

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

**магістра**

(освітній ступінь)

на тему: **Технології виявлення функціональних та нефункціональних вимог до комп'ютерних систем**

Виконав: студент (ка) 6 курсу, групи СІМ-62  
спеціальності 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

	(підпис)	<b>Шалапай Р.І.</b> (прізвище та ініціали)
Керівник	(підпис)	<b>Тиш Є.В.</b> (прізвище та ініціали)
Нормоконтроль	(підпис)	<b>Луцик Н.С.</b> (прізвище та ініціали)
Завідувач кафедри	(підпис)	<b>Осухівська Г.М.</b> (прізвище та ініціали)
Рецензент	(підпис)	(прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**ЗАТВЕРДЖУЮ**

Завідувач кафедри Осухівська Г.М.

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня магістр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студенту Шалапаю Руслану Ігоровичу  
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) Технології виявлення функціональних та нефункціональних вимог до комп'ютерних систем

Керівник проекту (роботи) Тиш Євгенія Володимирівна, к.т.н.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «01» грудня 2023 року №4/7-1132

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи Типи вимог до комп'ютерних систем, алгоритми кластерного аналізу, алгоритми класифікації

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз методів і підходів в області визначення та аналізу вимог до комп'ютерних систем 2. Методи кластерного аналізу при визначенні функціональних та нефункціональних вимог. 3. Програмна реалізація методу кластеризації функціональних та нефункціональних вимог до комп'ютерних систем 4. Охорона праці та безпека в надзвичайних ситуаціях.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Процес виявлення вимог до КС. 4. Фактори визначення приналежності вимог до класів. 5. Алгоритм K-means. 6. Алгоритм ієрархічної кластеризації. 7. Результати кластеризації. 8. Висновки

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>Осухівська Г.М., зав. каф. КІ</i>		
	<i>Стадник І.Я., проф. каф. ОХ</i>		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Аналіз методів і підходів в області визначення та аналізу вимог до комп'ютерних систем</i>	<i>01.12.2023-05.12.2023</i>	<i>виконано</i>
2.	<i>Методи кластерного аналізу при визначенні функціональних та нефункціональних вимог</i>	<i>05.12.2023-12.12.2023</i>	<i>виконано</i>
3.	<i>Програмна реалізація методу кластеризації функціональних та нефункціональних вимог до комп'ютерних систем</i>	<i>12.12.2023-17.12.2023</i>	<i>виконано</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>18.12.2023</i>	<i>виконано</i>
5.	<i>Оформлення пояснювальної записки</i>	<i>20.12.2023</i>	<i>виконано</i>
6.	<i>Оформлення графічного матеріалу</i>	<i>21.12.2023</i>	<i>виконано</i>
7.	<i>Попередній захист кваліфікаційної роботи магістра</i>	<i>22.12.2023</i>	<i>виконано</i>
8.	<i>Захист кваліфікаційної роботи магістра</i>		

Студент \_\_\_\_\_

(підпис)

*Шалапай Р.І.*

(прізвище та ініціали)

Керівник проекту (роботи) \_\_\_\_\_

(підпис)

*Тим Є.В.*

(прізвище та ініціали)

## АНОТАЦІЯ

Технології виявлення функціональних та нефункціональних вимог до комп'ютерних систем // Кваліфікаційна робота магістра // Шалапай Руслан Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІм-62 // Тернопіль, 2023 // с. – 89, рис. – 27, табл. – 23, аркушів А1 – 8, додат. – 1, бібліогр. – 24.

Ключові слова: технологія, виявлення, вимога, кластеризація, комп'ютерна система.

У кваліфікаційній роботі проаналізовано підходи щодо управління вимогами до комп'ютерних систем на етапах їх життєвого циклу і підтверджено важливу роль відповідних процесів, дій та активностей, оскільки вимоги є фундаментом для подальших стадій розробки комп'ютерних систем.

Запропоновано процедуру і метод агломеративної ієрархічної кластеризації для визначення груп функціональних та нефункціональних вимог до комп'ютерних систем та обгрунтовано доцільність застосування метрик точності і повноти для визначення якості кластеризації вимог до комп'ютерних систем, а також введено метрику порівняння, яка дає змогу визначити відхилення кількості ідентифікованих кластерів, визначених за допомогою методу агломеративної ієрархічної кластеризації та фактично наявних кластерів.

Побудовано програмний конвеєр опрацювання функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу практично реалізувати технології машинного навчання в процесі їх аналізу тим самим скоротивши часові витрати на виконання проекту та підвищивши якість його виконання.

## ABSTRACT

Technologies for detecting functional and non-functional requirements for computer systems /Master's graduation thesis / Shalapai Ruslan / Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm -61 // Ternopil, 2023// p. - 89, fig. – 27, table. – 23, Posters A1 – 8, Add – 1, Ref. – 24.

Keywords: technology, detection, requirement, clustering, computer system

The qualification work analyzed the approaches to managing requirements for computer systems at the stages of their life cycle and confirmed the important role of relevant processes, actions and activities, as requirements are the foundation for further stages of development of computer systems.

The procedure and method of agglomerative hierarchical clustering for determining groups of functional and non-functional requirements for computer systems is proposed. The expediency of using accuracy and completeness metrics for determining the quality of clustering of requirements for computer systems is substantiated, and a comparison metric is also introduced, which makes it possible to determine the number of deviations identified clusters determined using the method of agglomerative hierarchical clustering and actually existing clusters.

A software pipeline for processing functional and non-functional requirements for computer systems was built, which made it possible to practically implement machine learning technologies in the process of their analysis, thus reducing the time spent on project implementation and increasing the quality of its implementation.

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ПІДХОДІВ В ОБЛАСТІ ВИЗНАЧЕННЯ ТА АНАЛІЗУ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ.....	13
1.1. Аналіз основних понять при визначенні вимог до комп'ютерних систем .....	13
1.2. Аналіз процесів інженерії вимог при реалізації комп'ютерних систем .....	15
1.3. Управління процесом розробки вимог до комп'ютерних систем.....	21
1.4. Висновки до розділу .....	25
РОЗДІЛ 2 МЕТОДИ КЛАСТЕРНОГО АНАЛІЗУ ПРИ ВИЗНАЧЕННІ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ .....	26
2.1. Методи та моделі формування кластерів із застосуванням алгоритмів машинного навчання.....	26
2.2. Алгоритм кластеризації K-means .....	28
2.3. Алгоритм ієрархічної кластеризації.....	32
2.4. Машинне навчання при опрацюванні вимог до комп'ютерних систем .....	35
2.5. Метод кластеризації вимог до комп'ютерних систем.....	39
2.6. Обґрунтування метрик оцінювання якості при кластеризації та класифікації вимог до комп'ютерних систем .....	42
2.7. Висновки до розділу .....	45
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ КЛАСТЕРИЗАЦІЇ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ.....	47
3.1. Аналіз та препроцесинг наборів даних для проведення кластеризації вимог .....	47
3.2. Результати векторного представлення вимог до комп'ютерних систем .....	56

3.3. Результати групування вимог до комп'ютерних систем за відомими мітками класу.....	57
3.4. Імплементція алгоритмів кластеризації функціональних вимог до комп'ютерних систем .....	59
3.5. Висновки до розділу .....	67
<b>РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....</b>	<b>68</b>
4.1. Охорона праці.....	68
4.2. Особливості роботи та розлади здоров'я користувачів комп'ютерів, що формується під впливом роботи за ком'ютером.....	71
4.3. Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань .....	75
<b>ВИСