування всього складу експертів.

Інтерв'юер повинен добре знати аналізовану проблему, вміти чітко формулювати питання, створювати невимушену обстановку та вміти слухати.

Метод Дельфи є багатотуровою процедурою анкетування з обробкою та повідомленням результатів кожного туру експертам, які працюють інкогніто по відношенню один до одного. Метод названий на ім'я грецького міста, в якому в давнину жив знаменитий оракул [19].

Відомі приклади застосування методу Дельфи пов'язані з постановкою питань, що вимагають відповідей числової оцінки параметрів.

У першому турі опитування методом Дельфи експертам пропонуються питання, куди вони дають відповіді без аргументування. Отримані від експертів дані обробляються з метою виділення середнього чи медіани та крайніх значень оцінок. Експертам повідомляються результати обробки першого етапу опитування із зазначенням розташування оцінок кожного експерта. Якщо оцінка експерта сильно відхиляється від середнього значення, його просять аргументувати свою думку чи змінити оцінку.

У другому етапі (турі) експерти аргументують чи змінюють свою оцінку з поясненням причин коригування. Результати опитування на другому етапі обробляються та повідомляються експертам. Якщо після першого етапу проводилося коригування оцінок, результати обробки другого туру містять нові середні та крайні значення оцінок експертів. У разі сильного відхилення індивідуальних оцінок від середніх експерти мають аргументувати чи змінити свої судження, пояснивши причини коригування.

Проведення наступних турів здійснюється за аналогічною процедурою. Зазвичай після третього чи четвертого туру оцінки експертів стабілізуються, що є критерієм припинення подальшого опитування.

Ітеративна процедура опитування з повідомленням результатів обробки після кожного туру забезпечує краще узгодження думок експертів, оскільки експерти, які дали оцінки, що сильно відхиляються, змушені критично осмислити свої судження і докладно їх аргументувати. Необхідність аргументації чи коригування своїх оцінок означає, що метою експертизи є отримання повної узгодженості думок експертів. Кінцевим результатом може бути виявлення двох або більше груп думок, що відображають приналежність експертів до різних наукових шкіл, відомств або категорій осіб. Отримання такого результату є також корисним, оскільки дозволяє з'ясувати наявність різних точок зору та поставити завдання на проведення досліджень у цій галузі.

Під час проведення опитування у методі Дельфи зберігається анонімність відповідей експертів стосовно друг до друга. Це забезпечує виключення впливу конформізму, тобто. придушення думок з допомогою

“ваги” наукового авторитету чи посадового становища одних експертів стосовно іншим.

Для підвищення ефективності проведення експертизи методом Дельфі необхідно автоматизувати процес фіксації, обробки та повідомлення експертам інформації. Це досягається шляхом використання ЕОМ.

Мозковий штурм є груповим обговоренням з метою отримання нових ідей, варіантів вирішення проблеми. Мозковий штурм часто називають також мозковою атакою, методом генерації ідей. Характерною особливістю цього виду експертизи є активний творчий пошук принципово нових рішень у важких тупикових ситуаціях, коли відомі шляхи та способи вирішення виявляються непридатними. Для підтримки активності та творчої фантазії експертів категорично забороняється критика їхніх висловлювань [19].

Основні правила організації та методика проведення мозкового штурму полягають у наступному. Здійснюється підбір експертів у групу до 20-25 осіб, до якої включаються фахівці з проблеми і люди з широкою ерудицією і багатою фантазією, причому не обов'язково добре знають розглянуту проблему. Бажано включення до групи осіб, які займають однакове службове та суспільне становище, що забезпечує більшу незалежність висловлювань та створення атмосфери рівноправності.

Для проведення сеансу призначається ведучий, основним завданням якого є управління обговоренням для вирішення поставленої проблеми. Ведучий на початку сеансу пояснює зміст та актуальність проблеми, правила її обговорення та пропонує для розгляду одну-дві ідеї.

Сеанс триває приблизно 40-45 хвилин без перерви. Для виступу надається 2-3 хвилини, і вони можуть повторюватися. У кожному виступі експерти повинні прагнути висунути якнайбільше нових, можливо, на перший погляд фантастичних ідей або розвивати раніше висловлені ідеї, доповнюючи та поглиблюючи їх. Важливою вимогою до виступів є конструктивний характер ідей та пропозицій. Вони мають бути спрямовані на вирішення проблеми. Ведучий і всі члени групи повинні своїми діями та висловлюваннями сприяти створенню загальної синхронно працюючої колективної думки, збудженню розумових процесів, що суттєво впливає на результативність обговорення.

У процесі генерування ідей та їх обговорення пряма критика заборонена, проте вона має місце у неявній формі та виражається у ступені підтримки та розвитку висловлювань.

Виступи експертів фіксуються шляхом стенографування або магнітофонного запису та після закінчення сеансу піддаються аналізу, який полягає у групуванні та класифікації висловлених ідей та рішень за різними ознаками, оцінкою ступеня корисності та можливості реалізації. Приблизно через добу – дві після проведення сеансу експертів просять повідомити, чи не виникли ще якісь нові ідеї та рішення. Експерименти показують, що й у процесі сеансу було створено хороша творча атмосфера з активною участю у роботі всіх експертів, після закінчення обговорення у мозку людини триває

процес генерації та аналізу своїх та інших пропозицій, який протікає як усвідомлено, а й підсвідомо. В результаті зіставлення висловлювань, проведення аналогій та узагальнення часто, приблизно через добу, експерти формулюють найцінніші пропозиції та ідеї. Тому збір інформації щодо можливих нових ідей сприяє підвищенню ефективності методу мозкового штурму.

Існує ряд різновидів мозкового штурму, в яких пропонується чергувати п'ятихвилинні штурми з обмірковуванням його результатів, чергувати періоди генерації з дискусіями та груповим прийняттям рішень, застосовувати послідовні етапи висування пропозицій та їх обговорення, включати до групи експертів "підсилювачів" та "подавлювачів".

Мозковий штурм застосовується на вирішення різноманітних прикладних проблем.

Дискусія. Цей вид експертизи широко застосовується практично для обговорення проблем, шляхів їх вирішення, аналізу різних чинників тощо. Для проведення дискусії формується група експертів трохи більше 20 людина. Група управління проводить попередній аналіз проблем дискусії з метою чіткого формулювання завдань, визначення вимог до експертів, їхнього підбору та методики проведення дискусії.

Сама дискусія проводиться як відкрите колективне обговорення проблеми, основним завданням якого є всебічний аналіз всіх факторів, позитивних та негативних наслідків, виявлення позицій та інтересів учасників.

Під час дискусії дозволяється критика.

Велику роль дискусії грає ведучий. Від його вміння створити творчу доброзичливу атмосферу, чітко виступити із постановкою проблеми, коротко та глибоко резюмувати виступи та, головне, вміло направити хід дискусії на вирішення проблеми, суттєво залежить ефективність результатів обговорення.

Дискусія може проводитись протягом кількох годин, тому необхідно визначити регламент роботи: час на доповідь ведучого та виступи, проведення перерв. Слід пам'ятати, що під час перерв дискусія триває, тобто. мають місце кулуарні обговорення. У зв'язку з цим не слід робити перерви надто короткими, оскільки локальні обговорення дають позитивний ефект.

Результати дискусії фіксуються як стенограм чи магнітної записи. Після закінчення дискусії проводиться аналіз цих записів для чіткішого представлення основних результатів, виявлення відмінностей у думках. У дискусіях приблизно через добу після закінчення може збиратися додаткова інформація від експертів.

Розглянуті види опитування доповнюють один одного і до певної міри є взаємозамінними. Для генерації нових об'єктів (ідей, подій, проблем, рішень) доцільно застосовувати мозковий штурм, дискусії, анкетування та метод Дельфи (перші два тури).

Всебічний критичний аналіз наявного переліку об'єктів ефективно може бути проведений у формі дискусії. Для кількісної та якісної оцінки властивостей, параметрів, часу та інших характеристик об'єктів застосовуються анкетування та метод Дельфі. Інтерв'ювання доцільно використовуватиме уточнення результатів, отриманих іншими видами експертизи.

#### 10.4 Обробка експертних оцінок

Після проведення опитування групи експертів здійснюється опрацювання результатів. Вихідною інформацією для неї є числові дані, що виражають переваги експертів, та змістовне обґрунтування цих переваг. Метою обробки є отримання узагальнених даних та нової інформації, що міститься у прихованій формі в експертних оцінках. На основі результатів обробки формується вирішення проблеми [19].

Наявність як числових даних, так і змістовних висловлювань експертів призводить до необхідності застосування якісних та кількісних методів обробки результатів групового експертного оцінювання. Питома вага цих методів залежить від класу проблем, вирішуваних експертним оцінюванням. Представимо та проведемо опис методів опрацювання проблем першого класу, що характеризуються достатнім інформаційним потенціалом. Ці проблеми найпоширеніші у практиці прийняття рішень.

Залежно від цілей експертного оцінювання під час обробки результатів опитування вирішують такі основні задачі:

- визначення узгодженості думок експертів;
- побудова узагальненої оцінки об'єктів;
- визначення залежності між судженнями експертів;
- визначення відносних ваг об'єктів;
- оцінка надійності результатів експертизи

Визначення узгодженості оцінок експертів необхідне підтвердження правильності гіпотези у тому, що експерти є досить точними вимірниками, і виявлення можливих угруповань у експертній групі. Оцінка узгодженості думок експертів провадиться шляхом обчислення кількісного заходу, що характеризує ступінь близькості індивідуальних думок. Аналіз значень міри узгодженості сприяє виробленню правильного судження про загальний рівень знань з розв'язуваної проблеми та виявлення угруповань думок експертів, зумовлених різницею поглядів, концепцій, існуванням наукових шкіл, характером професійної діяльності тощо.

Завдання побудови узагальненої оцінки об'єктів за індивідуальними оцінками експертів виникає за групового експертного оцінювання. Якщо експерти проводили оцінку об'єктів у кількісній шкалі, то завдання побудови групової оцінки полягає у визначенні середнього значення медіани оцінки. При вимірі в порядковій шкалі методом ранжирування чи парного

порівняння метою обробки індивідуальних оцінок експертів є побудова узагальненого впорядкування об'єктів на основі опосередкування оцінок експертів.

Обробкою результатів експертного оцінювання можна визначати залежність між судженнями різних експертів. Виявлення цих залежностей дозволяє встановлювати міру близькості у думках експертів. Важливе значення також має визначення залежності між оцінками об'єктів, побудованими за різними показниками порівняння. Це дає можливість визначити пов'язані між собою показники порівняння та здійснити їх угруповання за рівнем взаємозв'язку.

При вирішенні багатьох завдань недостатньо здійснити впорядкування об'єктів за одним чи групою показників. Бажано мати кількісні значення відносної важливості об'єктів. Для вирішення цього завдання можна відразу застосувати метод безпосередньої оцінки. Однак це завдання за певних умов можна вирішити шляхом обробки результатів ранжувань або парних порівнянь групи експертів.

Оцінки об'єктів, одержувані результаті обробки, є випадкові величини, тому однією з важливих завдань є визначення їх достовірності, тобто. надійності результатів експертизи. Методи вирішення перелічених завдань розглядаються у відповідній літературі.

Обробка результатів експертизи вручну пов'язані з великими трудовими витратами (навіть у разі вирішення простих завдань упорядкування), тому її доцільно проводити з урахуванням обчислювальної техніки. Застосування ЕОМ висуває проблему розробки машинних програм, які реалізують алгоритми обробки результатів експертного оцінювання. При організації обробки результатів опитування слід ретельно проаналізувати трудомісткості вирішення завдань з урахуванням розробки математичного забезпечення для ЕОМ.

## **10.5 Визначення узгодженості експертів**

Як ілюстрацію методів рішення, перерахованих вище завдань, розглянемо завдання визначення узгодженості думок експертів.

При оцінці об'єктів експерти зазвичай розходяться в думках щодо проблеми, що вирішується. У зв'язку із цим виникає необхідність кількісної оцінки ступеня згоди експертів. Отримання кількісної міри узгодженості дозволяє більш обґрунтовано інтерпретувати причини розбіжності думок.

Оцінка узгодженості суджень експертів ґрунтується на використанні поняття компактності, наочне уявлення про яке дає геометрична інтерпретація результатів експертизи. Оцінка кожного експерта представляється як точка у певному просторі, де є поняття відстані. Якщо точки, що характеризують оцінки всіх експертів, розташовані невеликій відстані друг від друга, тобто. утворюють компактну групу, то, очевидно,

можна це інтерпретувати як хорошу узгодженість думок експертів. Якщо ж точки у просторі розкидані на значні відстані, то узгодженість думок експертів невисока. Можливо, що точки - оцінки експертів - розташовані у просторі так, що утворюють дві або кілька компактних груп. Це означає, що в експертній групі існують дві або кілька суттєвих точок зору на оцінку об'єктів.

Конкретизація викладеної ідеї оцінки узгодженості думок експертів провадиться залежно від використання кількісних чи якісних шкал вимірювання та вибору міри ступеня узгодженості.

При використанні кількісних шкал вимірювання та оцінки всього одного параметра об'єкта всі думки експертів можна подати як точки на числовій осі. Ці точки можна як реалізації випадкової величини і для оцінки угруповання і розкиду точок використовувати добре розроблені методи математичної статистики. Центр угруповання точок можна визначити як математичне очікування (середнє значення) або медіану випадкової величини, а розкид кількісно оцінюється дисперсією випадкової величини. Мірою узгодженості оцінок експертів, тобто. компактності розташування точок на числовій осі, може бути відношення середньоквадратичного відхилення до математичного очікування випадкової величини.

Якщо об'єкт оцінюється кількома числовими параметрами, то думка кожного експерта представляється як точка у просторі параметрів. Центр групування точок знову визначається як математичне очікування вектору параметрів, а розкид точок - дисперсією вектору пара метрів. Мірою узгодженості суджень експертів служить у разі сума відстаней оцінок від середнього значення, віднесена до відстані математичного очікування початку координат. Мірою узгодженості може також бути кількість точок, розташованих у радіусі середньоквадратичного відхилення від математичного очікування, до кількості точок. Різні методи визначення узгодженості кількісних оцінок на основі поняття компактності розглядаються в теорії угруповань та розпізнавання образів.

При вимірюванні об'єктів у порядковій шкалі узгодженість оцінок експертів у вигляді ранжувань або парних порівнянь об'єктів також ґрунтується на понятті компактності.

При ранжуванні об'єктів як міру узгодженості думок групи експертів використовується дисперсійний коефіцієнт конкордації (коефіцієнт згоди) [19].

Розглянемо матрицю результатів ранжування  $m$  об'єктів групою з експертів

$$d ||r_{is}|| (s = \overline{1, d}; i = \overline{1, m}),$$

де  $r_{is}$  - ранг, що присвоюється  $s$ -м експертом  $i$ -му об'єкту. Складемо суми рангів за кожним рядком. В результаті отримаємо вектор із компонентами

$$r_i = \sum_{s=1}^d r_{is} (i = \overline{1, m}).$$

Розглянемо величини  $r_i$  як реалізації випадкової величини і знайдемо оцінку дисперсії. Як відомо, оптимальна за критерієм мінімуму середнього квадрата помилки оцінка дисперсії визначається формулою:

$$D = \frac{1}{m-1} \sum_{i=1}^m (r_i - \bar{r})^2, \quad (7.1)$$

де  $\bar{r}$  - оцінка математичного очікування, рівна

$$\bar{r} = \frac{1}{m} \sum_{i=1}^m r_i \quad (7.2)$$

Дисперсійний коефіцієнт конкордації визначається як відношення оцінки дисперсії (7.1) до максимального значення цієї оцінки:

$$W = D / D_{\max} \quad (7.3)$$

Коефіцієнт конкордації змінюється від нуля до одиниці, оскільки:

$$0 \leq D \leq D_{\max}.$$

Максимальне значення дисперсії дорівнює:

$$D_{\max} = d^2 (m^3 - m) / 12 (m - 1). \quad (7.4)$$

Введемо позначення:

$$S = \sum_{i=1}^m \sum_{s=1}^d (r_{is} - \bar{r})^2$$

Використовуючи (7.5), запишемо оцінку дисперсії (7.1) як

$$D = 1/(m - 1) \times S \quad (7.6)$$

Підставляючи (7.4), (7.6) (7.3) і скорочуючи на множник  $(m - 1)$ , запишемо остаточний вираз для коефіцієнта конкордації:

$$W = 12 \times S / d^2 \times (m^3 - m) \quad (7.7)$$

Ця формула визначає коефіцієнт конкордації у разі відсутності пов'язаних рангів.

Якщо ранжування є пов'язані ранги, то максимальне значення дисперсії в знаменнику формули (7.3) стає менше, ніж за відсутності пов'язаних рангів. Доведено [5], що за наявності пов'язаних рангів коефіцієнт конкордації обчислюється за такою формулою:

$$W = 12 \times S / d^2 \times (m^3 - m) - d \times \sum_{s=1}^d T_s, \quad (7.8)$$

де:

$$T_s = \sum_{s=1}^d T_s. \quad (7.9)$$

У формулі (7.9)  $T_s$  – показник пов'язаних рангів у  $s$ -му ранжуванні,  $H_s$  – число груп рівних рангів у  $s$ -му ранжируванні,  $h_k$  – число рівних рангів у  $k$ -й групі пов'язаних рангів при ранжуванні  $s$ -м експертом. Якщо збігаються рангів немає, то  $H_s = 0$ ,  $h_k = 0$  і, отже,  $T_s = 0$ . І тут формула (7.8) збігається з формулою (7.7).

Коефіцієнт конкордації дорівнює 1, якщо всі ранжування експертів однакові, і дорівнює нулю, якщо всі ранжування різні. Коефіцієнт конкордації, що обчислюється за формулами (7.7) і (7.8), є оцінкою істинного значення коефіцієнта і, отже, є випадковою величиною.

Для визначення значущості оцінки коефіцієнта конкордації необхідно знати розподіл частот для різних значень числа експертів  $d$  та кількості об'єктів  $m$ . Розподіл частот для  $W$  при різних значеннях  $m$  та  $d$  може бути визначений за відомими статистичними таблицями. При числі об'єктів  $m > 7$  оцінка значущості коефіцієнта конкордації може бути здійснена за критерієм  $\chi^2$ .

Розмір  $d^*(m - 1) W$  має  $\chi^2 =$  розподіл з  $\nu = m - 1$  ступенями свободи.

За наявності пов'язаних рангів  $\chi^2 =$  розподіл з  $\nu = m - 1$  ступенями свободи має значення:

$$\chi^2 = 12 \times S / d \times m \times (m + 1) - \frac{1}{m-1} \sum_{s=1}^d T_s, \quad (7.10)$$

Поряд з дисперсійним коефіцієнтом конкордації використовується як міра узгодженості суджень експертів ентропійний коефіцієнт конкордації.

Приклад. Результати ранжирування шести об'єктів ( $O_1, O_2, \dots, O_6$ ) п'ятьма експертами ( $E_1, E_2, \dots, E_5$ ) представлені в табл. 7.1.

Таблиця 10.1 – Ранжирування об'єктів

	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$
$O_1$	1	2	1,5	1	2
$O_2$	2,5	2	1,5	2,5	1
$O_3$	2,5	2	3	2,5	3
$O_4$	4	5	4,5	4,5	4
$O_5$	5	4	4,5	4,5	5,5
$O_6$	6	6	6	5	5,5

Обчислимо коефіцієнт конкордації і зробимо оцінку його значимості. Середнє значення  $\bar{r}$  за формулою (7.2) дорівнює:

$$\bar{r} = \frac{1}{m} \sum_{i=1}^m \sum_{s=1}^d r_{is} = \frac{1}{6} \sum_{i=1}^6 \sum_{s=1}^5 r_{is} = 17,5$$

Величина  $S$  відповідно до формули (7.5) дорівнює:

$$S = \frac{1}{6} \sum_{i=1}^6 (\sum_{s=1}^5 r_{is} - 17,5)^2 = 361$$

Оскільки в ранжування є пов'язані ранги, то обчислення коефіцієнта конкордації виконаємо за формулою (7.8).

Попередньо обчислимо величини  $T_s$ , використовуючи формулу (7.9). В даному випадку з таблиці 7.1 випливає, що в ранжируванні експертом  $E_1$  є одна група пов'язаних рангів, тому  $H_1 = 1$ . У цій групі тримається два пов'язаних рангу, рівних 2,5, тому  $h_1 = 2$ . Звідси  $T_1 = 23 - 2 = 6$ . Аналогічним чином обчислюємо  $T_2 \div T_5$ :

$$T_2 = 3^3 - 3 = 24; T_4 = 2^3 - 2 + 2^3 - 2 = 12;$$

$$T_3 = 2^3 - 2 + 2^3 - 2 = 12; T_5 = 2^3 - 2 = 6.$$

Підставляючи значення  $T_s$ ,  $S$  і  $m = 6$ ,  $d = 5$  у формулу (7.8) і роблячи обчислення, отримуємо:

$$W = 12 \times 361 / 5^2 \times (6^3 - 6) - 5 \times 60 = 0,874.$$

Оцінимо значущість коефіцієнта конкордації. У разі число ступенів свободи  $\nu = 5$ . Табличне значення  $\chi^2$  для  $\nu = 5$  і 5% рівня значимості  $\chi^2$  табл. = 11,07. Підставляючи значення величини формулу (7.10), отримуємо

$$\chi^2 = 12 \times 361 / 5^2 \times 7 - 0,2 \times 60 = 21,8.$$

Оскільки  $11,07 < 21,8$ , то гіпотеза про згоду експертів у ранжуваннях приймається.

Експертний аналіз є важливим інструментом у прийнятті рішень, особливо в ситуаціях, коли даних недостатньо для використання лише кількісних методів. Процес включає збір, обробку та використання знань від фахівців для оцінки складних проблем. Для ефективного застосування експертного аналізу важливо правильно організувати роботу з експертами, систематизувати отримані знання та інтегрувати їх в автоматизовані системи підтримки прийняття рішень.

## ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ

1. Описати етапи методу експертних оцінок (від постановки задачі до агрегації результатів). Які обмеження і джерела похибок має цей метод?
2. Які критерії та підходи до відбору експертів ви застосуєте для проєкту з оцінки ризиків впровадження КІС?
3. Наведіть порівняння форм методів опитування експертів (індивідуальні інтерв'ю, анкетування, Delphi) - переваги й недоліки кожного.
4. Описати методи обробки експертних оцінок: просте середнє, вагове середнє, метод консенсусу Delphi, методи байєсової агрегації - коли яку агрегацію застосувати?
5. Як оцінити узгодженість експертів (наприклад, використовуючи коефіцієнт Кендалла  $W$  або інші статистичні міри)? Інтерпретуйте значення  $W = 0.7$  у контексті групової оцінки.

## **Тема 11. Евристичні методи аналізу і оптимізації систем**

### **11.1 Класифікація евристичних методів.**

У таких сферах як інформаційні технології прийняття рішень та управління, часто стикаються з задачами, для яких неможливо знайти точне рішення через складність або невизначеність. В таких випадках на допомогу приходять евристичні методи - методи, які дозволяють знаходити "добре достатнє" рішення за обмежений час, навіть якщо воно не є оптимальним [29].

Евристика - це прийом або стратегія для пошуку рішень у складних або невизначених ситуаціях. Вона заснована на досвіді, інтуїції та практичних спостереженнях, а не на строгих математичних розрахунках. У системному аналізі евристичні методи використовуються для аналізу та оптимізації складних систем, особливо коли точне математичне рішення важко отримати або воно не може бути знайдене через великий обсяг обчислень.

Існують різні типи евристичних методів, які застосовуються в залежності від типу проблеми та галузі застосування.

Різні автори по-різному характеризують кількість існуючих евристичних методів. Одні автори називають кілька базових, інші говорять про сотні методів. У стандарті РМВОК як евристичні (групові творчі методи) запропоновані методи: мозкового штурму, номінальних груп, Дельфи, складання інтелект карт. Також пропонуються для використання методи, які не віднесені до евристичних: експертні оцінки, фокус-групи, семінари за участю модератора, пошук альтернативи, контрольні картки, матричні діаграми, метрики якості, контрольні списки якості. Пошук альтернативи у стандарті РМВОК згадується як метод, водночас інші автори зазначають, що він складається з набору методів – мозковий штурм, всебічний розгляд питання, парні порівняння, подібність.

З методологічної точки зору завжди можна розділити будь-які методи на загальні та приватні, але чи є метод перетворення зворотного поступального руху на обертальний метод чи це лише евристичний прийом?

Евристичні методи можна поділити на прості та складні: «До простих зараховують способи постановки, рішення, реалізації винахідницького завдання, що містять елементарні операції, що застосовуються у певних типових ситуаціях. Такі, наприклад, метод змішування інгредієнтів речовини, метод застосування промінних гнучких елементів для з'єднання технічних об'єктів або їх частин і т.д. Складні методи містять елементи кількох простих. Так, метод поетапної мозкової атаки містить елементи зворотної мозкової атаки, прямої мозкової атаки, подвійної мозкової атаки та мозкової атаки експертів. Прості та складні методи винахідництва зазвичай застосовуються для виконання певної стадії або кроку творчого процесу винахідника»

Таблиця 11.1 – Класифікація евристичних методів

Загальні евристичні методи	Приватні евристичні методи
Методи евристичної аналогії	Метод біомеханіки Метод збільшення чи зменшення розмірів Метод пристосування природних конструкцій та речовин для технічних цілей
методи евристичної інверсії	Метод інвертування Метод інверсії шкідливих сил у корисні Методи евристичної інверсії форми технічного об'єкта
Методи евристичного комплексу	Метод агрегування Метод уніфікації Метод модульних елементів Метод легування Методом збільшення кількості одночасно виконуваних функцій Метод компаундування Метод резервування

Іншу класифікацію евристичних методів запропоновано В.М. Засничком. «Перша група – це так звані евристичні (інтуїтивні, або ірраціональні) методи, які спираються на активізацію творчої діяльності людини та розвиток її творчих здібностей на основі розвитку інтуїтивних процедур діяльності, фантазії, аналогій та ін.

Друга група методів заснована на використанні оптимальної логіки аналізу технічного або іншого об'єкта, що вдосконалюється, закономірностей його розвитку. Це раціональні, чи логічні, методи вирішення творчих завдань, у яких переважають логічні правила аналізу, порівняння, узагальнення, класифікації, індукції, дедукції тощо». До першої групи віднесемо методи: "мозкова атака", синектика, метод евристичних питань, метод емпатій, метод порівняльних стратегій, асоціативні методи; до другої групи: морфологічний аналіз, метод багатовимірних матриць, алгоритм вирішення винахідницьких завдань, функціонально-фізична методика конструювання»

Більшість класифікацій має описовий характер і не дають ключа до створення нових евристичних методів. Крім того, існує різночитання, що вважати методом, що прийомом, які методи є приватними (спеціальними) і що являють собою загальні методи, що включають приватні. Немає і чіткої класифікації евристичних методів застосування їх на різних стадіях проектування.

Таблиця 11.2 – Класифікація евристичних методів

№	Стадія творчого процесу	Основні методи
1	Підготовка	пошуку інформації виявлення суспільних потреб прогнозування вибору теми
2	Задум	аналізу інформації постановки задачі визначення поля розв'язання вибору засобів розв'язання задачі
3	Пошуки рішення	Генерування ідей Апробація ідей Верифікація ідей
4	Реалізація	Конкретизація та оформлення рішення Досвідчена перевірка рішення Освоєння, поширення та пропаганда рішення

Важливим є також розподіл евристичних методів їх застосування для різних груп процесів управління проектами. Евристичні методи, в першу чергу, використовуються в управлінні змістом проекту, а саме в групі процесів планування, це методи: фокус-групи, семінари за участю модератора, мозковий штурм, метод Дельфі, складання інтелект-карт, пошук альтернатив. Найбільш широко евристичні методи використовують у управлінні якістю проекту групи процесів планування: контрольні карти, мозковий штурм, методи номінальних груп, матричні діаграми, метрики

якості, контрольні списки якості. У групі процесів моніторингу та управління: метрики якості, контрольні списки якості, контрольні картки. У публікаціях з управління термінами проєкту в оцінці ресурсів операцій необхідним вважається використання однієї з таких методів - експертна оцінка та пошук альтернатив. Загалом, немає такої сфери управління проєктами, де не згадувалися б евристичні методи, принаймні, метод експертних оцінок [30].

Можна побачити, що для вирішення однієї проблеми використовуються схожі за змістом методи. Так, в управлінні змістом проєкту методи: фокус-групи, семінари за участю модератора та мозковий штурм передбачають колективне обговорення з метою пошуку ідей. А в управлінні якістю проєкту схожими є методи, що визначають критерії якості: метрики якості, контрольні списки якості, контрольні картки. Доцільно, схожі за змістом та напрямом застосування методи розділити на групи. Нами запропоновано методи пошуку ідей об'єднати під назвою «методи активізації творчої діяльності». Методи контролю визначають критерії якості, доречно поєднати під назвою «методи оцінки варіантів конструктивного рішення».

Є ще одна група методів, яка передбачає зіставлення різних факторів чи елементів з метою пошуку закономірностей, або коли потрібна зміна стратегії. Подібні методи негаразд поширені, але є найпотужнішими. Насамперед, це матричні діаграми, а також серед згаданих група методів пошук альтернативи. Методи, які мають таку спрямованість, віднесемо до «метод дослідження структури проблеми».

## **11.2 Класифікація евристичних прийомів**

У визначенні евристичних методів ми вказали, що метод складається з прийомів. Розглянемо існуючі визначення поняття евристичний прийом «евристика, еристика – сукупність прийомів дослідження; вчення про такі прийоми. Відповідно до даного визначення еристика це сукупність методів, а не прийомів, а ось з визначенням еристики – прийоми, мабуть, можна погодитися.

Термін «евристики» зустрічається в кібернетиці: «з цього погляду суть еристики проєктувальника полягає в тому, що вони є прийомами, що скорочують кількість можливих виборів шляхів у лабіринті, тобто кількість можливих варіантів розв'язання». У програму закладаються прийоми, що обмежують кількість способів вирішення, що перебираються, і полегшують пошуки потрібного. Такі прийоми в кібернетиці названі "евристики", а програми, що розробляються - евристичними».

Термінологічно вживання поняття «евристика» стосовно прийому вважаємо виправданим. Питання полягає в наступному, наскільки евристичним є прийом зменшення перебору варіантів. Звернемося до поняття евристичного програмування.

Під евристичним програмуванням (моделюванням) розуміють еволюційне моделювання, яке є розширеною модифікацією статистичного моделювання, за рахунок зміни в процесі моделювання алгоритму, який еволюціонує: «у процесі моделювання статистично вдосконалюється (прогресивно еволюціонує) сам алгоритм, відповідно до якого імітуються процеси функціонування».

Еволюційне моделювання може бути реалізовано у вигляді генетичного та еволюційного програмування. У генетичному програмуванні особини (проектвані об'єкти) є програми. Зручно представляти ці програми як дерев, де функції – внутрішні вузли, яких, як вхідних параметрів, приєднані під дерев. Листям такого дерева будуть константи, вхідні параметри завдання або директивні команди програми.

Таблиця 11.3 – Класифікація методів проектування за кібернетичним принципом

<b>Генерація нових рішень</b>	<b>Лабіринтні</b>
Морфологічні карти	Системні випробування
Синектика	Анкетні випробування
Мозковий штурм	Упорядкований пошук
Вибір шкал виміру	Мережа взаємозв'язку
Ліквідація тупикових ситуацій	Матриця взаємодій
Аналіз взаємопов'язаних областей рішення (AIDA)	Трансформація системи
	Контрольні переліки
Проектування нововведень шляхом усунення кордонів	Ранжування та зважування
	Індекс надійності за Квіркком
Визначення компонентів по Александеру	Фундаментальний метод проектування Метчетта (FDM)
	Вартісний аналіз

Над особинами (проектваними об'єктами) можуть здійснюватися дві операції:

- кросинговер – підміна одного з піддерев першого батька на яке-небудь подерево другого батька;

- мутація – випадкове зміна однієї з вузлів дерева (наприклад, зміна функції чи константи).

Відмінною рисою еволюційного програмування є операція кросингвера. Тут нас, насамперед цікавлять можливість випадкових змін параметрів чи структури об'єкта, Спочатку з'являється набір різних об'єктів, а потім проводиться їх порівняльний аналіз, Це досить пряма аналогія з методом гірлянд і асоціацій, завдання якого дати можливість підсвідомості людини генерувати випадкові комбінації елементів. Евристичним прийомом – евристиккою у разі є генерація нових варіантів структури об'єкта, створених випадковим комбінуванням.

Для завдань машинобудування проєктовані об'єкти можуть бути моделі, не обов'язково представлені у вигляді програм, а реалізовані у вигляді розрахункової схеми, підготовленої для досліджень, за допомогою програм інженерного аналізу САЕ.

Іншим прикладом евристичного прийому, який використовується в кібернетиці, є концептуальні евристики. Основним механізмом пошуку рішення в концептуальних евристичних вважається генерування безлічі таких шляхів вирішення задачі, серед яких з великою ймовірністю міститься і результативний шлях. При аналізі вихідної ситуації та співвіднесенні її з результуючою проєктувальник не просто збирає інформацію, необхідну для вирішення завдання, а будує (навіть часом не усвідомлюючи цього) структуровану модель проблемної ситуації. Виокремлюючи, у своїй, у вихідній інформації важливі елементи і формуючи, з їхньої основи, узагальнені елементи, і навіть, відносини з-поміж них.

Такі узагальнені елементи та відносини названі концептами, звідки отримала назву і сама теорія, що розглядається. Концепти грають основну роль осмисленні вихідної ситуації, створенні її моделі та уявної роботи з моделлю. Відповідно до концептуальної теорії набір концептів універсальний, і йому відповідають наявні в людини механізми обчислення, трансформації та формування відносин.

У кібернетиці одним із основних є метод метасистемних переходів. Кожен метасистемний перехід можна як об'єднання низки підсистем нижнього рівня і формування системи високого рівня, яка може бути включена як підсистема в наступний метасистемний перехід.

У педагогіці, також існує тлумачення терміна евристики як прийому: «евристики – прийом чи сукупність логічних прийомів у вирішенні завдань, виконання теоретичних досліджень, конструювання моделей».

У психолого-педагогічних дослідженнях зазвичай виділяють такі особливості евристичного прийому: евристичні прийоми задовольняють принцип редуції (відомості) під цілей; евристики обмежують перебір; на відміну від алгоритмів евристики здатні «відвести убік»; вони не забезпечують досягнення гарантованого успіху; використання евристики високоефективна; евристичні прийоми можна як теорію поведінки людини під час вирішення завдань. Поряд з терміном евристичний прийом у

педагогіці використовується поняття розумовий прийом як: «система процесів або операцій аналізу, синтезу, абстрагування та узагальнення, а також інших спеціально організованих для вирішення завдань-проблем певного типу та різного ступеня спільності».

Вказівки Д. Пойа щодо структури деяких конкретних евристичних прийомів та сформульовані ним правила пошуку рішень є однією з перших спроб представити метод у вигляді послідовності прийомів. Л. Ларсон запропонував свою систему евристичних прийомів, всього їх дванадцять:

- досліджуй частинами;
- намалюй картинку;
- формулюй еквівалентну проблему;
- модифікуй;
- вибери ефективну систему позначень;
- скористайся симетрією,
- скористайся періодичністю;
- поділяй на випадки;
- звертай дії;
- міркуй від протилежного,
- шукай контр приклади;
- розглянь екстремальні випадки; узагальни;
- розглянь кілька моделей задачі.

Практичне значення має класифікація за операційним складом, прийоми поділяються на: елементарні - одне нерозчленовану дію, як лінійної послідовності операцій та складні прийоми – сукупність кількох послідовностей дій, кожна з яких може здійснюватися в лінійному порядку, але незалежно одне від іншого. З такою класифікацією можна погодитись, але не прийомів, а методів. Тому що ми встановили, що метод складається із евристичних прийомів.

У зв'язку з використанням у педагогіці поняття операційного складу методу, частина дослідників вважають, що аналіз, синтез, порівняння, абстрагування, узагальнення, класифікація, систематизація, аналогія є операціями. Вважаємо, що це прийоми хоча термін операційний склад має право існування і склад методу можуть входити як прийоми, і операції.

Оригінальна класифікація евристичних прийомів розроблена Смержевським [31] (табл. 11.5). Усі евристики поділені на чотири класи. Евристики – підказки та евристики - прийоми розумової діяльності, без сумніву є евристичні прийоми, що стосується евристики - стратегій, це прийоми, а методи. Визначення евристики як прийоми вирішення певних класів завдань, які не піддаються алгоритмізації, але можуть бути виражені у вигляді послідовності дій, дуже загальне під це визначення можуть підпадати і методи та процес проектування в цілому.

Таблиця 11.5 – Класифікація прийомів проектування

Евристики - підказки	Евристики - евроритми	Евристики - прийоми думкової діяльності	Евристики - стратегії
евристики як різні способи (графічні схеми, друковані інструкції, вказівки вчителя, наочні матеріали, відомості тощо), які містять інформацію, застосування якої полегшує розв'язання задачі;	евристики як прийоми вирішення певних класів завдань, що не піддаються алгоритмізації. Це евристики можуть бути виражені у вигляді послідовності дій, яка не є строго детермінованою: евристичні схеми, поради-орієнтири, загальні схеми-вказівки;	евристики як розумові прийоми правдоподібного характеру;	евристики як специфічні розумові прийоми, що становлять пошукові стратегії.

У дослідженні запропоновано класифікацію, що підрозділяє прийоми на загальні та спеціальні. До загальних прийомів віднесено індукція, акцентуація, трансляція та ін. Ці прийоми співвідносяться з прийомами уявної діяльності, проте не ясно, чому до цього переліку не входять такі прийоми як - аналіз, синтез, аналогія та інші, загальноприйняті в психолого-педагогічній літературі. (Табл. 11.6).

Наведені приклади спеціальних евристичних прийомів: виділення підзавдання, розгляд окремого випадку, граничний перехід, пошук контрприкладу, заслуговують на увагу, хоча, в основному, відносяться до математики. Цікаво, що спеціальні евристичні прийоми прямо зіставлені із загальними.

Таблиця 11.6 – Класифікація евристичних прийомів

Загальні евристичні прийоми					
Акцентуація	Варіювання об'єкта	Трансляція	Реверсія	Індукція	Варіювання середовища
Спеціальні (предметні) евристичні прийоми					
виділення підзавдання виділення ключової фігури виділення властивостей (наприклад, інваріанту); інші	додавання допоміжного елемента розгляд окремого випадку граничний перехід інші	перформулювання перехід від геометричних об'єктів до співвідношення алгебри зведення до раніше вирішеної задачі інші	доказ «від протилежного» пошук контрприкладу рух від вимоги завдання до її даних інші	пошук закономірностей аналогія між стереометричною ситуацією та планиметричною пошук загальної формули або узагальнюючого правила інші	введення системи координат розгляд відрізка як елемента різних фігур пошук наочних прикладів із практики інші

Таблиця 11.7 – Евристичні вказівки для різних типів стратегій

Тип стратегії	Евристичні вказівки
Стратегії функціонально-цільового аналізу:	Навіщо це потрібно зробити? (Аналіз потреб.) Що потрібно зробити? (Які цілі розв'язання задачі?) Чому слід це зробити? (Аналіз та синтез причин) Де це слід зробити? (Уточнення місця дії) Коли це можна зробити? (Час дії) За допомогою чого? (Засіб) Як це зробити? (Метод)
Стратегії аналізу протиріччя:	Проаналізуйте протиріччя, як воно є, тобто його вихідний стан. Сформулюйте, конкретизуйте суть протиріччя. Зміцніть протиріччя, тобто доведіть його до ступеня конфлікту. Розгляньте протиріччя динаміці, початку його виникнення. Осмисліть найімовірніші процедури вирішення протиріччя. Проаналізуйте, що станеться, якщо суперечність вирішуватиметься самопливом. Виявіть умову, за якої можна керувати процесом вирішення протиріччя.

Тип стратегії	Евристичні вказівки
Стратегії подолання бар'єру (перешкоди):	<p>Усуньте перешкоду.  Обійдіть перешкоду.  Дозвольте перешкоду.  Частково впливайте на перешкоду.  Зміцніть перешкоду.  Подолайте перешкоду на етапах.  Впливайте на перешкоду з несподівано новою позицією або принципово новими засобами.</p>
Стратегії використання інформації	<p>Використовуйте відому інформацію, що застосовується у вирішенні цього завдання.  Зберіть додаткову інформацію із суміжних наук.  Використовуйте досвід інших.  Перетворіть інформацію з урахуванням специфіки завдання.  Позбавтеся другорядної інформації.  Перевірте достовірність, точність, надійність інформації.  Використовуйте принципово нову та нову інформацію</p>
Стратегії пошуку ідеї, протилежної загальноприйнятій чи найбільш очевидної	<p>Якщо в процесі вирішення завдання всі прагнули:  Зменшити щось, а чи не краще збільшити?  Прискорити щось, а чи не краще сповільнити?  Розширити поле пошуку, а чи не доцільніше його, навпаки, звужити?  Розглянути явище у статиці, а чи не зробити це динаміці?  Поєднати елементи, а чи немає можливості залишити їх роз'єднаними?  Вирішити завдання відразу, а чи не краще вирішити її частинами?</p>
Стратегії пошуку ідеї, протилежної загальноприйнятій чи найбільш очевидної	<p>Якщо в процесі вирішення завдання всі прагнули:  Зменшити щось, а чи не краще збільшити?  Прискорити щось, а чи не краще сповільнити?  Розширити поле пошуку, а чи не доцільніше його, навпаки, звужити?  Розглянути явище у статиці, а чи не зробити це динаміці?  Проаналізувати минуле, а чи не краще осмислити, що станеться у майбутньому?  Поєднати елементи, а чи немає можливості залишити їх роз'єднаними?  Вирішити завдання відразу, а чи не краще вирішити її частинами?</p>

Тип стратегії	Евристичні вказівки
Стратегії оціночних суджень:	<p>Оцініть складність, складність вихідної ситуації.</p> <p>Уточніть критерії (ознаки), за якими будуть дані оціночні судження.</p> <p>Оцініть результати найважливіших етапів розв'язання задачі.</p> <p>Оцініть рівень ризику.</p> <p>Оцініть переваги та недоліки кожного варіанту рішення.</p> <p>Порівняйте та оцініть найбільш оригінальні варіанти вирішення задачі.</p> <p>Порівняйте еталон – ідеальний кінцевий результат – із найбільш оригінальним, оптимальним варіантом рішення.</p>
Стратегії прийняття рішення:	<p>Подумки програйте, уявіть найбільш оригінальне рішення задачі у його остаточному варіанті.</p> <p>Скасуйте рішення, але обґрунтуйте чому.</p> <p>Прийміть оригінальне, але тимчасове рішення.</p> <p>Проаналізуйте всі можливі рішення, продиктовані здоровим глуздом, і вибирайте з них найбільш ефективне.</p> <p>Проаналізуйте всі можливі рішення, які висуваються попри здоровий глузд, оцініть їх ефективність.</p> <p>Шукайте серію поетапних рішень.</p> <p>Прийміть остаточне рішення.</p>

Наприклад, загальному евристичному правилу індукція відповідають спеціальні: пошук закономірностей, аналогія між стереометричною ситуацією та планіметричною, пошук загальної формули або узагальнюючого правила.

Приєм аналогії між стереометричною і планіметричною ситуацією, можна трактувати як, частину загального методу аналогія, а не індукція. Пошук закономірностей чи узагальнюючого правила може бути частиною загального правила - аналіз, тут, напевно, мається на увазі пошук закономірності за аналогією, але тоді загальний метод - аналогія. Як видно питання зіставлення загальних та спеціальних евристичних прийомів навіть у рамках дисципліни математика не є вирішеним [32].

Нам здається, що цей перелік має значення як методу морфологічного аналізу, а й інших методів і може бути основою класифікації прийомів проектування.

Зазначена класифікація оформлена нами у вигляді таблиці 11.8

Таблиця 11.8 – Класифікація евристичних прийомів

<b>Тип прийому</b>	<b>Характеристика прийому</b>
Прийом аналогії	пошук аналога та використання всіх процедур виведення за аналогією.
Прийом дроблення	пошук компетентного складу системи, розчленування її підсистеми.
Прийом укрупнення	збільшення розмірів, показників, якісних характеристик системи
Прийом інверсії	зміна процедур діяльності на протилежні, звернення функцій, погляд на систему з протилежної точки зору, ніж загальноприйнята, заміна динаміки на статику і навпаки.
Прийом пристосування	адаптація системи або її окремих складових до зовнішніх умов, взаємодії нового і старого.
Прийом ідеалізації	пошук можливостей наближення системи або її складових до ідеального варіанту.
Прийом локалізації	пошуки можливостей тимчасового відокремлення частини системи, тимчасова зміна частини умов, тимчасове задоволення частини вимог задачі тощо.

На закінчення дамо своє визначення евристичного прийому. Евристичний прийом – це процедура, яка містить операції, що вимагають участі члена команди проекту.

### **11.3 Узагальнений метод активізації творчої діяльності**

Для використання евристичних методів в управлінні проектами необхідно дати чітку класифікацію описаних у літературі методів. Доцільно, щоб класифікація базувалася на аналізі їхньої структури. Для цього з кожного методу мають бути виділені евристичні прийоми, що у ньому використовуються. Такий аналіз дозволить вичленувати з величезної кількості описаних у літературі методів, які є змістовними та оригінальними. Наступним етапом відбору евристичних методів є можливість їх застосування в управлінні проектами. Отримавши перелік евристичних прийомів використовуваних у межах обраної групи методів, можна сформулювати один узагальнений метод, куди входять всі методи цієї групи. Знайдені узагальнені методи мають бути орієнтовані рішення завдань відповідних групам процесів управління проектами [33].

Для відбору евристичних методів використовуємо дані визначення евристичного методу і евристичного прийому.

Евристичний метод – це сукупність евристичних прийомів та процедур, що дозволяє інтенсифікувати та організувати мислення члена команди проєкту. Евристичний прийом – це процедура, яка містить операції, що вимагають участі члена команди проєкту. А також сформулюємо аксіому.

Аксіома. Евристичний метод повинен мати хоча б один евристичний прийом.

Існують вдалі літературні описи роботи менеджера проєкту, які претендують визначення метод. Сформульована аксіома дозволяє відсіяти методи беззмістовні, названі методами в силу претензій їх авторів та методи, які можуть бути представлені у вигляді алгоритму, що реалізується на комп'ютері.

Існує безліч варіацій методів мозкового штурму, аналогії, морфологічного аналізу, кожна з яких носить окрему назву. Найбільш відомим із методів активізації творчої діяльності є метод мозкового штурму. У літературі зустрічаються також методи мозкової атаки, прямого мозкового штурму, подвійного мозкового штурму, електронного мозкового штурму, масового мозкового та індивідуального мозкового штурму. Чи є перелічені методи, насправді окремими методами або різними назвами одного і того ж методу дозволить з'ясувати аналіз їх структури.

Різні автори по-різному характеризують кількість існуючих методів активізації творчої діяльності. Одні автори називають кілька базових, інші говорять про сотні методів. Найчастіше той самий метод у різних авторів фігурує під різними назвами чи кілька модифікацій однієї й тієї ж методу представлені як різні методи. Щоб розібратися який із описаних у літературі методів є оригінальним, а який лише варіацією відомого методу вимагаємо, щоб методи відрізнялися використовуваними прийомами.

Теорема. Якщо два евристичні методи містять одні й самі евристичні прийоми і відрізняються лише виконуваними процедурами, це модифікації однієї й тієї ж методу.

Спираючись на сформульовані нами визначення евристичного методу та прийому, спробуємо розібратися у структурі відомих методів.

Використовуючи аксіому і теорему, розберемо: які з відомих евристичних методів мають право називатися методами і які є оригінальними.

Найбільш відомим з таких методів є метод мозкового штурму (англ. brainstorming), який був розроблений Алексом Осборном в 1953 році. Осборном, ґрунтується на двох основних принципах - "відстрочка винесення вироку ідеї" і "з кількості народжується якість".

Метод мозкового штурму включає такі основні етапи:

- чітке формулювання мети та обмежень;
- генерація ідей;
- відбір кращого рішення, з експертних оцінок.

При цьому мають виконуватися такі принципи:

- ретельне формування складу учасників за чисельністю та спеціалізацією, визначення ведучого та розподіл ролей серед інших учасників;
- забезпечення максимальної свободи учасникам, надання слова кожному, повна заборона на критику висловлюваних ідей та будь-яку (у тому числі позитивну) оцінку;
- ієрархічне ведення обговорень: спочатку - максимально вшир, потім оцінка перспективності варіантів та відбір найкращих, потім знову "вшир".

Мозкова атака відрізняється наявністю двох окремих груп: учасники, що пропонують нові варіанти вирішення завдання та члени комісії, що обробляють запропоновані рішення. Метод прямого мозкового штурму на відміну класичного методу, передбачає процес формулювання проблеми та обмежень проходить у процесі мозкового штурму, причому із тим самим складом учасників. Метод масового мозкового штурму використовується на вирішення глобальних проблем і передбачає розбиття вихідної проблеми на блоки. Потім окремо в кожному блоку проводиться метод мозкового штурму. Наступний етап - збирання керівників груп та обговорення всіх ідей. Метод подвійного (парного) мозкового штурму відрізняється введенням критики ідей і включає три етапи: прямий мозковий штурм, обговорення, продовження висування ідей. Зворотний мозковий штурм використовується під час реалізації проєктів, які з багатьох етапів (елементів). У разі невдачі одного етапу – зрив усього процесу. Отже, найважливіше – переконатися у вірності кожного елемента.

Мета мозкового штурму – максимальне виявлення всіх недоліків. Етапи: складання списку існуючих, потенційних та можливих у майбутньому недоліків за допомогою мозкового штурму; їх ранжування. Єдина відмінна риса, часто згадуваного методу корабельної ради, полягає в тому, що висловлювання проводяться відповідно до ієрархії.

До недоліків даного методу та всіх його модифікацій відносяться недостатня формалізованість та надзвичайно сильна залежність результатів від підготовки та проведення. Ще одним істотним недоліком методу є участь в обговоренні великої кількості фахівців. Для одного проєктувальника можливість використання цього методу пов'язана із застосуванням електронного мозкового штурму (online brainstorming), який використовує інтернет-технології, а також метод індивідуального мозкового штурму. При індивідуальному мозковому штурмі ролі фіксатора, генератора та оцінювача ідей виконує одна людина.

Тривалість сеансу – 3-10 хвилин. Фіксація за допомогою ручки, ПК чи диктофона. Оцінка ідей має бути відкладена. Були проведені численні експериментальні дослідження, з метою порівняння кількості та якості ідей, створених групами в процесі мозкового штурму та людьми, що працюють індивідуально. Результати свідчать про те, що за умови правильного

застосування цієї техніки групи генерують більшу кількість значущих ідей, ніж окремі індивіди через відсутність синергетичного ефекту.

При розгляді модифікацій методу мозкового штурму можна побачити, що не всі етапи є обов'язковими і не всі принципи виконуються суворо. Так, при прямому мозковому штурмі формулювання проблеми та обмежень проходить безпосередньо в процесі обговорення. При індивідуальному мозковому штурмі не виконується принципу розподілу ролей серед учасників обговорення. А в методі подвійного мозкового штурму допускається критика ідей до закінчення процесу їх висування. Аналіз методу мозкового штурму дозволив вичленувати евристичні прийоми, які є спільними для всіх його численних модифікацій та відкинути рекомендації та вказівки, які не вносять істотних змін до його реалізації. Вважаємо, що метод мозкового штурму можна звести до трьох евристичних прийомів, що повністю відображають суть методу: колективне обговорення; рознесеність у часі висування ідей та його критики; використання експертних оцінок.

Мабуть, одне із найстаріших (1926 р.) евристичних методів - метод фокальних об'єктів (метод каталогу, метод випадкових об'єктів) автором якого є Ф. Кунці. Ідея методу полягає у перенесенні ознак випадково обраних об'єктів на вдосконалюваний об'єкт, який лежить як би у фокусі перенесення і тому називається фокальним. Незвичайні поєднання, що виникли, намагаються розвинути шляхом вільних асоціацій.

Метод фокальних об'єктів реалізується за таким планом: з умови завдання виділяють об'єкт (прототип), що підлягає удосконаленню (фокальний об'єкт), та уточнюють мету;

вибирають 3-4 випадкові об'єкти;

виписати кожному їх кілька характерних ознак (властивостей);

отримані ознаки переносять на прототип (фокальний об'єкт) та отримують нові поєднання;

нові поєднання розвивають шляхом вільних асоціацій. Фіксують усі цікаві ідеї.

оцінюють нові ідеї та відбирають найефективніші з погляду реалізації, формулюють завдання на розробку нових модифікацій об'єкта.

Недоліком даного методу є його непридатність при вирішенні складних завдань, тому що отримують лише прості поєднання, а також відсутність правил відбору та внутрішніх критеріїв оцінки одержуваних ідей.

Подальшим розвитком методу фокальних об'єктів є метод гірлянд випадковостей та асоціацій, розроблений Г.Я. Бушем. Він допомагає знайти велику кількість підказок для нових ідей шляхом створення асоціацій.

Метод гірлянд випадковостей та асоціацій реалізується за певним сценарієм та планом:

- визначають синоніми об'єкта;
- вибирають довільні об'єкти;
- складають комбінації з елементів гірлянди синонімів об'єкта та елементів гірлянди випадкових об'єктів;

- комбінації складають із двох елементів шляхом спроби об'єднання кожного синоніма об'єкта, що розглядається, з кожним випадковим об'єктом;
- складають перелік ознак випадкових об'єктів;
- визначають ознаки випадково вибраних об'єктів з можливою кількістю ознак протягом обмеженого відрізка часу;
- складають таблицю випадкових об'єктів та його ознак;
- генерують ідеї шляхом почергового приєднання до об'єкта та його синонімів ознак випадково вибраних об'єктів;
- далі відбувається генерування гірлянд асоціацій. Почергово із ознак випадкових об'єктів, виявлених раніше, генерують гірлянди вільних асоціацій. Генерування слід обмежити за часом чи кількістю елементів гірлянди;
- генерування нових ідей. До елементів гірлянд синонімів технічного об'єкта намагаються приєднати елементи гірлянд асоціацій;
- вибір альтернативи. На цьому кроці вирішують питання – продовження генерування гірлянд асоціацій;
- оцінка та вибір раціональних варіантів ідей.

Класифікація евристичних методів В.М. Заєнчик містить метод вільних асоціацій. Психологами доведено, що результативність творчої діяльності, особливо на етапі генерування нових ідей, суттєво підвищується, якщо широко використовувати асоціації, що у результаті породжують продуктивні ідеї вирішення проблеми. В.М. Заєнчик в обґрунтування методу зазначає, що: «У процесі зародження асоціацій встановлюються неординарні взаємозв'язки між компонентами розв'язуваної проблеми та елементами зовнішнього світу, включаючи компоненти колишнього досвіду творчої діяльності осіб, які беруть участь у колективному вирішенні проблеми, творчого завдання. Внаслідок процесу зародження нових асоціативних зв'язків і виникають творчі ідеї вирішення проблеми».

Принципи, на які слід спиратися у процесі застосування цього методу:

- вільних асоціацій;
- антиконформізму;
- відстроченого критичного аналізу.

Для посилення антиконформізму необхідно, щоб кожен член групи прагнув запропонувати та пропонував своє слово, поняття, яке має бути базисом для встановлення асоціативних зв'язків із процесом генерування нових ідей.

Метод вільних асоціацій, метод гірлянд випадковостей та асоціацій безсумнівно містять багато загальних положень, та є окремими методами чи варіантами – відкрите питання.

Автором методу синектики (1952) вважається Дж. Гордон, термін "синтетика" позначає "об'єднання різнорідних елементів". Творчий процес навіть окремої людини, з погляду Дж. Гордона, аналогічний до творчого

процесу колективу людей, які мають у сукупності різнобічну підготовку. Метод синектики отримав розвиток і теоретичне обґрунтування на роботах Г.Я. Буша.

Суть методу синектики представлено в наступному :

На перших етапах його застосування йде процес навчання "механізмів творчості". Частина цих механізмів автори методики пропонують розвивати навчанням.

Перші механізми називають – "операційними механізмами". До них відносять пряму, особисту та символічну аналогію.

Такі явища, як інтуїція, натхнення, абстрагування, вільні роздуми, використання можливостей, що не стосуються справи, застосування несподіваних метафор та елементів гри, вважають "неопераційними механізмами", розвиток яких не гарантується навчанням, хоча може вплинути на їх активізацію позитивного впливу.

При використанні даного методу необхідно впровадити та використати наступні принципи:

- відстрочка – першочергово відбувається пошук нових точок зору та перспективного напрямку рішень;

- автономність об'єкта аналізу - розпочинати обговорення не з вирішення приватних питань процесу пошуку рішення, а з аналізу загальних ознак, які вводять у ситуацію постановки проблематики, неодноразово уточнюючи її зміст;

- включення/виділення - виділення загального та специфічного, приватного для визначення та розгляду його як частина більшого;

- використання методу аналогій. Використовуються всі види аналогії: особиста (емпатія), пряма, фантастична та символічна;

- критичний відбір та оцінка ідей вирішення творчої задачі здійснюють у кілька етапів. На першому етапі дається аналіз кожної висунутої ідеї; на другому - ідеї відокремлюють в когорти, критично аналізують та відбирають найоригінальніші.

Ідея методу аналогій полягає у відокремленні проблеми та спробі її вирішення за допомогою ідей з інших сфер життя та науки. Буш виділяє методи евристичної аналогії, що ґрунтуються на природному прагненні людини до наслідування: «За допомогою цих методів винахідницькі завдання вирішуються шляхом розсуду аналогічних ситуацій у природі, техніці, суспільних та інших явищах та використання знайдених аналогій для усунення протиріч, що створили проблемну ситуацію».

Використання методу біомеханіки. П.Л. Чебишев наприкінці минулого століття запропонував "стопохідну машину", використовуючи принципи руху ніг коника», також описаний та впроваджений метод біоархітектури, який полягає у використанні аналогії з формами, архітектонікою та пропорціями живої природи для вирішення винахідницьких завдань [34].

Архітектор А. Карбовський застосував у житловому будівництві досвід бджіл у спорудженні воскових сот, які є ідеальною формою для монолітних конструкцій – стільникових стін, огорож, радіаторів та інше.

Ще один варіант, пов'язаний з аналогією, виділяють у вигляді самостійного методу - метод емпатії. Суть методу полягає у ототожненні людини з проєктованим об'єктом: «Людина як би зливається з об'єктом, об'єкту приписується поведінка, яка можлива у фантастичному варіанті» емпатія, або особиста аналогія, у вирішенні творчої задачі розуміється як ототожнення процесом, деякою системою – тому методу емпатії називають ще методом особистої аналогії.

До методів активізації творчої діяльності Джонс також відносить метод перемикання стратегії. Суть методу полягає в перемиканні пошуку на нові напрямки, коли проведені методи перемикання стратегії базуються на наступних принципах:

- визначення напряму дослідження з урахуванням аналізу спонтанних думок (спонтанні думки постійно фіксують);
- постійне порівняння обраного напряму дослідження із напрямом дослідження, що відповідає результатам аналізу спонтанних думок;
- зміна напряму досліджень на певному етапі, якщо стратегія, що реалізується, не дає результату.

До методів активізації творчої діяльності належить також метод морфологічного аналізу. Метою якого є розширення області пошуку рішень проєктної задачі. Метод полягає у заповненні морфологічних таблиць (матриць) з вибором можливих рішень. Спочатку в таблиці заноситься набір характеристик, яким має задовольняти елемент на аналізованому рівні проєктування. Потім для кожної характеристики перераховуються всі можливі, навіть часткові рішення. При цьому основний принцип пошуку рішення полягає у виборі, за деякими критеріями (масою, габаритами, вартістю) найкращого рішення з кожного ряду.

Морфологічний аналіз відомий також, як метод морфологічних карт та морфологічний ящик, який найбільш повно обґрунтував Ф. Цвік – розв'язувані проєктувальником проблеми, розділені на три класи:

- проблеми, на вирішення яких можна використовувати порівняно невелика кількість вже відомих елементів;
- проблеми, на вирішення яких потрібно використовувати ще невідомі нові елементи;
- проблеми великих чисел.

Ідея методу полягає у використанні матриць для опрацювання всіх можливих комбінацій елементів. Недоліком методу є громіздкість отриманої матриці, що містить велику кількість варіантів рішень, вибір з яких оптимального виявляється складним. Тому бажано, щоб кількість рядків та стовпців не перевищувала десяти.

Інтелект-карт (асоціативна карта, mind-maps) – спосіб зображення процесу загального системного мислення за допомогою схем. Інтелект-карта

є деревоподібною схемою, на якій зображені слова, ідеї, завдання або інші поняття, пов'язані гілками, що відходять від центрального поняття або ідеї. В основі цієї техніки лежить принцип «радіантного мислення», що відноситься до асоціативних розумових процесів, відправною точкою або точкою застосування яких є центральний об'єкт. Інтелект-карта використовуються для створення, візуалізації, структуризації та класифікації ідей, а також як засіб для навчання та прийняття рішень. Відомі такі рекомендації щодо складання інтелект-карток:

- зазвичай включає трохи більше 5-7 елементів чи груп елементів;
- має бути логічною: дотримані співвідношення між елементами та їх послідовність;
- включає різні візуальні образи;
- зазвичай симетрична;
- один з елементів є ключовим.

Оптимізація складних систем. У реальних системах часто є безліч змінних і параметрів, що впливають на результат. Евристичні методи дозволяють скоротити час на пошук рішень і знайти оптимальні варіанти у великих просторах можливих рішень.

Моделювання складних ситуацій. Евристичні методи використовуються для моделювання ситуацій, де кількість змінних або варіантів занадто велика для того, щоб використовувати стандартні математичні підходи. Наприклад, при плануванні ресурсів, розподілі задач або прогнозуванні результатів.

Евристичні методи аналізу і оптимізації є потужним інструментом для вирішення складних задач у системному аналізі. Вони дозволяють знаходити практично хороші рішення в обмежений час, коли точні методи або моделі не можуть бути застосовані через високі вимоги до обчислень або недостатність даних. Застосування евристичних підходів широко використовується в різних галузях, таких як ІТ, інженерія, економіка та бізнес, і дозволяє досягати ефективних результатів навіть у складних і невизначених умовах.

## **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Наведіть класифікацію евристичних методів (місцеві пошуки, метаевристики, стохастичні, детерміновані). Для кожного класу - типовий приклад алгоритму.

2. Описати основні евристичні прийоми (жадібні, релабелінг/перестановки, генетичні оператори, відбір за локальним поліпшенням). Наведіть приклад застосування жадібного прийому.

3. Поясніть ідею генетичного алгоритму: як формуються хромосоми, що таке кросовер і мутація, як здійснюється селекція? Які параметри найбільш критичні?

4. Що таке «узагальнений метод активізації творчої діяльності» (метод генерування ідей)? Наведіть один структурований приклад використання цього методу для пошуку рішень у проекті оптимізації процесу.

5. Обговоріть переваги та обмеження евристичних методів порівняно з точними (оптимізаційними) методами; у яких задачах доцільні евристики (масштаб, NP-важкість, час відповіді)?

## **Тема 12. Корпоративна інформаційна система**

### **12.1. Визначення та роль корпоративної інформаційної системи**

Корпоративна інформаційна система (КІС) - це інформаційна система, яка підтримує автоматизацію функцій управління на підприємстві (в корпорації) і постачає інформацію для прийняття управлінських рішень. У ній реалізована управлінська ідеологія, яка об'єднує бізнес-стратегію підприємства і прогресивні інформаційні технології [35].

Корпоративна інформаційна система (КІС) є важливим елементом сучасного підприємства, оскільки дозволяє забезпечити ефективну автоматизацію процесів управління та надання необхідної інформації для прийняття управлінських рішень. КІС сприяє інтеграції різних функціональних підсистем компанії, що дозволяє зберігати та обробляти інформацію на всіх рівнях організації - від операційного до стратегічного.

Цей розділ дає можливість розглянути ключові аспекти корпоративних інформаційних систем: їх роль, основні компоненти, функції та технології, а також важливість для стратегічного управління підприємством.

Суть системного підходу до проектування економічних ІС полягає в комплексному вивченні об'єкта керування як єдиного цілого на основі аналізу і синтезу. Аналіз припускає виділення ознак структуризації і декомпозиції системи. Економічна система є складною багатофункціональною. Тому існує безліч способів її декомпозиції. Традиційно економічну інформаційну систему розглядають як сукупність двох компонентів або підсистем - функціональної і забезпечувальної. Функціональна частина є моделлю системи керування об'єктом і, як будь-яка модель, вимагає чіткого математичного обґрунтування й опису. Забезпечувальна частина припускає такі види забезпечення, як математичне, технічне, програмне, інформаційне, лінгвістичне, організаційне, правове, ергономічне.

Корпоративна інформаційна система (КІС) - це інтегрована система, що включає в себе набір програмних та апаратних засобів для автоматизації бізнес-процесів підприємства і забезпечення ефективного управління всіма рівнями діяльності організації.

#### **Загальна характеристика корпоративних інформаційних систем**

Останнім часом в Україні на підприємствах різного масштабу широко почали впроваджуватись корпоративні інформаційні системи (КІС). Вони поступово витісняють традиційні автоматизовані системи управління підприємством (АСУП), що створювались у 80-х роках минулого століття на базі потужних обчислювальних центрів та тогочасної ідеології управління та опрацювання інформації.

Корпоративна інформаційна система - це інформаційна система, яка підтримує автоматизацію функцій управління і надає інформацію для

поглиблення знань та прийняття управлінських рішень. В ній реалізована сучасна управлінська ідеологія, яка поєднує бізнес-стратегію підприємства і прогресивні інформаційні технології

Сучасні КІС мають певні характеристики:

- масштабованість. Це одна з важливих характеристик КІС, оскільки вони повинні створюватись на масштабованій програмно-апаратній платформі (сервери, операційні системи, системи комунікації, СУБД). Оскільки варіантів конфігурації базового устаткування і програмного забезпечення може бути багато, то КІС має бути багатоплатформовою.

- багатоплатформність. В КІС виникає потреба в тому, щоб прикладна програма працювала на кількох апаратних і програмних платформах. При цьому мають бути забезпечені однакові інтерфейси та логіка роботи. Реалізувати прикладну програму одночасно в кількох середовищах нелегко. В зв'язку з цим з'явилися інтегровані програмні середовища розробки, які значно полегшують перенесення прикладних програм з одного середовища в інше. До них зокрема належать: Windows Open System (WOSA), Win32, загальне відкрите програмне середовище UNIX COSE App Ware Foundation тощо;

- розподілені обчислення. Це один із видів роботи в клієнт-серверній архітектурі, коли дані чи запити, які надходять з робочих станцій, розподіляються між кількома серверами, що забезпечує можливість багатозадачної роботи та оптимізацію використання обчислювальних ресурсів.

Забезпечення розподіленої роботи і віддаленого доступу до документів - це обов'язкова вимога до сучасних інформаційних систем корпоративного рівня. Невід'ємною складовою частиною цієї вимоги є підтримка роботи в мережевій архітектурі.

КІС - це цілісний апаратно-програмний комплекс, що дозволяє задовольнити як оперативні, так і стратегічні потреби підприємства в опрацюванні даних.

Цілісність КІС забезпечується чотирма чинниками:

- концептуальна узгодженість бізнес-процесів, для автоматизації яких створюється ІС, що зберігається впродовж усього життєвого циклу;

- технологічна цілісність, яка проявляється в застосуванні погодженого набору інформаційних технологій для управління інформаційними ресурсами;

- відповідність функціональності робочих місць співробітників їхнім посадовим обов'язкам;

- єдиний регламент обслуговування та експлуатації всіх компонентів ІС, який розробляється при її створенні.

Поєднання цих властивостей принципово відрізняє КІС від суми компонентів з тим же набором функцій і дозволяє справлятися з комплексом проблем, які складно вирішувати у разі безсистемної інформатизації бізнесу.

Відповідно – сучасні КІС використовуються для автоматизації різних видів господарського обліку та управління підприємством. Їх можна умовно поділити на три класи: локальні системи, середні інтегровані системи, великі інтегровані системи.

## 12.2. Основні компоненти корпоративної інформаційної системи

Передумови появи розподілених ІС та проблеми проектування розподілених БД. Однією з найбільш характерних рис інформаційної системи кооперації є те, що вона є розподіленою, реалізована в гетерогенному мережному середовищі, програмне забезпечення і СУБД неоднорідні. При цьому в корпорації має підтримуватися єдиний інформаційний простір [36].

Появлення розподілених інформаційних систем (РІС) було зумовлене тим, що структура великих підприємств логічно та фізично розділена. Тобто, в такій системі розподілені усі ресурси. У кожному організаційному рівні виконується робота з певними наборами даних. Схематично поділ даних між серверами БД підрозділів може бути подано у такий спосіб (рис. 12.1). У цьому випадку єдина БД підприємства розділена між серверами БД. У загальному випадку розподілена або дистрибутивна БД (DDB Distributed DataBase) - це сукупність множини взаємопов'язаних баз даних, що розподілені в комп'ютерній мережі.

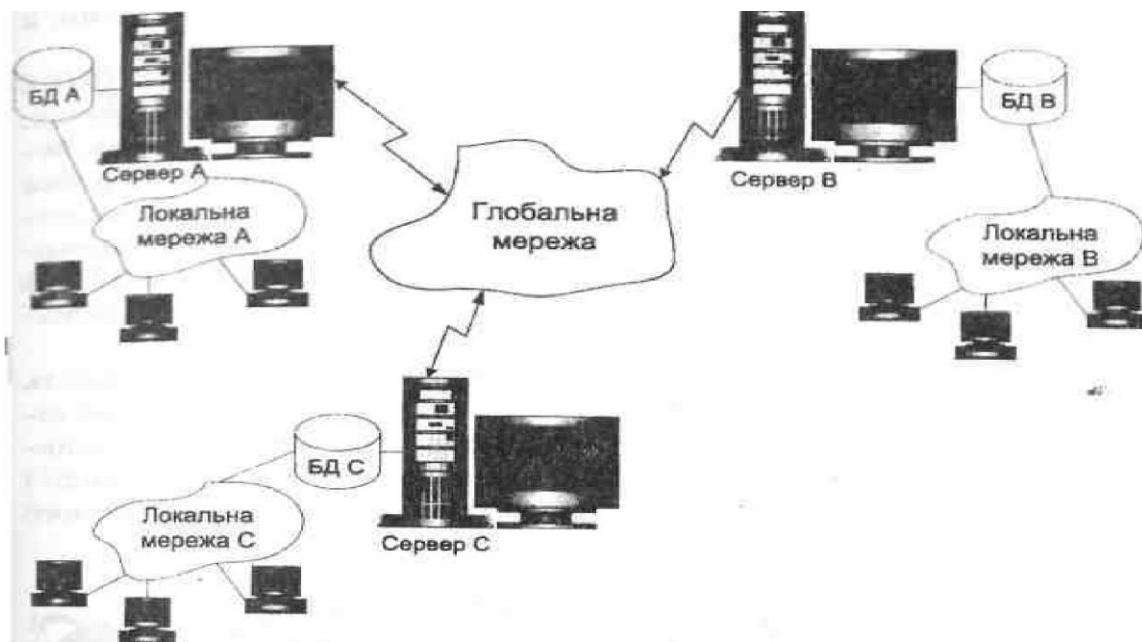


Рис. 12.1. Схема розподілу даних між серверами

При цьому інформаційна система, як правило, є такою, що реалізована на базі різномірних технічних та програмних засобів, або неоднорідною, або гетерогенною.

Клієнт такої системи повинен мати можливість одержувати доступ до даних, які знаходяться на будь-якому сервері так, ніби всі дані знаходяться на його локальному ПК. Така БД корпорації може бути подана або як система баз даних з віддаленим доступом на основі розширеної архітектури „клієнт-сервер”, або як розподілена база даних, що відповідає фундаментальному принципу РБД.

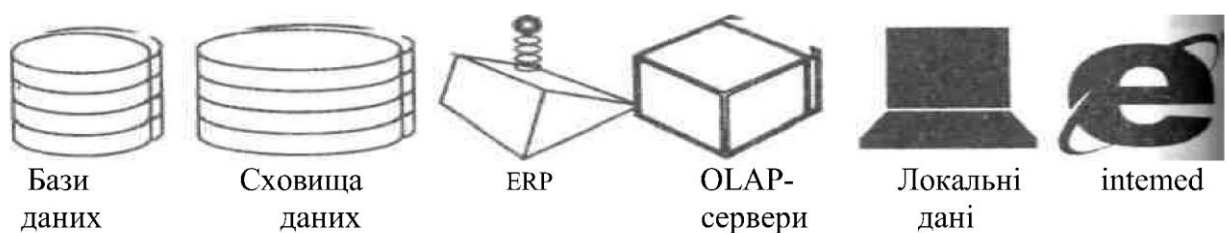
У випадку системи баз даних з віддаленим доступом дані зберігаються на серверах, а прикладні програми виконуються на станціях клієнтів, причому місця їх з'єднання не сховані від користувача.

У випадку дистрибутивної БД має виконуватися фундаментальний принцип РБД «Rule Zero», або «Правило нуль»: «Для користувача розподілена система має виглядати такою ж, як і нерозподілена». Тобто, усі проблеми розподілу мають бути внутрішніми, а не зовнішніми (не для користувачів).

Дж. Дейт сформулював дванадцять цілей забезпечення «RUIE ZERO»: локальна автономія, незалежність від центрального вузла безперервне функціонування, незалежність від розташування, ні залежність від фрагментації, незалежність від реплікації, обробка розподілених запитів, керування розподіленими транзакціями, ні залежність від апаратного забезпечення, незалежність від операційної системи, незалежність від мережі, незалежність від СУБД [37]. Кожна з них є ідеальною, але сучасні розробки баз повинні наближатися до їх виконання.

Під єдиним інформаційним простором розуміється сукупність методичних, організаційних, програмних, технічних і телекомунікаційних засобів, що забезпечують оперативний доступ до будь-яких інформаційних ресурсів підприємства в межах компетенції прав доступу фахівців.

Рис. 12.2 є ілюстрацією організації єдине інформаційного простору корпорації.



**Рис. 12.2. Єдине рішення для всієї корпоративної інформації**

Таблиця 12.1 – Бази даних

<b>Oracle</b>	Реляційні	Спеціалізовані рішення відповідно до стандарту ERP	Oracle Express Server	Text Excel	HTML XML ...
<b>SQL Server</b>	Багатовимірні		SQL Server		
<b>Teradata</b>	Паралельні				
<b>Sybase</b>					
<b>DB2</b>					

Рис. 12.3 ілюструє архітектуру програмного забезпечення бізнес об'єктів корпорації.



Рис. 12.3. Архітектура програмного забезпечення бізнес об'єктів корпорації

Представлена архітектура призначена для забезпечення ефективного доступу до даних, створення звітності й аналізу інформації, підтримки прийняття рішень вищим ешеленом адміністрації підприємства. Однією з найважливіших проблем організації даних у такій системі є максимальне задоволення вимог користувачів та здійснення ефективного контролю і керування самою ІС. А для користувачів повинні бути забезпечені:

- автономність роботи з даними;
- простота і зручність інтерфейсу. Контроль роботи самої системи вимагає:
- могутньої системи безпеки;
- масштабованої архітектури;
- розширюваності, відкритості компонентів.

### 12.3. Особливості архітектури та концепції організації даних в аналітичній частині корпоративної інформаційної системи

База даних корпорації поділяється на дві частини – *оперативну* (деталізовану, операційну або *транзакційну*) та *аналітичну*.

Традиційні системи обробки даних (СОД) з деталізованими даними, яким властива OLTP (оперативна транзакційна обробка) призначені для виконання першого етапу автоматизації в організації. Цей етап зазвичай полягає в наведенні порядку в процесі рутинної обробки даних. Вони використовуються на нижньому операційному рівні управління. СОД не призначені для тривалі збереження даних; у міру старіння дані вивантажуються в архіви, вилучаються з транзакційної БД.

СППР є вторинними відносно операційних систем та визначають стратегічний рівень управління. Вони можуть містити ситуаційні центри, засоби багатомірного аналізу даних та інші OLAP-інструменти аналітичної обробки даних. Використовуваний на цьому рівні спеціальні ММ (математичні методи) дозволяють прогнозувати динаміку різних показників, аналізувати витрати за різними видами діяльності, усвідомлювати їх детальну структуру, формувати та будувати аналітику по різних схемах.

Головною вимогою до аналітичної системи чи СППР є забезпечення аналітиків/експертів ефективним інструментом для проведення оперативного аналізу даних, які надійшли з багатьох джерел та накопичені за достатньо тривалий період (дані характеризують об'єкт керування в історичній ретроспективі) за довільною комбінацією критеріїв.

Для повноцінного функціонування аналітичної системи необхідні не тільки внутрішньо корпоративні дані, а й дані з різних електронних статистичних збірників (загальнодоступні та комерційні), прогнозів розвитку регіонів і сфер економіки, законодавчої бази.

Одним із головних моментів при розробці СППР є організації бази даних.

Для виконання аналізу даних повинні бути розроблені сучасні варіанти організації єдиного інформаційного простору корпорації, що базується на розділених базах даних (РБД) різних СОД. Кожний запит до такої РБД динамічно транслюється в запити до розподілених компонентів, отримані результати блискавично узгоджуються, пов'язуються, агрегуються і повертаються користувачу. Однак такий спосіб має ряд істотних недоліків:

- час обробки запитів до розподіленої БД часто є неприйнятні для аналітичних

- систем, тому що структура бази даних СОД, розрахованої на інтенсивне відновлення, у високому ступені нормалізована, і в аналітичному запиті до неї потрібне з'єднання значної кількості таблиць;

- інтегрований погляд на розподілене корпоративне схови не можливий тільки

– при виконанні вимоги постійного зв'язку всіх джерел даних у мережі. При цьому тимчасова недосяжність хоча б одного із джерел може зробити роботу інформаційно-аналітичної системи неможливою або призвести до невірних результатів;

– головним недоліком є відсутність практичної можливості огляду тривалих історичних послідовностей.

Відомі такі концепції побудови аналітичних систем, принципово відмінні від варіанта РБД:

Information Warehouse - сховища даних (СД);

Data Mart - вітрини даних (ВД), або кіоски даних, для яких властива OLAP- обробка даних;

Relational On-Line Analytical Processing (ROLAP) - оперативна аналітична обробка даних реляційної моделі;

Multidimensional On-Line Analytical Processing (MOLAP) - оперативна аналітична обробка даних багатовимірної моделі;

Hybrid On-Line Analytical processing (HOLAP) - оперативна аналітична обробка даних гібридної моделі.

Концепція формування та використання БД передбачає не просто єдиний логічний погляд на дані організації, а дійсну реалізацію єдиного інтегрованого джерела даних. Автором концепції БД є Б. Інмон, який визначив БД, як «предметно-орієнтовані, інтегровані, немінливі, підтримуючі хронологію набори даних, організовані для цілей підтримки керування», покликані виступати в ролі «єдиного джерела істини», що забезпечує менеджерів і аналітиків достовірною інформацією, необхідною для оперативного аналізу та прийняття рішень.

У табл. 22.1 наведено порівняння характеристик даних в інформаційних системах, орієнтованих на операційну та аналітичну обробку.

Сховище даних функціонує за таким сценарієм. Згідно з заданим регламентом до нього вносять дані про об'єкт керування з різних джерел - баз даних систем оперативної обробки. Дані приводяться до єдиного формату, узгоджуються і (в ряді випадків) агрегуються мінімально необхідного рівня узагальнення. Звичайно у сховищі підтримується хронологія даних.

Полегшеним варіантом корпоративних БД є дані (Data Mart). Концепція цих видів БД була запропонована Forrester Research у 1991 р. При цьому головна ідея полягала в тому, що дані містять тематичні підмножини заздалегідь агрегованих даних, за розмірами набагато менші, ніж загальнокорпоративне сховище, та вимагають менш потужної техніки.

Таблиця 12.1 – Порівняльна характеристика даних в операційних та аналітичних системах

<b>Характеристика</b>	<b>Операційні системи</b>	<b>Аналітичні системи</b>
Частота оновлення	Висока	Мала
Джерела даних	В основному внутрішні	В основному зовнішні
Об'єми даних, які зберігаються	Сотні мегабайт, гігабайти	Гігабайти і терабайти
Вік даних	Поточні (за період від кількох місяців до року)	Поточні і історичні (за період від кількох років, десятки років)
Призначення	Фіксація, оперативний пошук та перетворення даних	Зберігання деталізованих та агрегованих історичних даних, аналітична обробка, прогнозування і моделювання

У табл. 12.2 наведені основні вимоги до даних у сховищі даних  
Таблиця 12.2 – Основні вимоги до даних у сховищі даних

<b>Вимога</b>	<b>Засоби забезпечення</b>
Предметна орієнтованість	Всі дані про деякий предмет (бізнес-об'єкти збираються (з безлічі різних джерел), очищаються, узгоджуються, доповнюються агрегуються і представляються в єдиній формі, яка є зручною для бізнес-аналізу
Інтегрованість	Всі дані про різні бізнес-об'єкти взаємно узгоджені і зберігаються в єдиному загальнокорпоративному сховищі
Незмінність	Початкові (історичні) дані після того, як вони були узгоджені, верифіковані та внесені в загальнокорпоративні сховища, залишаються незмінними і використовуються виключно в режимі читання
Підтримка хронології	Дані хронологічно структуровані і відображують історію за достатній для виконання задач бізнес-аналізу і прогнозування період часу

За формулюванням авторів вітрини даних - це множина тематичних баз даних, які містять інформацію, що відноситься до окремих аспектів діяльності організації.

Переваги концепції вітрин даних:

- аналітики бачать тільки ті дані, які їм реально потрібні;

- цільова вітрина даних максимально наближена до кінцевого користувача;
- вітрини даних містять безліч заздалегідь агрегованих даних, які простіше проєктувати і настроювати;
- вітрини даних не потребують потужної обчислювальної техніки

Саме вітрини даних або щось дуже близьке до них мав на увазі Е. Кодд, коли використав термін OLAP-Server.

Головним недоліком концепції є те, що територіальна розподіленість не дозволяє контролювати надмірність та, відповідно, цілісність БД.

Ідея поєднання СД та ВД належить М. Демаресту (M. Demarest), у 1994 році запропонував об'єднати дві концепції та використати СД як єдине інтегроване джерело для числених вітрин даних.

У наслідок цього з'явилася багаторівнева структура аналітичної системи яка представлена на рисунку 12.3. Таке багаторівневе рішення поступово стало стандартом, дозволяючи найбільш повно реалізувати та використовувати переваги кожного з підходів - компактного зберігання деталізованих даних при підтримці великої БД, що забезпечуються реляційними СУБД, простоту настроювання при роботі з агрегованими даними, які забезпечуються багатовимірними СУБД.

Сучасні аналітичні системи не є системами штучного інтелекту, вони не можуть ні допомогти, ні перешкодити в прийнятті рішення їх метою є своєчасно забезпечити користувача всією інформацією необхідною для прийняття відповідних рішень.

Кінцеві користувачі СППР за характером діяльності можуть бути об'єднані у три основні категорії:

- Аналітики, які повинні володіти не тільки методами дослідження предметної сфери, але й мати уявлення про СД, а також володіти інструментами автоматизованої розробки додатків.
- Середня ланка керівних працівників, які використовують дані СД для підготовки рішень на рівні свого підрозділу. Ця категорія рідко використовує деталізовані дані, зосереджуючись на слабо, та дуже агрегованих даних. Інструментами їх роботи є стандартні звіти, настроєні на інтерактивний режим роботи зі спеціалізованими додатками.
- Вищий ешелон керівництва використовує надто агреговані дані з основних показників, спеціалізовані додатки у вигляді інтерактивних звітів, які відображують діяльність організації загалом для прийняття стратегічних рішень.

Розглянута концепція сховищ даних орієнтована винятково на збереження, а не на обробку корпоративних даних. Вона не визначає архітектуру цільових аналітичних систем, а тільки створює поле діяльності для їх функціонування, концентруючись на вимогах до даних.

Таким чином, вона залишає можливості вибору в усьому, що стосується:

- способів представлення даних у цільовому сховищі (наприклад, реляційний або багатовимірний);

- режимів аналізу даних (статичний чи динамічний). За критерієм режиму аналізу даних інформаційно-аналітичні системи підрозділяються на дві категорії - статичні та динамічні.

Статичні СППР, або інформаційні системи (executive Information Systems (EIS)) - обмежені за функціональністю які містять у собі визначені заздалегідь запити, та не здатні відповісти на всі питання до наявних даних, які можуть виникнути при прийнятті рішень [38]. Результатом роботи такої системи, як правило, є багатосторінкові звіти, після ретельного вивчення яких аналітика з'являється нова серія питань. Кожний новий, наперед передбачений при проектуванні такої системи запит, має бути спочатку формально описаний, переданий програмісту, закодований і тільки потім виконаний. Термін очікування в такому випадку може складати години і дні, що не завжди прийнятною. Таким чином, зовні простота статичних СППР, за яку активно бореться більшість мовників інформаційно-аналітичних систем, обертається катастрофічною втратою гнучкості.

Динамічні СППР орієнтовані на обробку нерегламентованих несподіваних (ad hoc) запитів аналітиків до даних та формуванні звітів довільної форми. Найбільш глибоко вимоги до таких систем розглянув Е. Кодд, поклавши початок концепції OLAP. Робота аналітиків з цими системами полягає в інтерактивній послідовно формування запитів і вивчення їх результатів, кожний з яких може викликати потребу в новій серії запитів. У табл. 12.3 наведена порівняльна характеристика запитів у статичній і динамічній аналітичній системах.

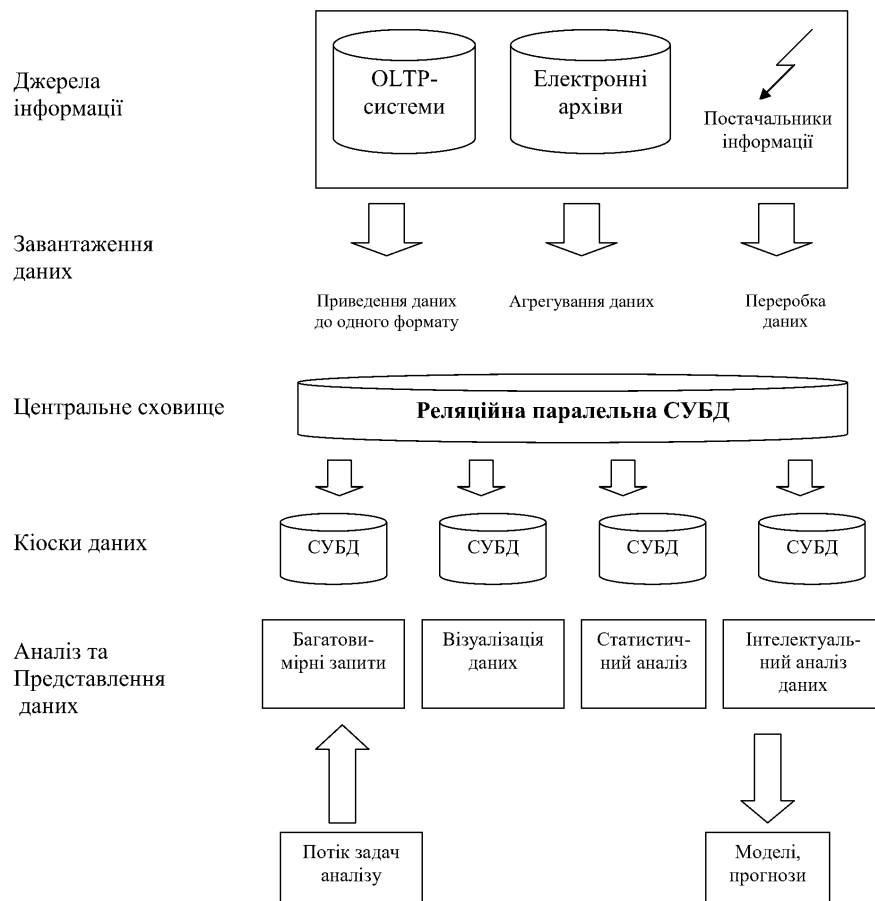


Рис. 12.4. Багаторівнева структура аналітичної частини інформаційної системи

Таблиця 12.3 – Порівняльна характеристика запитів при статичному та динамічному аналізі даних

Характеристика	Статистичний аналіз	Динамічний аналіз
Типи питань	Скільки? Як? Коли?	Чому? Що буде якщо?
Час відгуку	Не регламентувати	Секунди
Типові операції	Регламентований звіт, діаграма	Послідовність інтерактивних звітів, діаграм, екранних форм
Рівень аналітичних вимог	Середній	Високий
Тип екранних форм	В основному визначений заздалегідь, регламентований	Обумовлений користувачем
Рівень агрегації даних	Деталізовані і агреговані	В основному агреговані
Вік даних	Історичні, поточні і прогнозовані	Історичні, поточні і прогнозовані
Типи запитів	В основному передбачувані	Непередбачені від випадку до випадку
Призначення	Регламентована аналітична обробка	Багато прохідний аналіз, моделювання та побудова прогнозів

## 12.4 Система управління ресурсами підприємства Oracle Application

Будучи світовим лідером в області автоматизації підприємств і використовуючи свій більш ніж двадцятирічний досвід з розробки комплексного програмного забезпечення, Oracle пропонує компаніям і організаціям всіх сфер діяльності програмні рішення - сімейство модулів Oracle Applications, призначене для створення корпоративних інформаційних систем. Пакет бізнесу-додатків Oracle Applications - це велика кількість інтегрованих програмних модулів, кожен з яких представляє повністю функціональні рішення в області управління кадрами, фінансами, виробництвом, матеріально-технічним постачанням і збутом. У сукупності модулі додатків Oracle утворюють могутню систему ділової активності, здатну задовольнити всі вимоги сучасного бізнесу і вирішити практично будь-які задачі, що з ними може стикнутися сучасне підприємство.

Перша версія корпоративних бізнесу-додатків Oracle Applications з'явилася на ринку в 1989 р. Вона була результатом більш ніж 4-х річної роботи розробників Oracle по узагальненню досвіду створення «замовних» рішень. У створення першої версії було залучено біля 200 розробників, і капітальні витрати на її випуск становили \$165 млн. У наступному році першими клієнтами Oracle Applications стали біля 90 компаній в західній Європі і США.

Починаючи з 1990 року корпорація Oracle проводить планомірну політику по розширенню частки готових рішень сімейства Oracle Applications у загальній лінійці продуктів компанії. Власна технологічна база і передові технічні рішення Oracle роблять пакет готових додатків Oracle Applications неперевершеною за багатьма техніко- експлуатаційними характеристиками. Багата функціональність і масштабність прикладних програм забезпечують провідні позиції Oracle на ринку корпоративних ERP-систем.

Остання версія R11 Oracle Applications підтримує 29 мов, і на сьогоднішній день клієнтами корпоративних додатків Oracle є більше за 7700 організацій і компаній в 79 країнах світу. Версія R11 повністю реалізована в архітектурі Інтернет/Інтранет (ICA, Internet Computer Architecture) і за багатьма параметрами не має аналогів на ринку корпоративних систем. Трудовитрати Oracle на її створення становили 700 людино-років. Закладена в Oracle Applications бібліотека бізнес- моделей узагальнює світовий досвід управління в різних галузях промисловості. Реструктурування виробничих процесів підприємства з використанням Oracle Applications - це прибутковий і швидкий шлях до побудови продуктивної системи управління компанією будь-якої сфери діяльності. Модульний підхід при впровадженні Oracle Applications дозволяє замовнику почати з мінімального набору модулів і поступово розширювати його, доповнюючи базову функціональність системи і позбавляючи від необхідності придбання зайвих у даний момент функціональних блоків.

Існують такі групи модулів.

Модулі Oracle Applications для управління фінансами: (Oracle Financials): Головна книга (General Ledger),

Кредитори (Accounts Payable), Дебітори (Accounts Receivable), Рух грошових засобів (Cash Management), Основні засоби (Fixed Assets), Фінансовий аналізатор (Financial Analyzer).

Модулі управління матеріальними потоками: Планування матеріальних потоків (Supply Chain Management), Управління матеріальними запасами (Inventory Management), Планування постачальників (Supplier Shediling), Закупівля (Purchasing), Введення замовлень (Order Entry), Конфігуратор продукції (Configuration Management), Послуги (Service Management), Контроль якості (Quality Management).

Модулі Oracle Applications для управління виробництвом: Технологічне проєктування (Engineering), Конфігуратор продукції (Configuration Management), Специфікації (Bill of Materials), Планування матеріального постачання (Supply Chain Planning), Планування виробництва (Master Production Scheduling), Планування виробничих потужностей (Capacity Planning), Управління матеріальними запасами (Inventory Management), Планування постачальників (Suppliers Scheduling), Закупівля (Purchasing), Незавершене виробництво (Work in Process), Управління витратами (Cost Management), Контроль якості (Quality Management), Управління безперервним виробництвом (Oracle Process Manufacturing).

Модулі Oracle Applications по управлінню проєктами: Облік витрат по проєктах (Project Costing), Виставлення рахунків по проєктах (Project Billing), Виробництво по проєктах (Project Manufacturing), Облік персональних витрат і часу (Personal and Expenses).

Представимо основні функціональні можливості Oracle Applications.

Посилення зв'язків між підприємством і бізнес- партнерами. За допомогою Oracle Applications компанія може добитися значних поліпшень в роботі з постачальниками і замовниками. Суть цих поліпшень у розширенні ринків збуту за рахунок залучення нової клієнтури і поліпшення рівня обслуговування вже існуючих клієнтів. Замовники можуть працювати з Інтернетом-додатками Oracle так само, як і співробітники компанії. Вони самостійно можуть відвідувати Інтернет-сервер підприємства для отримання свіжої інформації про ціни на продукти, що поставляються, і послуги, про доступність номенклатурних позицій на складі готової продукції, для розміщення замовлень на постачання і відстеження етапів їх проходження. Клієнти вибирають продукт, який вони хочуть, вирішують, чи влаштовує їх поточна ціна і відразу оформляють свої замовлення на комп'ютері. Замовлення напряму передається на склад або комерційний відділ без яких-небудь ручних операцій. Клієнт отримує повний звіт, що містить статус замовлення, який оновлюється в реальному часі в міру того, як товар переміщується зі складу на відвантаження.

Використовуючи Інтернет-додатки Oracle Applications, постачальники компанії можуть отримувати інформацію про потребу підрозділів підприємства в матеріалах і послугах, і відповідно коректувати власні плани постачання. Наприклад, вони можуть мати доступ до інформації про виробничі запаси компанії. Коли кількість на складі якихось матеріалів вичерпується, постачальники можуть за власною схемою організувати їх відвантаження і доставку. Цей процес дозволяє зробити постачальника реальним членом виробничої групи, скоротити дистанцію між компаніями і знизити витрати обох підприємств. Розподіл з допомогою Oracle Applications бізнес-процесів в Інтернет, за межі окремого підприємства, перетворює електронну торгівлю з інструмента маркетингу в новий ефективний зразок ведення бізнесу. Додатки CRM (Customer Relationships Management) Oracle Applications є комплексним рішенням задач побудови пов'язаних бізнес-процесів всередині компанії, орієнтованих на роботу з клієнтами (продаж, маркетинг, післяпродажне обслуговування).

Система інформаційної підтримки керівництва. Oracle Business Intelligence System (OBIS, Система інформаційної підтримки керівництва) являє собою набір з 20 Інтернет-прикладних програм, призначених для відстежування і оперативного відображення для керівництва найбільш значущих, з точки зору управління, показників діяльності підприємства. Система виконує роль аналітичної надбудови над базовою системою управління ресурсами підприємства ERP (Enterprise Resource Planning). За її допомогою керівники підрозділів і відділів можуть отримувати оперативну звітність про функціонування своїх підрозділів. OBIS автоматично виконує семафорні функції, відстежуючи значення ключових показників (наприклад, поточну оборотність, дебіторську заборгованість, рівень незнижуваних складських залишків і т. д.).

Фінансовий аналіз і планування. Фінансовий аналіз вимагає надання інформації не тільки керівництву підприємства, але і більш широкому колу співробітників, яким щодня треба приймати обґрунтовані рішення на основі оперативних, несуперечливих даних. Фінансовий аналізатор Oracle дозволяє всім уповноваженим особам організації провести пошук потрібної інформації, деталізувати дані до потрібної міри і виводити негайно результати, не чекаючи втручання співробітників відділу автоматизації або бухгалтерії. Аналіз може бути будь-яким складним, оскільки засоби інтерактивних запитів Oracle підтримують необмежену кількість аналітичних зрізів, що набагато перевершує можливості звичайних засобів звітності. Модулі Головної книги і Фінансового аналізатора дозволяють вести фінансове планування по схемах зверху вниз, знизу вгору і розподіленим методом, як на корпоративному рівні, так і на рівні окремих підрозділів.

Фінансова консолідація. Oracle Applications володіє великою гнучкістю і легко підтримує часті зміни, що нерідко відбуваються в сучасних організаціях. Зміни організаційної структури підприємства відбиваються в системі за допомогою Редактора організаційної структури. Вбудована

система глобальної консолідації (GCS, Global Consolidation System) дозволяє задавати необмежену кількість правил консолідації результатів господарської діяльності і оперативних балансів підприємств на рівні окремих господарських операцій або залишків по рахунках. За будь-якої структури організації допускається автоматична консолідація окремих господарських одиниць незалежно від відмінностей у операційних валютах, календарях і планах рахунків.

Управління платіжами. Сучасні умови жорсткої конкуренції вимагають високої ефективності операцій в поєднанні з потужним інструментом управління. Тісна інтеграція модулів Закупівлі, Кредитори і Основні засоби усуває необхідність роботи з паперовими документами і надмірного введення даних, оскільки вся необхідна інформація автоматично може перенаправлятися всередині системи. Для підвищення ефективності операцій застосовуються засоби автоматичного створення замовлень на придбання, утримання податків і обробки рахунків-фактур та платежів. Потужність управління досягається за рахунок багаторівневого блокування платежів за інвойсами постачальників і узгодження відфактурованих поставань за двома, трьома або чотирма параметрами контролю.

Виставлення рахунків і збір засобів. Модуль Дебітори забезпечує весь необхідний набір засобів для ведення господарських операцій, включаючи підтримку необмежених способів платежів, облік місцевих і федеральних податків, а також видачу акредитивів. Система Oracle Applications надає можливість істотної автоматизації процесів виставлення рахунків і збору дебіторської заборгованості. Наприклад, допускається автоматична обробка надходжень з використанням електронних переказів, векселів та списання з рахунку. Модуль Дебіторів дозволяє вводити оплату за відвантажені товари або надані послуги без прив'язки до документів відвантаження і товарів. Автоматично реалізована можливість розподілу оплати, що поступила по кожному товару або товарній групі пропорційно заборгованості клієнта або за іншими алгоритмами, що задаються користувачем. Модуль Дебіторів підтримує паралельне ведення книги продажу по оплаті і по відвантаженню, для цілей оподаткування і для цілей бухгалтерського відповідного обліку. У разі мультивалютного обліку відповідні сумарні і курсові різниці розраховуються автоматично. Гнучкі облікові схеми і необмежена глибина аналітичного обліку господарських операцій з реалізації продукції і послуг дозволяють істотно спростити процедуру узгодження даних книги продажу і податкових декларацій по розрахунку з бюджетів по ПДВ. Гнучка структура рахунків Головної книги забезпечує необхідну глибину аналітичного обліку, що дає можливість формувати інформацію в розрізі, необхідному для достовірного визначення податкової бази і складання звітності по податках і інших обов'язкових фіскальних платежах.

Рух грошових засобів. Інтеграція модулів Руху грошових коштів, Дебітори, Кредитори і Головна книга забезпечують повне рішення для

мультивалютного вивіряння банківських виписок і управління рухом грошових коштів на рахунках організації.

Управління основними засобами. Модуль Основні засоби призначений для організації точного обліку майна, капітальних вкладень і нематеріальних активів підприємства. Його використання дозволяє підприємству забезпечити оптимальний вибір стратегії обліку і оподаткування основних засобів з можливістю паралельного обліку основних засобів в декількох стандартах обліку (наприклад, українських або міжнародних стандартах). Паралельне ведення основних засобів за двома стандартами виключає необхідність дублювання інформації за картками аналітичного обліку. Усі бухгалтерські проведення при операціях з активами (оприбуткування, рух, зміна вартості, переоцінка, вибуття, переміщення і перепідпорядкування) генеруються автоматично.

Управління постачанням і збутом. Модулі управління постачанням і збутом включають багатофункціональні інтегровані інструменти планування і виконання, що оптимізують управління попитом і пропозицією. Канали збуту розміщують замовлення через центри розподілу, які, у свою чергу, можуть задовольнити попит з використанням різних варіантів постачання. Додаткова автоматизація постачання і збуту досягається за рахунок повної підтримки електронного обміну даними з постачальниками і замовниками в стандарті EDI.

Планування матеріальних потоків. Модуль Планування матеріальних потоків Oracle реалізує сучасний підхід до інтеграції планування виробництва і збуту. За допомогою списків розподілу і правил вибору джерела одночасно виконується планування всієї мережі матеріальних потоків, після чого замовлення на виробництво, поповнення і придбання виписуються автоматично. Модуль Закупівлі дозволяє покупцям переглядати каталоги постачальників, передавати прогнози і замовлення на придбання, а також заздалегідь отримувати повідомлення про постачання в електронному вигляді. Крім того, за допомогою Web-прикладних програм Oracle, постачальники отримують можливість переглядати свої власні прогнози, договори, рахунки-фактури і платежі. Модуль Закупівлі Oracle Applications підтримує різні типи замовлень, включаючи постійні контракти. Він дозволяє відстежувати історії замовлень на купівлю для зіставлення отриманої кількості і кількості за рахунком-фактурою. При цьому система автоматично контролює стани запасів на складах для визначення оптимальної кількості, що замовляється.

Управління запасами. Надійне функціонування матеріальних потоків неможливо без гнучкості операцій і точності обліку запасів. Модуль управління матеріальними запасами дозволяє визначити складські структури і необхідні параметри для декількох місць фізичного зберігання запасів. Є можливість гнучкого контролю по номерах партій, серійних номерах або номерах версій. Облік партіями включає інформацію щодо статусу контролю якості, терміну зберігання партії, дати закінчення терміну зберігання, статусу

партії, а також параметрів партії, які додатково задаються. Точність оперативного обліку складських залишків досягається за рахунок застосування вибіркової періодичної інвентаризації (АВС аналізу) по графіку, що автоматично генерується системою. Модуль управління матеріальними запасами Oracle забезпечує точне ведення складських залишків, розподіл і резервування запасів за місцями зберігання, складах, партіях, підпартіях і статусу контролю якості кількох одиницях вимірювання. Система автоматично формує бухгалтерські проводки і зберігає контрольну інформацію за операціями із запасами (коригування кількості, зміну вартості, руху між матеріально-відповідальними особами, резервування, видача зі складу у виробництво і т. д.). При цьому виключається дублювання даних складського сортового обліку, за картками аналітичного обліку і оборотних відомостях бухгалтерії, оскільки оперативна інформація, що вводиться комірником, стає одночасно доступною і бухгалтеру.

Управління замовленнями на продаж. Модуль введення замовлень забезпечує ефективність і високу якість обслуговування замовників. Кожний канал збуту може визначити власну політику обслуговування для максимального прискорення обробки замовлень. Наприклад, допускається вибір політики ціноутворення, кредитної політики, політики затвердження і доставки залежно від каналу і конкретного замовника. Торгові підрозділи можуть перевіряти наявність товарів, розміщувати замовлення на наявні запаси чи майбутнє постачання або забезпечувати постачання з будь-якої точки світу. Щоб гарантувати точне і своєчасне постачання, Конфігуратор продукції виконує перевірку складних конфігурацій перед розміщенням замовлень на них. Модуль введення замовлень Oracle Applications підтримує весь цикл реалізації готової продукції і послуг: введення і зміну замовлень на продаж з використанням інформації реального часу про запаси на складах; генерацію вимог на виконання замовлень; перевірку кредитної лінії замовника і контроль виконання відвантаження; ведення преїскуранта і множинних знижок на рівні однієї позиції або всього замовлення; автоматичне обчислення планової дати відвантаження на основі необхідної дати замовника і терміну транспортування; ведення даних з упаковки і відвантаження (число упаковок, маса нетто і брутто) і т. д.

Післяпродажне обслуговування. Для управління післяпродажним обслуговуванням замовників призначений модуль Послуги. Він дозволяє вести базу даних продуктів, встановлених у замовників, вести контракти на обслуговування, записувати контракти на обслуговування, записувати заявки на обслуговування, приймати повернення і виконувати ремонт.

Планування і управління виробництвом. Модулі управління виробництвом Oracle дозволяють вирішувати задачі комплексної автоматизації як дискретного, так і безперервного типу виробництва сучасного підприємства. На відміну від інших систем управління виробництвом Oracle дозволяє легко розвивати можливості виробництва від

малосерійного складання прототипів до великих обсягів серійного складання і далі до складного планування виробничих замовлень.

Проектування нових товарів. Скорочення часу випуску нових товарів на ринок неможливе без тісної взаємодії проєктувальників, виробничників, співробітників відділу маркетингу і замовників. Унікальний механізм потоків операцій Oracle дозволяє налагодити зв'язок всередині підприємства і за його межами протягом всього циклу розробки і моделювання. Модуль Технологічного проєктування скорочує час розробки товарів за рахунок використання потужного каталогу позицій. Існує можливість швидкого створення прототипу і гнучкого планування впровадження нових розробок. Окрім того, модулі Управління витратами і Планування виробництва допомагають визначити вартість реалізації технологічних змін і мінімізувати застарілі запаси.

Планування і моделювання. Мінімізація запасів, що зберігаються, досягається за рахунок своєчасного планування, оптимізованого по безлічі виробничих майданчиків і місцям зберігання, моделювання комплексної структури складського господарства організації. Oracle пропонує повний набір засобів планування, що охоплюють весь виробничий цикл від прогнозування до графіків запуску виробництва, замовлень на поповнення і замовлень на придбання матеріальних цінностей. Відповідальні за планування матеріалів можуть автоматично запускати замовлення на придбання і передавати їх в електронному вигляді безпосередньо постачальникам на виробництво. Потужний механізм планування виробництва Oracle Applications дозволяє формувати прогнози на основі історичних даних з урахуванням сезонних змін; здійснювати автоматичне формування плану на основі прогнозу і замовлень на продаж із засобами візуалізації даних, що прогнозуються, і реальних; виконувати формування планів всіх рівнів і різних горизонтів; автоматично забезпечувати процедури перевірки потужностей стосовно підготовленого виробничого плану.

Управління матеріальними потоками. Сучасні умови жорсткої конкуренції вимагають скорочення тривалості виробничого циклу. Модуль Закупівлі спрощує виконання звичайних операцій, скорочує кількість паперових документів і надає засоби електронного зв'язку для проведення щоденних закупівель. Електронні каталоги позицій і списки постачальників дозволяють легко створювати заявки і інші закупівельні документи. Робота постачальників організована по переліках пріоритетних дій із застосуванням затвердження електронних документів і повідомлень про закінчення терміну дії відкладених контрактів. Є можливість електронної пересилки замовлень на придбання і графіків постачання зовнішнім і внутрішнім постачальникам. Виробництво. Як правило, однієї системи управління виробництвом виявляється замало для обробки всіх вимог виробництва, що неминуче приводить до застосування декількох систем. Модуль незавершене виробництво застосовується для управління як для потокового виробництва з високими обсягами випуску, так і при виконанні унікального разового

складання - виробництво під замовлення. Наприклад, є можливість планувати критично важливі виробничі цикли з точністю до хвилин, отримувати вимоги замовлень на збирання безпосередньо з модуля Введення замовлень, автоматично запускати графіки виробництва за допомогою спеціалізованих систем планування. Система точно моделює інтегроване виробниче середовище для контролювання випуску продукції або вироблення, і допомагає контролювати випускання продукції на підприємстві, відстежуючи вироблення на кожній стадії виробничого циклу.

Управління витратами. Розуміння структури вартості товару є важливим чинником роботи компаній. Модуль Управління витратами і Фінансовий аналізатор дозволяють оцінювати, контролювати і аналізувати рентабельність операцій, аналізувати складові статей витрат підприємства для порівняння планових і фактичних показників складної, багатопрофільної організації. Модуль Управління витратами включає не тільки потужні засоби управління витратами за операціями (при цьому число елементів витрат не обмежується), але і апарат моделювання витрат. Калькуляція собівартості може проводитися за середніми і стандартними цінами з заданою необмеженою кількістю типів витрат, стандартних вартостей матеріалів, накладних витрат і ресурсів. Система дозволяє вести декілька стандартів калькуляції собівартості одночасно, для різних підприємств всередині цілісної групи компаній. Бухгалтерські проводки автоматично переносяться з цехової в центральну бухгалтерію, де вони відбиваються по відповідних статтях калькуляції для подальшої обробки і аналізу.

Управління якістю. На вибір даних з численних систем контролю якості звичайно йде дуже багато часу, якого потім не вистачає для аналізу і прийняття рішень. Модуль Контроль якості надає легкий доступ до даних про якість товарів за допомогою спеціального сховища даних на рівні підприємства. Система замкнених циклів модуля Контроль якості дозволяє легко визначати причини зниження якості, виконувати узгоджений аналіз і робити коригуючі впливи. Допускається також автоматизація коригуючих дій, таких як управління обладнанням, повідомлення персоналу і припинення операцій, причому без жодного втручання співробітників.

Управління персоналом. Витрати на персонал звичайно становлять найбільшу частину витрат організації, причому персонал одночасно є і найбільшим активом. Щоб зберегти конкурентоспроможність, управління персоналом має бути так само динамічне, як і сучасний світ з його постійними змінами законодавства, правил господарської діяльності і умов праці. Модуль Управління персоналом Oracle Applications R11 (Oracle Human Resources) дозволяє добитися максимальної віддачі від співробітників за рахунок ефективного набору персоналу, управління кадрами, навчання, оплати праці і планування кар'єри.

На сьогоднішній день в своєму класі продуктів модуль Управління персоналом Oracle Applications є однією з найбільш функціонально повних

систем для організації роботи відділу кадрів сучасного підприємства. Його використання дозволяє вирішувати такі завдання:

Планування організаційних змін, включаючи моделювання структурних і посадових ієрархій організації (ситуаційний аналіз «що якщо»).

Планування структурних підрозділів, опис розряду, посади, позиції, ведення довідників .

Персональний облік співробітників і кандидатів з повним записом їх професійних якостей (для оптимального використання трудових ресурсів), даних для нарахування зарплати, даних про використання робочого часу, послужного списку співробітників, потреби в підвищенні кваліфікації і результатів навчання співробітників.

Аналіз і ведення звітності за типовими державними формами, по нестандартній звітності за допомогою засобів розробки Oracle - оперативний аналіз даних і підтримку прийняття рішень на базі технології OLAP.

Планування організаційних змін. Багато які організації схильні до постійних змін: злиттю, розукрупненню, крупномасштабним реорганізаціям і т. д. Система управління персоналом повинна забезпечувати швидкий відгук на об'єднання різнорідних структур, спеціальностей і посадових ієрархій, а також на зміну організаційної структури компанії. Створення і оновлення організаційних структур, спеціальностей і посад проводиться за допомогою графічного редактора організації. Для прогнозування змін можна створити модель, яка буде переглянена і затверджена перед її прийняттям. Існує можливість використання нестандартних організаційних одиниць, таких як комітети і робочі групи. Для великомасштабних організацій застосовуються засоби масового перенесення. Нарешті, допускається збереження історичних даних про організації і введення в дію нових структур в заздалегідь заданий час.

Управління проектами. У сучасній економіці все більше місце займає сектор послуг, тому рентабельність багатьох організацій визначається за допомогою обліку проектів. Багато які компанії, в тому числі і ті, які раніше не вели обліку витрат, використовують модуль Проекти для визначення рентабельності окремих проектів в рамках сукупного фінансового результату компанії. Сучасна точна інформація про витрати і прибутки дозволяє керівництву уникнути невиправданих затримок і перевитрати фінансових коштів.

Облік проектів. Керівникам проектів необхідно управляти їх ходом, а економістам-аналітикам потрібні зведення для розрахунку рентабельності. Модуль- проекти дозволяє керівникам проектів вести докладний облік витрат, дотримуючись кошторису і графіка робіт у процесі успішного досягнення мети проекту. Для економістів-аналітиків надається можливість перекладу відомостей про проект на мову фінансових операцій, більш зручних для аналізу. Модуль-проекти дозволяє організувати і контролювати операції по виконанню проекту за допомогою необмеженої структури

розбиття робіт, незалежно від використання плану рахунків Головної книги. Модуль-проекти дозволяє обмінюватися інформацією із зовнішніми системами обліку ремонтних і монтажних робіт, а також може бути інтегрований на рівні обміну даними в режимі реального часу з системами мережного планування третіх фірм [39].

Облік витрат. Більшість систем керівництва проектами надає надто жорсткі способи обліку витрат. На відміну від них модуль Обліку витрат по проектах використовує гнучку систему, що дозволяє визначити необхідний рівень контролю витрат. Допускається як пряме введення витрат, так і їх імпорт через пакетні інтерфейси. Існують зручні засоби запису операцій з постачальниками по проектах в модулях Закупівлі і Кредитори. Є можливість збору витрат на обладнання і інших витрат за допомогою існуючих механізмів обліку витрат з подальшим імпортом даних в модуль Обліку витрат по проектах. Крім збору витрат і зобов'язань для кожної операції проекту, надається можливість контролю над тим, які витрати допускаються за проектом. Спільне використання модулів Виробництво за проектами і Основні кошти дозволяє управляти капіталізацією витрат з капітального будівництва і витрат з пусково-налагоджувальних робіт. Це усуває надмірність введення даних і забезпечує точність обліку проектних робіт. Наприклад, вартість активів можна легко скоригувати, в тому числі і після капіталізації.

Облік персональних витрат і часу. Співробітникам, працюючим поза офісом, необхідно вести облік відомостей про проект навіть тоді, коли вони знаходяться на проектному об'єкті. Модуль Обліку персональних витрат і часу призначений для віднесення робочого часу і витрат співробітників за проектами під час їх роботи в офісі і поза ним. Застосування нової технології мобільного доступу дозволяє відрядженим співробітникам вводити дані на персональний комп'ютер в автономному режимі, а потім завантажувати їх на сервер підприємства. Введення даних про час і витрати на переносний комп'ютер забезпечує необхідну гнучкість для контролю витрат за проектом. Щоб запобігти введенню неправильних або недозволених витрат, усі операції за часом і витратами проходять перевірку по заданому набору умов. Залежно від правил діяльності, що застосовуються, подані зведення про час і витрати можуть бути направлені на перегляд і затвердження перед їх обробкою.

Капіталізація витрат за проектом. Для проектів капітального будівництва необхідні засоби легкого управління всіма витратами і витратами незавершеного будівництва за проектом. Тісна інтеграція модуля Обліку витрат по проектах і Основні засоби усуває надмірність введення даних і забезпечує точність обліку. Вартість активів можна легко скоригувати, у тому числі й після капіталізації.

Нарахування прибутків і виставлення рахунків. Перед кожним підприємством, працюючим по проектах, звичайно стоїть задача визначення прибутків і виставлення рахунків в досить гнучкій формі, щоб урахувати різні вимоги замовників. Модуль Виставлення рахунків по проектах

забезпечує гнучке нарахування прибутків відповідно до політики компанії - наприклад, за ставками рахунків за роботу і матеріали, за сумою витрат і націнок, за процентом витрачання або за будь-яким іншим алгоритмом, що визначається користувачем. Зі зміною політики можна легко змінити параметри виставлення рахунків за проєктами, причому без перепрограмування. Допускається застосування різних методів нарахування для різних типів проєктів і підрозділів. Крім того, для кожного проєкту можна виставляти рахунок по-різному, навіть якщо декілька проєктів відносяться до одного клієнта. Для управління подальшим збором коштів застосовується модуль Дебітори.

Інтерактивні запити і звіти. Рентабельність проєкту неможлива без своєчасного доступу до даних, необхідних для обґрунтованого прийняття рішень. Модулі управління проєктами Oracle дозволяють в будь-який час проаналізувати всі дані за проєктом і легко внести необхідні корективи, зберігаючи при цьому повний звіт відстеження операцій. Інструменти деталізування надають докладні відомості про джерело кожної операції. Кожний користувач може визначити власне представлення даних по проєкту за допомогою розробленої Oracle технології досьє. За наявності великого числа стандартних звітів керівникам більше не треба переглядати багато сторінок звітів, а можна зосередитися на проблемній ситуації.

Аналіз проєктів. Як додатковий крок по шляху вирішення проблеми доступу до інформації Oracle надає можливості повного багатомірного аналізу даних за проєктами. Сховище даних Oracle Applications дозволяє всім уповноваженим користувачам провести пошук важливої інформації за проєктом і аналізувати її, не звертаючись за допомогою до співробітників бухгалтерії або відділу автоматизації. Можна порівнювати числові показники за різними проєктами, а також фактичних, договірних, бюджетних і інших показниках по всій організації.

## **12.5. Локальні, середні та великі інтегровані інформаційні системи**

Локальні системи успішно справляються з вирішенням окремих задач обліку на підприємстві, але, як правило, не надають цілісної інформації для автоматизації управління. Перевагою цих систем є порівняно невисока ціна і відносна простота впровадження.

Прикладом таких систем можуть бути "Інфобухгалтер" чи "Турбобухгалтер". Ці системи мають можливості адаптуватись до особливостей підприємства, а деякі з них, наприклад, "Турбобухгалтер" є програмами-конструкторами з широкими адаптаційними можливостями. Вони найефективніші на малих і середніх підприємствах, які не займаються виробництвом продукції.

Щодо старих і нових інформаційних систем цього класу варто враховувати наступне міркування: системи, давно присутні на ринку

вимагають сумісності з попередніми версіями продукту. В багатьох випадках така сумісність не забезпечується. Тому під час вибору системи основними критеріями будуть не лише довіра до фірми-виробника, а й відповідність її особливостям бізнесу конкретного підприємства і перспективи розвитку функціональних можливостей системи.

Великі інтегровані системи - цей функціонально найрозвинутіші, найскладніші і найдорожчі системи. В них реалізуються зазвичай західні стандарти управління рівня MRPII та ERP. Цей вид систем на нашому ринку представлений продуктами фірм SAP, ORACLE, BAAN, People Soft.

До впровадження цих систем на наших підприємствах треба ставитись обережно, оскільки вони дуже дорогі, вимагають докорінної перебудови організаційної структури, системи оперативного управління виробництвом і переходу на жорсткий стандарт управління.

Стандарт управління рівня MRP (Material Requirements Planning) передбачає використання інформаційної системи планування матеріальних ресурсів. В практиці MRP - це програмний продукт, логіка роботи якого спрощено може бути наведена у вигляді схеми (рис. 12.5).

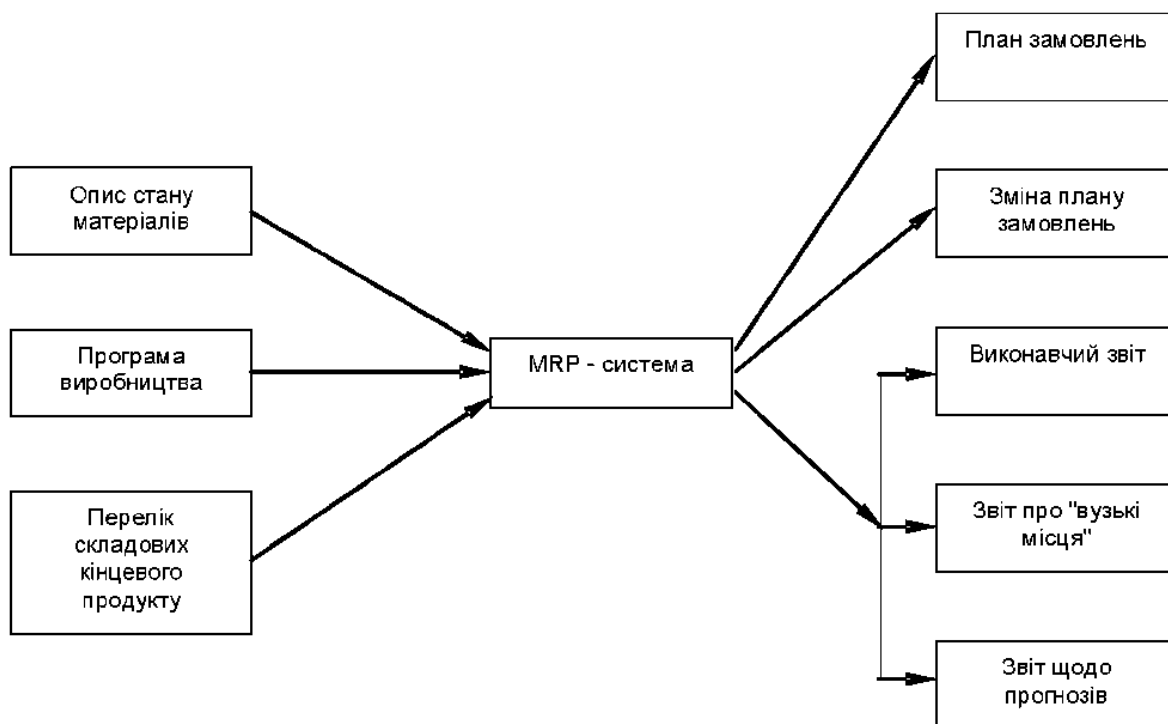


Рис. 12.5 Вхідні дані та результати роботи MRP-системи

Цикл роботи системи складається з наступних основних етапів:

1. Система аналізує прийняту програму виробництва визначає оптимальний графік виробництва на період, який планується.
2. Матеріали, не включені до виробничої програми, але вказані в поточних замовленнях, включаються в планування як окремий пункт.

3. На основі затвердженої програми виробництва і замовлень на комплектуючі, що не входять до неї, для кожного окремого матеріалу відповідно до переліку складових кінцевого продукту обчислюється повна потреба.

4. На основі повної потреби, враховуючи поточний статус матеріалу, для кожного періоду часу і для кожного матеріалу розраховується чиста потреба за виразом:

$$\text{ЧП} = \text{ПП} - \text{ІР} - \text{СЗ} - \text{Р},$$

де: ЧП - чиста потреба;

ПП - повна потреба;

ІР - інвентаризовано на руках;

СЗ - страховий запас;

Р - резервування для інших цілей.

Якщо чиста потреба в матеріалі більша нуля, то система автоматично генерує замовлення на нього.

5. Всі замовлення, створені раніше поточного періоду планування, розглядаються, і в разі необхідності в них вносяться зміни для уникнення передчасного постачання чи затримки постачання.

Подальшим розвитком цього підходу стали системи MRP II (Manufacturing Resource Planning - планування виробничих ресурсів).

Система MRP II включає 16 груп функцій:

1. Планування продажу і виробництва.
2. Управління попитом.
3. Складання плану виробництва.
4. Планування матеріальних потреб.
5. Специфікація продуктів.
6. Управління складами.
7. Планові поставки.
8. Управління на рівні виробничого підрозділу.
9. Планування потреб у потужності.
10. Контроль входу/виходу.
11. Матеріально-технічне постачання.
12. Планування розподілу ресурсів.
13. Планування та управління інструментальними засобами.
14. Управління фінансами.
15. Моделювання.
16. Оцінка результатів діяльності.

Системи MRP II не позбавлені, однак і певних недоліків, серед яких:

- орієнтація системи управління лише на чинні замовлення;
- слабка інтеграція з системами проектування продукції;
- слабка інтеграція з системами проектування технологічних процесів;

– недостатня насиченість системи управління функціями управління витратами;

– відсутність інтеграції з процесами управління кадрами.

Необхідність усунення перелічених недоліків спонукала трансформувати системи MRPII в системи нового класу ERP (Enterprise Resource Planning - система планування ресурсів). Системи цього класу більше орієнтовані на роботу з фінансовою інформацією для розв'язання задач управління у великих корпораціях з територіально розпорощеними ресурсами. Сюди включається все, що необхідно для отримання ресурсів, виготовлення продукції, її транспортування і розрахунків із замовниками. Крім цього в системах ERP реалізовані й нові підходи до застосування графіки, використання реляційних баз даних, CASE-технологій для їхнього розвитку, архітектура обчислювальних систем типу "клієнт-сервер" і реалізації їх як відкритих систем.

Системи ERP мають додатково наступні модулі:

- прогнозування;
- управління проектами і програмами;
- введення інформації про склад продукції;
- введення інформації про технологічні маршрути;
- управління витратами;
- управління фінансами;
- управління кадрами.

Таким чином, ERP є покращеною модифікацією MRPII. Її мета - інтегрувати управління всіма ресурсами підприємства, а не лише матеріальними, як це було в MRPII.

Корпоративна інформаційна система є не просто інструментом для автоматизації процесів, але й важливим засобом для підтримки стратегічного управління підприємством. Вона дозволяє інтегрувати різні підсистеми компанії, забезпечує ефективне прийняття рішень на всіх рівнях і сприяє досягненню стратегічних цілей організації. Впровадження КІС є важливим етапом у розвитку підприємства, що вимагає ретельного планування та уваги до деталей на всіх етапах від вибору системи до її експлуатації.

## **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Дайте визначення корпоративної інформаційної системи (КІС). Які основні бізнес-функції вона повинна підтримувати?

2. Назвіть і опишіть основні компоненти КІС (ERP, CRM, BI, Інтеграційна шина, Data Warehouse) і їх ролі в аналітичній частині системи.

3. Які особливості архітектури даних в аналітичній підсистемі КІС (шари зберігання, ETL, семантичні моделі)? Наведіть приклад структури Data Warehouse для виробничого підприємства.

4. Коротко опишіть Oracle Applications (Oracle E-Business Suite / Oracle Cloud / Oracle Applications) як систему управління ресурсами підприємства: які модулі та можливості для аналітики вона дає?

5. Порівняйте локальні, середні та великі інтегровані інформаційні системи за критеріями масштабованості, витрат впровадження і складності інтеграції

## **Тема 13: Аналіз трендів і часових рядів.**

### **13.1. Основні поняття часових рядів**

Системний аналіз - це міждисциплінарний підхід до вивчення складних систем, що передбачає інтеграцію різних методів для розуміння структури, поведінки та взаємозв'язків елементів системи. Аналіз часових рядів є одним із ключових інструментів системного аналізу, оскільки дозволяє досліджувати динаміку змін у часі, виявляти закономірності та прогнозувати майбутні стани системи.

Прогноз пов'язаний з певним майбутнім станом і (або) шляхами та термінами його досягнення. Прогноз ґрунтується на проведенні певного дослідження, деякого обґрунтування. Прогноз має ймовірнісний характер, тому він не може мати директивний характер [40].

Управлінські рішення часто приймаються в умовах невизначеності та змінного середовища. Аналіз часових рядів надає можливість:

- виявляти тренди та сезонні коливання, що допомагає адаптувати стратегії управління.
- прогнозувати майбутні події на основі історичних даних, що зменшує ризики та покращує планування.
- оцінювати ефективність впроваджених заходів шляхом аналізу змін ключових показників у часі.

Таким чином, аналіз часових рядів сприяє обґрунтованому прийняттю рішень та підвищує адаптивність організацій до змін.

Часовий ряд - це послідовність числових значень, упорядкованих у часі та зафіксованих через рівні інтервали. Кожне спостереження в такому ряді відповідає певному моменту часу. Окремі спостереження часового ряду називають його *рівнями* або *елементами*. Кожен рівень ряду відповідає певному моменту часу. Рівні ряду можуть набувати як детермінованих, так і випадкових значень. Порядок розташування рівнів є істотною характеристикою ряду і не може змінюватися довільно.

Часові ряди відрізняються від простих статистичних вибірок, адже мають такі *ознаки*:

- послідовні в часі показники часових рядів є взаємозалежними, особливо це відноситься до близько розташованих спостережень;

- залежно від моменту спостереження показники часового ряду мають різну інформативність, інформаційна цінність спостережень зменшується за ступенем їх віддалення від поточного моменту часу;

- зі збільшенням кількості показників часового ряду точність статистичних характеристик не збільшується пропорційно числу спостережень, а за появи нових закономірностей розвитку вона може навіть зменшуватися.

*Аналіз часових рядів* – сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування. Виявлення структури часового ряду необхідно для того, щоб побудувати математичну модель того явища, яке є джерелом часового ряду.

Аналіз часових рядів дозволяє виявити закономірності в динаміці даних, прогнозувати майбутні значення та приймати обґрунтовані управлінські рішення.

Якщо кожному моменту часу відповідають значення лише одного показника, ряд є *одновимірним*. Іноді кожному моменту часу приводять у відповідність декілька значень різних показників досліджуваного об'єкта. Тоді отримують *багатовимірний часовий ряд*.

Розрізняють також *детерміновані* та *випадкові* часові ряди. У першому випадку майбутнє значення часового ряду можна точно визначити за допомогою певної функції – моделі часового ряду. У другому випадку зі значенням часового ряду пов'язаний певний закон розподілу випадкової величини. Визначення випадкового часового ряду опирається на поняття *випадкової функції*  $\xi(t)$ , що залежить від змінної  $t$ , яка інтерпретується як час. Тобто мова йде про однопараметричне сімейство випадкових величин  $\xi(t)$ .

Компоненти часових рядів. Часовий ряд можна розкласти на кілька складових, кожна з яких відображає певний тип змін у даних.

*Тренд* або *системна компонента* ( $T$ ) – характеризує загальну тривалу тенденцію зміни динамічного ряду (економічного процесу) в довгостроковому періоді. Він є результатом стійких змін, що відбуваються в економічному середовищі та всередині самої системи. Тренд може бути зростаючим або спадним, але, незважаючи на це, він представляє собою гладку криву (рис. 13.1).

Тренд є *основою* прогнозованого числового ряду, на яку вже накладаються інші складові. Поряд із тривалими тенденціями у часових рядах економічних процесів часто мають місце регулярні коливання, тобто періодичні складові рядів динаміки. Якщо період коливань не перевищує одного року, то їх називають сезонними. За більшого періоду коливання вважають, що в часових рядах має місце циклічна складова.

*Сезонність* або *сезонна компонента* ( $S$ ) – повторення даних через визначені проміжки часу: дні, тижні, місяці чи квартали (найчастіше термін «сезонність» відноситься до настання зими, весни, літа й осені). Сезонні коливання відбуваються через погодні умови, календарні події. Вони

демонструють більш-менш стабільну схему змін, що повторюється з року в рік (рис. 13.2).

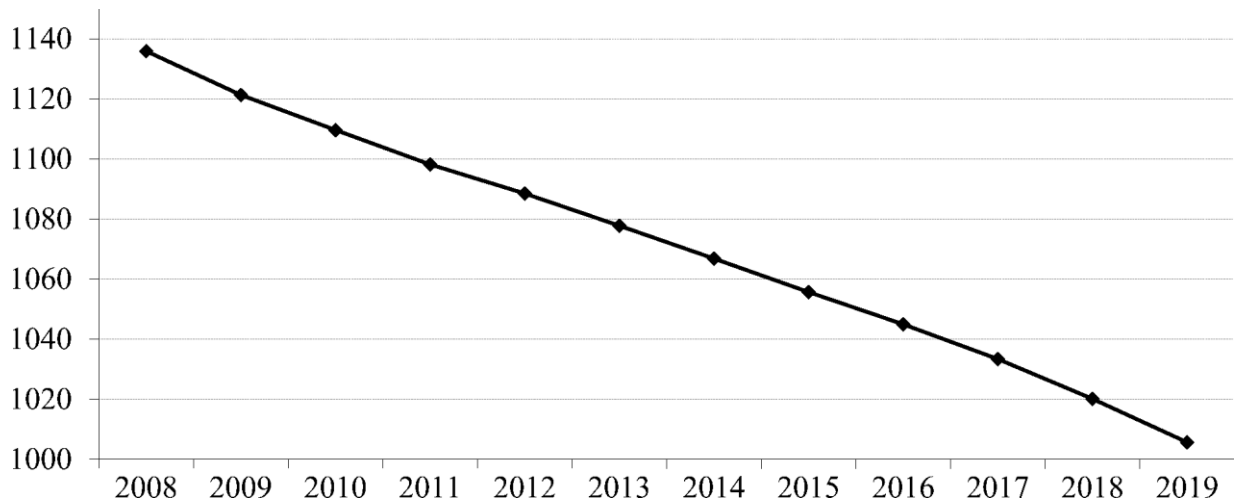


Рис. 13.1. Приклад трендової компоненти часового ряду

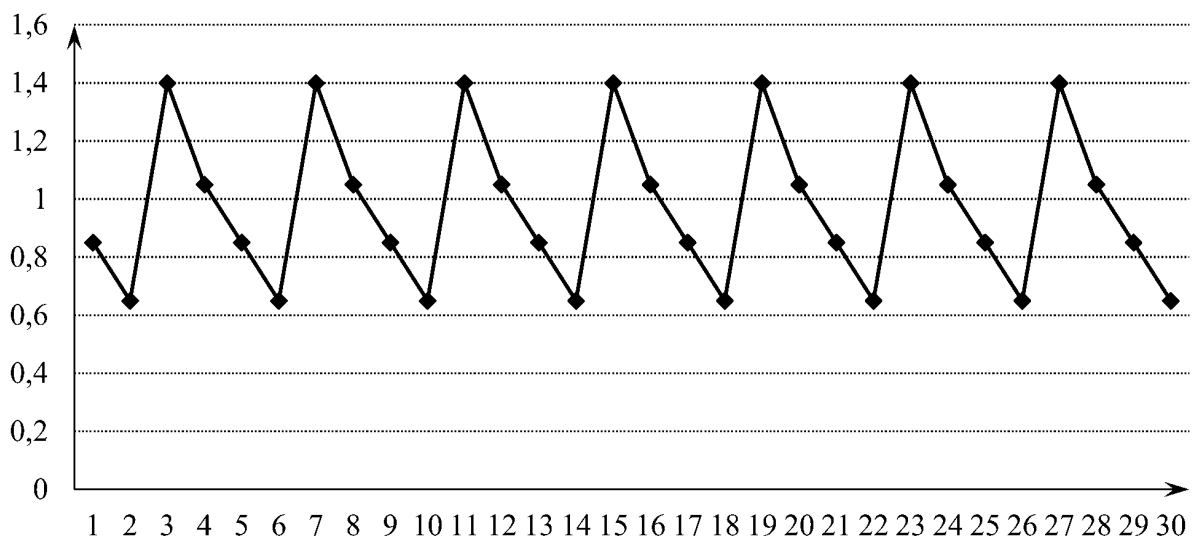


Рис. 13.2. Приклад сезонної компоненти часового ряду

**Цикли** або **циклічна компонента (C)** – це серія хвилеподібних коливань даних, що зустрічаються періодично і охоплюють періоди, як правило, в декілька років, відображуючи економічні підйоми та спади. Цикли не є регулярними чи стійкими, а в основному, пов’язані з циклами в бізнесі і, головним чином, важливі при короткостроковому аналізі та плануванні (рис. 13.3).

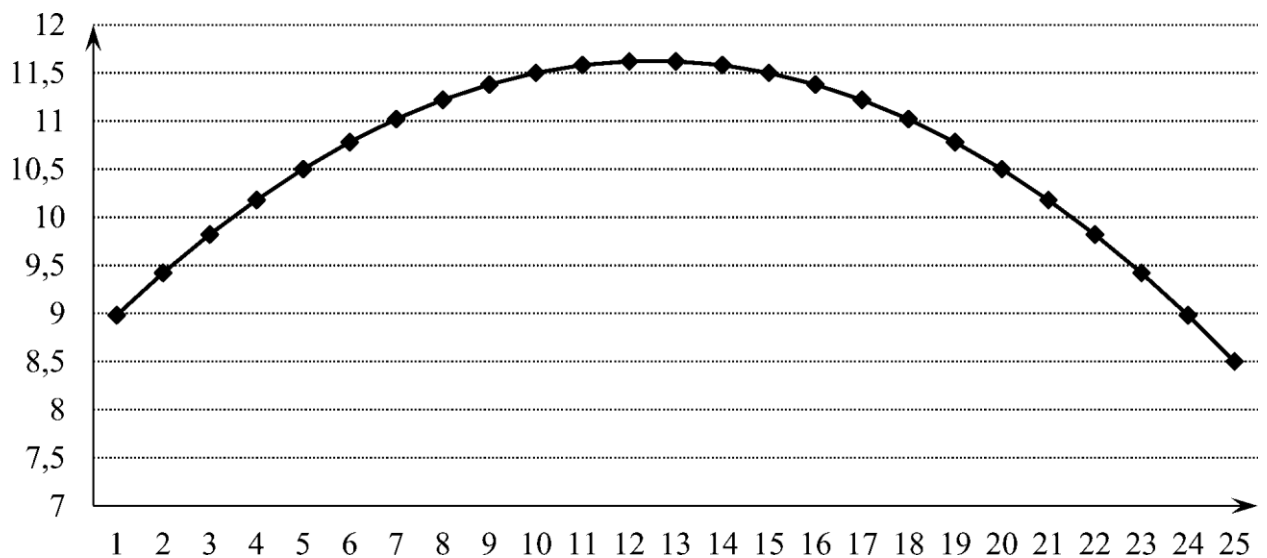


Рис. 13.3. Приклад циклічної компоненти часового ряду

**Випадкові варіації ( $R$ )** або **ірегулярні події (випадкова компонента)** – це відхилення даних, пов’язані з непередбачуваними і незвичайними ситуаціями: страйками, стихійними кліматичними чи природними явищами, конфліктами або війнами (рис. 13.4).

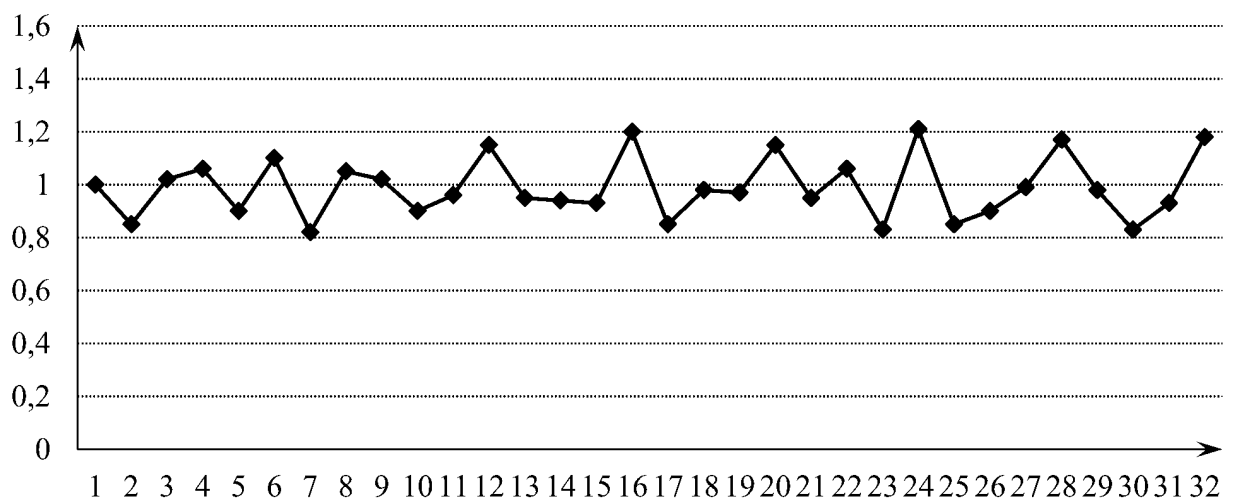


Рис. 13.4. Приклад випадкової компоненти часового ряду

Ці коливання є результатом тисяч подій, кожна з яких сама по собі може не бути особливо важливою, але ефект їх комбінації може виявитися великим. По суті, вона відображує ефект усіх факторів окрім тренду, сезонних та циклічних коливань. Саме ці нерегулярні сили є занадто непередбачуваними для прогнозування.

Аналіз трендів та сезонності - це прямий, інтуїтивний підхід до оцінки чотирьох основних компонентів місячного або квартального часового ряду: довгострокового тренду», сезонності, циклічної варіації та нерегулярної складової.

Базова модель часового ряду представляє числа в цьому ряду у вигляді добутку, отриманого шляхом множення перерахованих компонентів.

$$\text{Дані} = \text{Тренд} \times \text{Сезонність} \times \text{Циклічність} \times \text{Нерегулярність}$$

Тренд і циклічна складова: ковзне середнє.

Ковзне середнє - це новий ряд, отриманий шляхом усереднення сусідніх спостережень часового ряду і переходу до наступного часового відрізка - в результаті виходить більш плавний ряд.

$$\text{Ковзне середнє} = \text{Тренд} \times \text{Циклічність}$$

Сезонний індекс: середнє значення відношення до ковзної середньої відображає сезонну поведінку

Щоб розрізнити сезонну поведінку, перш за все, необхідно отримати відношення початкових значень до ковзної середньої. Результат буде включати сезонну і нерегулярну складові, так як ковзне середнє виключає тренд і циклічну складову з даних.

$$(\text{Сезонність}) \times (\text{Нерегулярність}) = (\text{Данні} / \text{Ковзне середнє})$$

Потім, щоб усунути нерегулярний компонент, треба усереднити ці значення для кожного сезону. Сезонний компонент проявляється, оскільки він присутній щорічно, тоді як нерегулярний компонент, як правило, вдається усереднити.

$$\text{Сезонний індекс} = \text{Середнє значення} (\text{Данні} / \text{Ковзне середнє}^*)$$

\*- за відповідний сезон

Поправка на сезон: ділення ряду на сезонний індекс.

Сезонне коригування знімає з результатів вимірювань очікувану сезонну складову (шляхом ділення ряду на сезонний індекс за відповідний період), що дозволяє безпосередньо порівнювати один квартал або місяць з іншим (після коригування на сезон), виявляючи певні приховані тенденції

$$\begin{aligned} \text{Значення із поправкою на сезон} &= (\text{Данні} / \text{Сезонний індекс}) = \\ &= \text{Тренд} \times \text{Циклічність} \times \text{Нерегулярність} \end{aligned}$$

При екстраполяції закономірностей розвитку будь-який процес з усіма властивими йому особливостями розглядається як функція часу. При цьому час не є фактором соціально-економічного процесу, він просто акумулює весь комплекс постійно діючих умов і причин, які визначають цей процес.

### 13.2. Методи аналізу часових рядів.

Візуалізація трендів. Першим кроком у аналізі трендів є візуалізація даних: побудова графіків часових рядів дозволяє виявити загальні тенденції, сезонні коливання та аномалії. Це надає початкове уявлення про структуру даних і допомагає вибрати відповідні методи аналізу [41].

У більшості випадків фактичний рівень часового ряду можна представити як суму або добуток трендової, циклічної та випадкової компонент.

Модель, у якій часовий ряд представлений як сума перерахованих компонент, називається адитивною моделлю часового ряду:

$$y_t = u_t + s_t + v_t + e_t,$$

де  $y_t$  - рівні часового ряду;  $u_t$  - часовий тренд;  $s_t$  - сезонний компонент;  $v_t$  - циклічна складова;  $e_t$  - випадкова складова.

Модель, в якій часовий ряд представлений як добуток перерахованих компонент, називається мультиплікативною моделлю:

$$y_t = u_t s_t v_t e_t.$$

Існує також мішана модель часового ряду:

$$y_t = u_t s_t v_t + e_t.$$

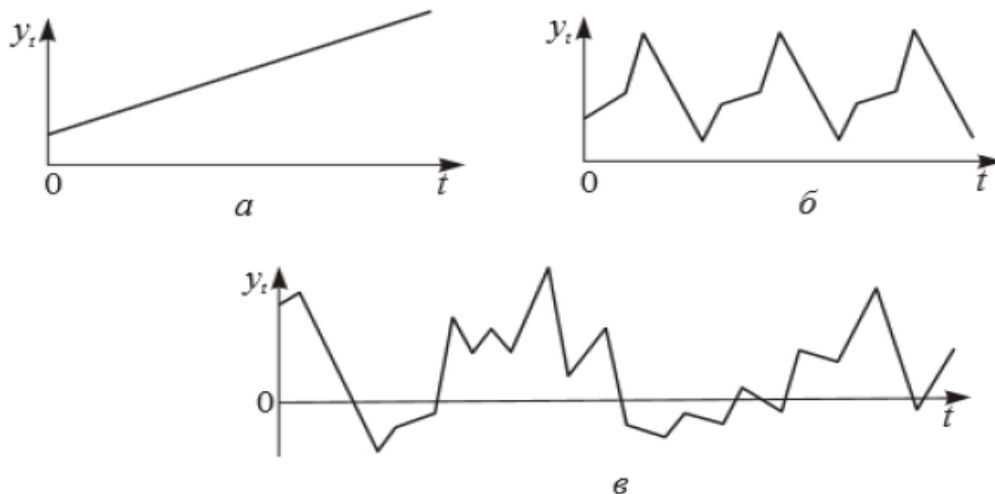


Рис. 13.5 – Головні компоненти часового ряду: а - тренд, що зростає; б - сезонна компонента; в - випадкова компонента.

Метод Ірвіна для виявлення аномальних рівнів часових рядів.

Поява аномальних значень може бути спричинена:

- технічними похибками при зборі, обробці, передачі інформації - помилки першого роду.

- аномальні значення можуть бути властивими реальним процесам (наприклад, падіння курсів валют) - це помилки другого роду, що не можна видалити

Нехай маємо часовий ряд:

$$\tilde{y}_t = y_1, y_1, \dots, y_n$$

Основна формула методу Ірвіна:

$$\lambda_t = |y_t - y_{t-1}|, t = \overline{1, n}$$

Де  $\sigma_y$  = середнє квадратичне відхилення часового ряду,

$$\sigma_y = \sqrt{\sum_i (y_i - \bar{y})^2 / (n - 1)}$$

Розрахункові значення  $\lambda_1, \lambda_2$  порівнюються з табличними значеннями критерію Ірвіна  $\lambda_\alpha$ .

Якщо одне з них є більшим за табличне, то відповідне значення у рівня ряду вважається аномальним.

Значення критерію Ірвіна при рівні значимості  $\alpha = 0,05$  наведено в таблиці 13.1.

Таблиця 13.1 – Значення критерію Ірвіна при рівні значимості  $\alpha = 0,05$

$n$	2	3	10	20	30	50	100
$\lambda_\alpha$	2,8	2,3	1,5	1,3	1,2	1,1	1,0

Після того, як виявлено аномальні рівні необхідно визначити причини їх виникнення. Якщо вони є наслідком технічних помилок, то можливі два шляхи вирішення проблеми: - заміна аномальних рівнів відповідними значеннями на кривій, що є апроксимацією часового ряду, - заміна аномалій середньою арифметичною двох сусідніх до них рівнів ряду. Помилки, що виникають під дією факторів об'єктивного характеру виправити неможливо.

Метод ковзного середнього (Moving Average). Методи згладжування використовуються для зменшення впливу випадкового компонента (випадкових коливань) у часових рядах. Вони дають можливість отримувати більш "чисті" значення, які складаються лише з детермінованих компонентів. Деякі з методів направлені на виділення лише деяких компонентів, наприклад, тренду.

До основних методів аналізу часових рядів можна віднести метод ковзного середнього, експоненціального згладжування і проектування трендах.

Метод ковзного середнього, є одним з найпростіших, який дозволяє виділити тренд. Для застосування цього методу дослідник повинен мати

доволі довгий ряд спостережень. При визначені тенденції за допомогою ковзаючої середньої треба сформувати укрупнені інтервали, що складаються з однакової кількості рівнів. Кожен наступний інтервал отримують, поступово рухаючись від початкового рівня динамічного ряду на один рівень. За сформованим більш крупним інтервалом визначають суми значень рівнів і розраховують плинні середні. При згладжуванні ряду за допомогою ковзаючої середньої більш крупний інтервал складають з непарної кількості рівнів ряду (3, 5, 7).

Для розгляду, як працюють ці методи, користуватимемося одним і тим же часовим рядом. Метод ковзного середнього згладжує короткострокові коливання, дозволяючи виявити довгострокові тенденції. Суть методу полягає в обчисленні середнього значення для певного числа послідовних спостережень.

Метод ковзного середнього:

$$MA_t = \frac{1}{n} \sum_{i=0}^{n-1} y_{t-i}$$

де:  $MA_t$  - ковзне середнє в момент часу  $t$ ;  $n$  - кількість періодів у вікні згладжування;  $y_{t-i}$  - значення часового ряду в момент часу  $t-i$ .

Цей метод ефективний для виявлення трендів у даних без вираженої сезонності.

Метод найменших квадратів (Least Squares Method). Метод найменших квадратів використовується для побудови трендової лінії, яка мінімізує суму квадратів відхилень фактичних значень від прогнозованих. Цей метод дозволяє оцінити параметри різних типів трендів: лінійного, експоненційного, логарифмічного тощо.

Лінійний тренд:

$$y_t = a + b_t + \varepsilon_t$$

де:  $y_t$  - значення часового ряду в момент часу  $t$ ;  $a$  - початковий рівень;  $b_t$  - коефіцієнт тренду (нахил лінії);  $\varepsilon_t$  - випадкова похибка.

Оцінка параметрів  $a$  і  $b$  здійснюється шляхом мінімізації суми квадратів похибок.

Експоненціальне згладжування (Exponential Smoothing).

Експоненціальне згладжування надає більшу вагу останнім спостереженням, що дозволяє швидше реагувати на зміни в даних. Цей метод особливо корисний для прогнозування в умовах, коли останні дані мають більшу інформативність.

Експоненціальне згладжування:

$$S_t = \alpha y_t + (1 - \alpha) S_{t-1}$$

де:  $S_t$  - згладжене значення в момент часу  $t$ ;  $y_t$  - фактичне значення в момент часу  $t$ ;  $\alpha$  - коефіцієнт згладжування ( $0 < \alpha < 1$ ).

Вибір оптимального значення  $\alpha$  залежить від характеру даних і бажаного рівня згладжування.

Поліноміальна апроксимація (Polynomial Approximation). Поліноміальна апроксимація використовується для моделювання нелінійних трендів, коли зміни в даних не можуть бути адекватно описані лінійною моделлю. Цей метод передбачає побудову полінома певного ступеня, який найкраще описує тенденцію в даних.

Загальна форма:

$$y_t = a_0 + a_1t + a_2t^2 + \dots + a_nt^n + \varepsilon_t$$

де:  $a_0, a_1, \dots, a_n$  - коефіцієнти полінома;  $n$  - ступінь полінома;  $\varepsilon_t$  - випадкова похибка.

Вибір ступеня полінома залежить від складності тренду та обсягу даних.

Оцінка лінійного, логарифмічного, експоненційного тренду. Різні типи трендів можуть бути оцінені за допомогою відповідних моделей:

Лінійний тренд:

$$y_t = a + b_t$$

Експоненційний тренд:

$$y_t = ae^{bt}$$

Логарифмічний тренд:

$$y_t = a + b \ln(t)$$

Оцінка параметрів цих моделей здійснюється методом найменших квадратів після відповідних перетворень даних (наприклад, логарифмування для експоненційного тренду).

Інтерпретація коефіцієнтів тренду у прикладному контексті.

Коефіцієнти трендових моделей мають практичне значення:

- Коефіцієнт  $b$  у лінійній моделі  $y_t = a + b_t$
- Визначає середню зміну показника за одиницю часу. Наприклад, якщо  $b = 5$ , це означає, що показник зростає на 5 одиниць щомісяця.
- Коефіцієнт  $b$  у експоненційній моделі  $y_t = ae^{bt}$
- Визначає темп зростання або спадання. Наприклад, якщо  $b = 0.02$ , це відповідає приблизно 2% зростанню за період.
- Коефіцієнт  $b$  у логарифмічній моделі  $y_t = a + b \ln(t)$
- Вказує на зменшення темпу зростання з часом. Наприклад, у випадку насичення ринку, коли зростання сповільнюється.
- Розуміння та інтерпретація цих коефіцієнтів дозволяє керівникам приймати обґрунтовані рішення щодо стратегічного планування, прогнозування та оптимізації бізнес-процесів.

### 13.3. Методи декомпозиції часових рядів

Основи декомпозиції. Декомпозиція часових рядів - це процес розкладу складного часового ряду на окремі складові, що дозволяє краще розуміти структуру даних та полегшує прогнозування [42].

Основні компоненти часового ряду:

Тренд ( $T_t$ ): довгострокова тенденція зміни рівня ряду.

Сезонність ( $S_t$ ): регулярні коливання, пов'язані з календарними або іншими періодичними факторами.

Циклічна складова ( $C_t$ ): довгострокові коливання, зумовлені економічними циклами.

Випадкова складова ( $E_t$ ): нерегулярні, випадкові коливання.

Залежно від характеру взаємодії між цими компонентами, використовуються різні моделі декомпозиції.

Адитивна та мультиплікативна моделі.

Адитивна модель:

$$Y_t = T_t + S_t + C_t + E_t$$

Ця модель передбачає, що вплив кожної компоненти є незалежним і додається до загального результату. Використовується, коли амплітуда сезонних коливань не залежить від рівня тренду.

Мультиплікативна модель:

$$Y_t = T_t \times S_t \times C_t \times E_t$$

У цій моделі компоненти взаємодіють мультиплікативно, тобто зміни в одній компоненті впливають на інші. Застосовується, коли амплітуда сезонних коливань зростає або зменшується разом із рівнем тренду.

Мультиплікативна модель часто використовується в економічних часових рядах, де сезонні коливання пропорційні до рівня ряду.

Класична декомпозиція. Класична декомпозиція передбачає послідовне виділення компонент часового ряду:

Оцінка тренду: зазвичай за допомогою ковзного середнього.

Виділення сезонності: обчислення середніх сезонних ефектів.

Оцінка залишкової складової: отримується шляхом віднімання або ділення тренду та сезонності від початкового ряду.

У R реалізація класичної декомпозиції доступна через функцію `decompose()`, яка дозволяє вибрати тип моделі (адитивна або мультиплікативна).

STL (Seasonal-Trend Decomposition using Loess).

STL - це сучасний метод декомпозиції, який використовує локальне згладжування (LOESS) для виділення тренду та сезонності. Основні переваги STL:

Гнучкість: може обробляти складні та змінні сезонні патерни.

Стійкість до викидів: менш чутливий до аномальних значень у даних.

Можливість обробки нерівномірно розподілених даних: ефективний при роботі з даними, що мають пропущені значення або нерегулярні інтервали.

STL широко використовується в економічному та екологічному аналізі для розділення часових рядів на тренд, сезонність та залишкову складову.

### 13.4. Стаціонарні і нестаціонарні часові ряди

Стаціонарний часовий ряд: Характеризується постійними статистичними властивостями (середнє, дисперсія, автокореляція) у часі. Не має тренду або сезонності. Придатний для багатьох статистичних методів аналізу та прогнозування.

Стаціонарність - це властивість часового ряду, при якій його статистичні характеристики (середнє, дисперсія, автокореляція) залишаються постійними в часі. Стаціонарні ряди не мають трендів або сезонних коливань, що робить їх більш передбачуваними та зручними для моделювання [42].

Розрізняють два типи стаціонарності:

Слабка (коваріаційна) стаціонарність: середнє, дисперсія та автокореляція не змінюються з часом.

Сильна (статистична) стаціонарність: повна розподільча функція не залежить від часу.

Більшість статистичних методів аналізу часових рядів вимагають слабкої стаціонарності.

Тест Дікі-Фуллера (Augmented Dickey-Fuller, ADF) - це статистичний тест, який перевіряє наявність одиничного кореня в часовому ряді, тобто визначає, чи є ряд нестаціонарним. Нульова гіпотеза тесту: ряд має одиничний корінь (нестаціонарний). Альтернативна гіпотеза: ряд стаціонарний.

Тестова модель має вигляд:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \dots + \delta_{p-1} \Delta y_{t-1+1} + \varepsilon_t$$

де:  $\Delta y_t$  - перша різниця ряду;  $\alpha$  - константа;  $\beta t$  - трендова складова;  $\gamma y_{t-1}$  - лагове значення ряду;  $\delta_1$  - коефіцієнти лагів перших різниць;  $\varepsilon_t$  - випадкова похибка.

Якщо статистика тесту менша (більш негативна), ніж критичне значення, нульова гіпотеза відхиляється, і ряд вважається стаціонарним.

Перетворення до стаціонарності. Нестационарний часовий ряд має змінні статистичні властивості у часі. Може містити тренд, сезонність або змінну дисперсію. Потребує трансформації (наприклад, диференціювання) для досягнення стаціонарності перед застосуванням певних методів аналізу. Визначення стаціонарності є важливим кроком у аналізі часових рядів,

оскільки багато моделей (наприклад, ARIMA) вимагають стаціонарності даних для коректної роботи.

Якщо ряд є нестационарним, його можна перетворити до стаціонарного стану за допомогою наступних методів:

Диференціювання. Диференціювання полягає у відніманні попереднього значення ряду від поточного:

$$\Delta y_t = y_t - y_{t-1}.$$

Це усуває трендову складову та стабілізує середнє значення ряду. Якщо після першого диференціювання ряд все ще нестационарний, можна застосувати друге диференціювання.

Логарифмування. Логарифмування допомагає стабілізувати дисперсію ряду, особливо коли коливання зростають з часом. Застосовується до позитивних значень ряду:

$$y'_t = \log(y_t).$$

Це перетворення часто використовується перед диференціюванням для покращення стаціонарності.

### 13.5. Короткий огляд моделей AR, MA, ARMA, ARIMA

ARIMA-процеси Бокса-Дженкінса являють собою сімейство лінійних статистичних моделей, заснованих на нормальному розподілі, які дозволяють моделювати поведінку багатьох різних рядів реального часу шляхом об'єднання процесів авторегресії, інтеграційних процесів і процесів ковзного середнього.

ARIMA - скорочення від Autoregressive Integrated Moving Average.

**Випадковий шумовий процес не має пам'яті: відправна точка**

Процес випадкового шуму складається з випадкової вибірки (незалежних спостережень) нормального розподілу з постійним середнім і стандартним відхиленням.

Тенденцій у цьому випадку немає, адже - через незалежність - спостереження не пам'ятають минулої поведінки серіалу.

Модель має вигляд:

$$Y_t = \mu + \zeta_t.$$

де  $\mu = 0$  - середнє значення, а  $\zeta_t$  - незалежні випадкові значення з нормального розподілу.

Характеристики білого шуму: немає пам'яті: кожне значення не залежить від попередніх. Стаціонарність - середнє і дисперсія залишаються

постійними з часом. Випадковість - значення коливаються навколо середнього без видимої структури або тренду.

AR (Autoregressive) - авторегресійна модель. Процес авторегресії і ковзного середнього (ARMA) поєднує AR і MA.

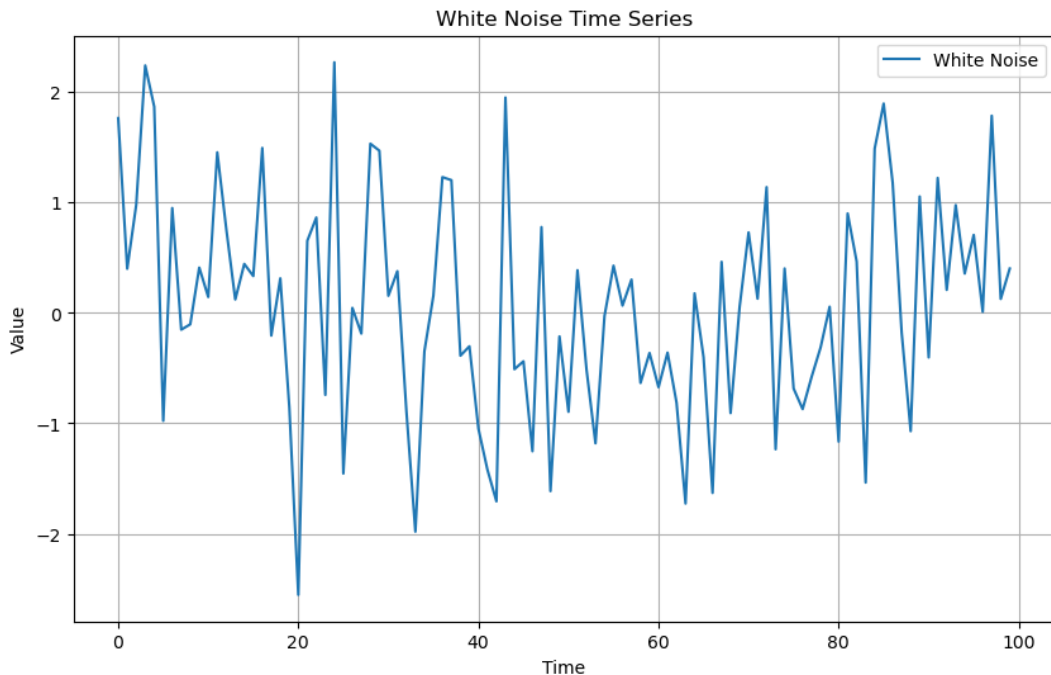


Рис. 13.6. Випадковий шумовий процес не має пам'яті

Будь-яке спостереження за процесом авторегресії і ковзною середньою складається з лінійної функції з попереднього спостереження плюс незалежний випадковий шум мінус деяка частка попереднього випадкового шуму. Процес авторегресії і ковзної середньої запам'ятовує як свій попередній стан, так і випадкову шумову складову попереднього стану. Таким чином, його пам'ять об'єднує пам'ять про процес авторегресії з пам'яттю процесу ковзної середньої.

Модель порядку  $p$  (AR( $p$ )) описує поточне значення часового ряду як лінійну комбінацію його попередніх  $p$  значень та випадкової похибки.

$$Y_t = \delta + \sum_{i=1}^p \varphi_i Y_{t-i} + \zeta_t.$$

Де:  $X_t$  - поточне значення ряду;  $\delta$  – константа;  $\varphi_i$  - коефіцієнти моделі;  $\zeta_t$  - випадкова похибка (білий шум).

Модель авторегресії з ковзним середнім (ARMA). ARMA( $p, q$ ) модель поєднує в собі авторегресійну частину порядку  $p$  та компоненту ковзного середнього порядку  $q$ :

$$Y_t = \delta + \sum_{i=1}^p \varphi_i Y_{t-i} + \sum_{j=1}^q \theta_j \zeta_{t-j} + \zeta_t.$$

Де:  $\theta_j$  - коефіцієнти ковзного середнього.

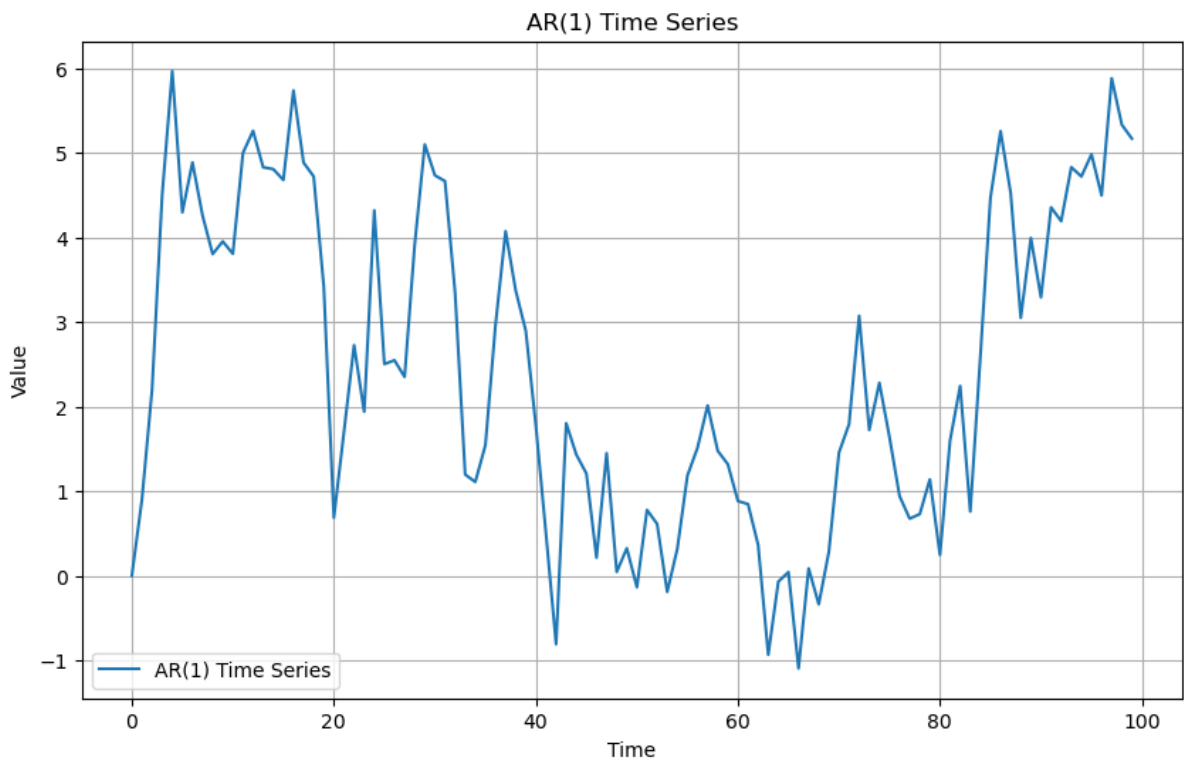


Рис. 13.7. Авторегресійна модель

Чистий інтегрований (i) процес. Чистий інтегрований (i) процес запам'ятовує, де він був, а потім рухається випадковим чином.

Кожне спостереження чистого інтегрованого (I) процесу, також званого випадковою ходою, є випадковим кроком від поточного спостереження. Цей процес знає, де він знаходиться, але забуває, як він туди потрапив.

Модель, в якій поточне значення ряду залежить від його попередніх значень.

$$Y_t = \delta + \varphi Y_{t-1} + \zeta_t.$$

Порівняння з іншими моделями: На відміну від білого шуму, цей процес нестационарний. На відміну від AR(1), тут  $\varphi = 1$ , що означає повну залежність від попереднього значення без затухання.

MA (Moving Average) - модель ковзного середнього:

Процес інтегрованого ковзного середнього процесу авторегресії (ARIMA) пам'ятає свої зміни.

Процес складається з лінійної функції попередньої зміни плюс незалежний випадковий шум мінус певна частка попереднього випадкового шуму. Цей процес знає, де він знаходиться, запам'ятовує, як потрапив в такий стан, і запам'ятовує навіть частину попередньої шумової складової.

Модель ковзного середнього:

$$Y_t - Y_{t-1} = \delta + \varphi(Y_{t-1} - Y_{t-2}) + \zeta_t + \theta\zeta_{t-1}.$$

де  $\delta$  - стала складова,  $\varphi$  - коефіцієнт залежності від попередньої зміни,  $\theta$  - коефіцієнт залежності від попереднього шуму,  $\xi_t$  - білий шум.

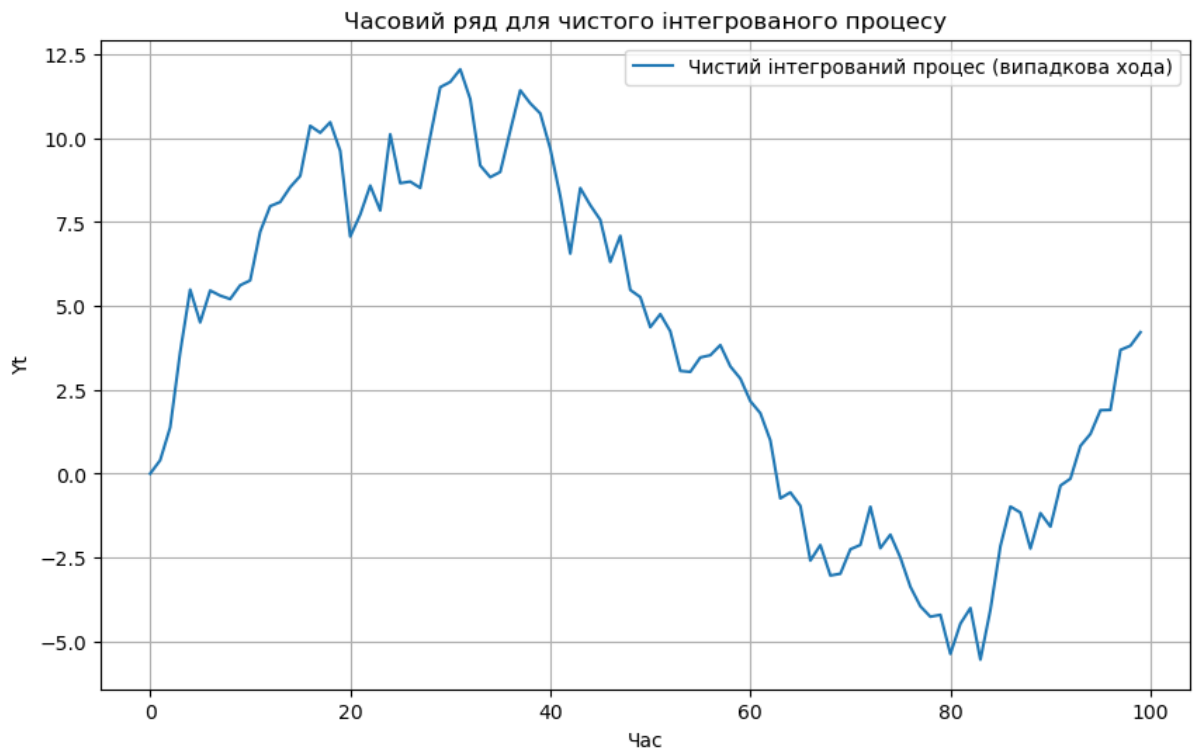


Рис. 13.8. Чистого інтегрований (I) процес

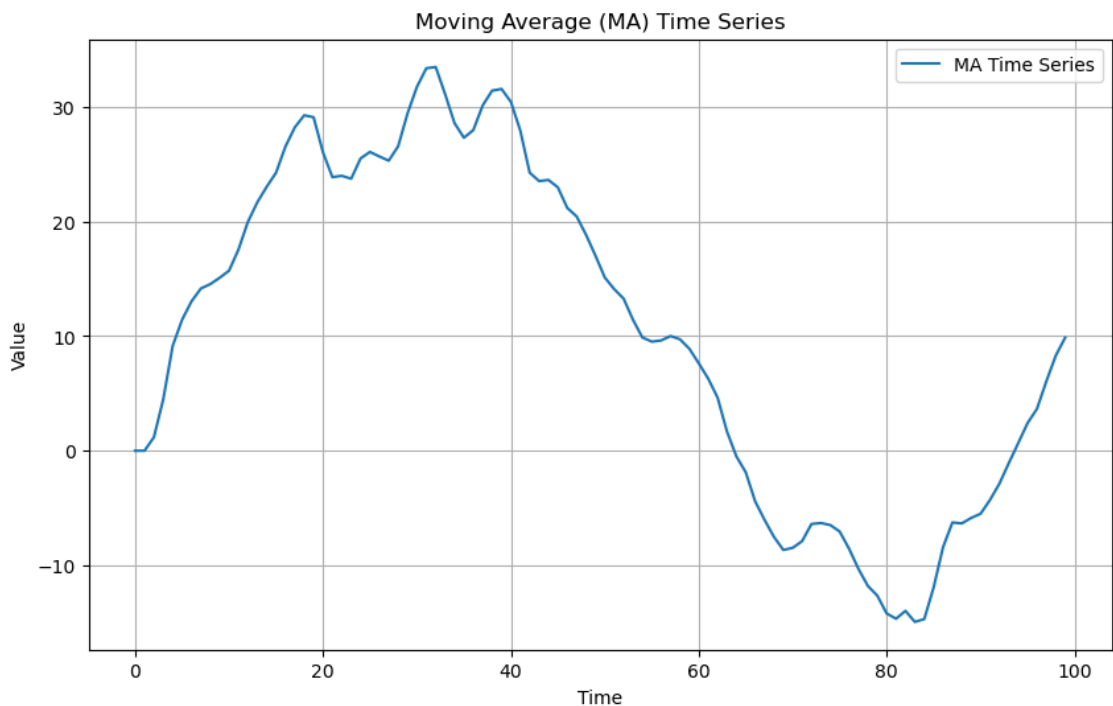


Рис. 13.9. Модель ковзного середнього

Цей процес пам'ятає попередню зміну (через  $\phi$ ) і частково враховує попередній шум (через  $\theta$ ). На відміну від випадкової ходи, тут є структура в

зміні значень, що робить ряд більш передбачуваним у короткостроковій перспективі. Така модель є частиною ширшого класу ARIMA, де поєднуються авторегресія, інтегрування (диференціювання) та ковзне середнє.

Аналіз часових рядів є фундаментальним інструментом у системному аналізі, оскільки дозволяє: виявляти закономірності та тенденції розкриття трендів, сезонних коливань та циклічних змін у даних. Прогнозувати майбутні значення на основі історичних даних можна будувати прогнози, що сприяють ефективному плануванню та управлінню ресурсами. Підвищувати обґрунтованість рішень при використанні статистичних моделей зменшує вплив суб'єктивних факторів у прийнятті рішень. Правильний аналіз часових рядів забезпечує надійні прогнози, що є критично важливими для стратегічного планування та оперативного управління в різних сферах діяльності.

### **ПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ ТА ОБГОВОРЕННЯ**

1. Дайте визначення часового ряду. Поясніть складові часових рядів: тренд, сезонність, циклічна і випадкова компоненти.
2. Назвіть і коротко опишіть методи аналізу часових рядів (гладіння - просте й експоненційне, декомпозиція, модель ARIMA, ETS). Коли доцільно застосувати ARIMA?
3. Поясніть процедуру тестування стаціонарності ряду (ADF-тест, KPSS) та чому стаціонарність важлива для ARMA/ARIMA-моделей.
4. Наведіть базовий алгоритм побудови ARIMA(p,d,q): як вибрати порядок p і q (ACF/PACF), як визначити d?
5. Для практичної задачі прогнозування продажів: опишіть етапи попередньої обробки даних (очищення, заповнення пропусків, розкладання на компоненти), вибір моделі і оцінку якості прогнозу (MAE, RMSE, MAPE).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gunderson L. H. and Holling C. S., Panarchy : Understanding Transformations in Human and Natural Systems, Washington, DC : Island Press, 2002. 507 с.
2. Грицюк П. М., Джоші О. І., Гладка О. М. Основи теорії систем і управління навч. посіб. Рівне : НУВГП, 2021. 272 с.
3. Системний аналіз інформаційних процесів : навч. посіб. / В. М. Варенко, І. В. Братусь, В. С. Дорошенко, Ю. Б. Смольников, В. О. Юрченко. Київ: Університет "Україна", 2013. 203 с.
4. Навчальний посібник з дисципліни «Системний аналіз» для здобувачів спеціальності 122 – Комп'ютерні науки / Укл.: В.М. Тонконогий, В.О. Вайсман, Л.В. Бовнегра, К.Г. Кіркопуло. Одеса : Нац. ун-т «Одеська політехніка», 2022. 84 с.
5. Горбань О.М., Бахрушин В.Є. Основи теорії систем і системного аналізу : навч. посіб. Запоріжжя : ГУ "ЗІДМУ", 2004. 204 с.
6. Tilley S., & Rosenblatt H. J. Systems Analysis and Design. Boston : Cengage Learning EMEA, 2021. 752 с.
7. Герасимчук О.П. Теорія технічних систем : навч. посіб. для здобувачів другого (магістерського) рівня вищої освіти денної та заочної форм навчання. Луцьк : ЛНТУ, 2023. 112 с.
8. Измайлова О. В. Проектування інформаційних систем : навч. посіб. Київ : КНУБА, 2022. 88 с.
9. Новожилова М. В., Чуб О. І. Методи та засоби прийняття рішень : навч. посіб. – Харків : ХНУМГ ім. О. М. Бекетова, 2024. 115 с.
10. Ситник Г.П., Комаха Л.Г., Рудик А.О. Основи теорії систем та системного аналізу : навч. посіб. Київ, 2024. 160 с.
11. Згуровський М.З., Панкратова Н.Д. Основи системного аналізу : навч. посіб. Київ : Видавнича група ВНУ, 2007. 544 с.
12. Кириченко О.О., Кириченко О.Л. Моніторинг та оцінка якості ІТ-проектів : навч. посіб. Чернівці : Чернівець. нац. ун-т ім. Ю. Федьковича. 2024. 104 с.
13. Желдак Т.А., Владико О.Б. Машинне навчання [Електронний ресурс] : конспект лекцій для здобувачів ступеня магістра зі спеціальності 124 Системний аналіз. Дніпро : НТУ «ДП», 2024. 162 с. url: <http://ir.nmu.org.ua/handle/123456789/167978>
14. Табунщик Г.В, Кудерметов Р.К., Каплієнко Т.І. Інженерія якості програмного забезпечення : навч. посіб. 2-ге вид. Запоріжжя : Дике Поле, 2016. 176 с.
15. Коломієць С. В. Теорія випадкових процесів : практикум. Суми : ДВНЗ "УАБС НБУ", 2011. 80 с.
16. Випадкові процеси: теорія, статистика, застосування : підручник / Ю. С. Мішура, К. В. Ральченко, Л. М. Сахно, Г. М. Шевченко. Київ : ВПЦ "Київський університет", 2023. 496 с.

17. Барабаш О. В., Свинчук О. В., Мусієнко А. П. Математичне моделювання та оптимізація процесів і систем [Електронний ресурс] : навч. посіб. Ч.1. Київ : КПІ, 2023. 160 с.
18. Томашевський, В. М. Моделювання систем : підручник. – Київ : Видавнича група ВНУ, 2005. 352 с.
19. Грабовецький Б. Є. Методи експертних оцінок: теорія, методологія, напрями використання : монографія. Вінниця : ВНТУ, 2010. 171 с.
20. Теорія систем керування : підручник / В.І. Корнієнко, О.Ю. Гусєв, О.В. Герасіна, В.П. Щокін. Дніпро : НГУ, 2017. 497 с.
21. Панкратова, Н. Д. Системний аналіз: теорія та застосування. Київ : КПІ, 2020. 450 с.
22. Литвинов А. Л. Теорія систем масового обслуговування : навч. посіб. Харків : ХНУМГ ім. О. М. Бекетова, 2018. 141 с.
23. Імітаційне моделювання систем масового обслуговування : навч. посіб. [для студентів техн. спец. вищ. навч. закл.] / В. Б. Толубко, А. Д. Кожухівський, В.В. Вишнівський, Г.І. Гайдур, О.А. Кожухівська. Київ : ДУТ, 2018. 175 с.
24. Хмарні технології : навч. посіб. О. В. Зінченко, С. М. Іщеряков, С. В. Прокопов, С. О. Серих, В. В. Василенко, Київ : ФОП Гуляєва В.М., 2020. 74 с.
25. Петрик М.Р. Петрик О.Ю. Моделювання програмного забезпечення : науково-метод. посіб. Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. 200 с.
26. Троцько В.В., Теорія алгоритмів : навч.- метод. посіб. Київ : Університет економіки та права «КРОК», 2023 124 с.
27. Beizer V. et al. Software Testing Techniques by Boris Beizer: An Overview, Justbooks.org. Canada. Retrieved 05 Apr 2026.
28. Wiegers K. and Joy B. Software Requirements. 3rd ed. Redmond : Microsoft Press. 2013 619 с.
29. Гнатюк Я.С. Еристика та евристика : навч. посіб. ІваноФранківськ : Симфонія форте, 2013. 268 с.
30. Системний аналіз в управлінні розвитком складних систем [Електронний ресурс] : навч. посіб. В.М. Молоканова, А.В. Малієнко, М.М. Одновол, О.Б. Владико. Дніпро : НТУ «ДП», 2024. 95 с. <http://ir.nmu.org.ua/handle/123456789/167908>
31. Смержевський Ю. Л. Прийоми евристичної діяльності учнів при вивченні геометрії: диференційовані завдання : навч.-мет. посіб. Кам'янець-Подільський : Абетка, 2004. 104 с.
32. Панова С.О. Викладання змістової лінії "Евристичне навчання математики" під час фахової підготовки майбутніх учителів математики [Електронний ресурс] // Вісник психології і педагогіки. Вип. 18. Київ, 2016 – 2017.
33. Верес, О. М., Литвин, В. В. Системний аналіз та проектування : навч. - мет. посіб. Київ : Техніка, 2021. 480 с.

34. Панюнін М. Андреев М. «Лямбда-механізм Чебишева», проєкт демонстрацій 1999. Wolfram. demonstrations. wolfram. Com / Chebyshevs Lambda Mechanism.
35. Літнарівич Р.М. Платформи корпоративних інформаційних систем. Курс лекцій. Рівне, 2012. 130 с.
36. Ярцев В.П. Розподілені бази даних : навч. посіб. Київ : ДУТ, 2018. 97 с.
37. Системний аналіз та проєктування інформаційних систем: С. О. Мащенко, В. Б. Мокін, Л. Л. Гарт, Ю. І. Дорофєєв, В. В. Литвин. Київ : Наука, 2021. 500 с.
38. Kendall K. E. & Kendall J. E. Systems Analysis and Design. Harlow, UK : Pearson, 2021. 752 с.
39. Hoffer J. A., George J. F., Valacich J. S. Modern Systems Analysis and Design. Harlow : Pearson, 2021. 600 с.
40. Холден К., Піл Д. А., Томпсон Дж. Л. Економічне прогнозування: вступ : навч. посіб. Київ : Інформтехніка, ЕМЦ, 1996. 216 с.
41. Костіна Н. І., Алексєєв А. А., Василик О. Д. Фінанси: система моделей і прогнозів : навч. посіб. Київ : Четверта хвиля, 1998. 304 с.
42. Науменко В., Панасюк Б. Впровадження методів прогнозування і планування в умовах ринкової економіки : навч. посіб. Київ : Глобус, 1995. 200 с.
43. Коряшкіна, Л., Малієнко, А., Станіна, О., Шевченко, Ю., Кодола, Я. (2025). Системний аналіз та оптимальний вибір комплексу заходів для підвищення безпеки на підприємстві. Information Technology: Computer Science, Software Engineering and Cyber Security, 72–80, doi: <https://doi.org/10.32782/IT/2025-2-7> url: <https://ir.nmu.org.ua/handle/123456789/172328>
44. Владико, О., Малієнко, А., Мальцев, Д., Хом'як, Т. (2025). Розвиток прогресивних технологій на основі системного підходу до організації. Information Technology: Computer Science, Software Engineering and Cyber Security, 40–51, doi: <https://doi.org/10.32782/IT/2025-3-5>
45. Vladyko, O.; Maltsev, D.; Gliwi ́nski, Ł.; Dychkovskiy, R.; Stecula, K.; Dyczko, A. Enhancing Mining Enterprise Energy Resource Extraction Efficiency Through Technology Synthesis and Performance Indicator Development. Energies 2025, 18, 1641. doi: <https://doi.org/10.3390/en18071641>
46. Системи та методи прогнозування [Електронний ресурс] : методичні рекомендації до виконання практичних робіт для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз / уклад.: Т.В. Хом'як, О.Б. Владико, К.С. Хабарлак, Д.М. Гаранжа ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 56 с. url: <https://ir.nmu.org.ua/handle/123456789/172379>

**Навчальне видання**

**Малієнко** Андрій Вікторович  
**Владико** Олександр Борисович  
**Хом'як** Тетяна Валеріївна  
**Козир** Світлана Василівна

## **СИСТЕМНИЙ АНАЛІЗ**

Навчальний посібник

Видано в авторській редакції.

Електронний ресурс.

Підписано до видання 23. 04. 2026. Авт. арк. 18,6

Підготовлено до видання  
в Національному технічному університеті «Дніпровська політехніка»  
Свідоцтво про внесення до Державного реєстру ДК №1842 від 11.06.2004 р.  
49005, м. Дніпро, просп. Д. Яворницького, 19.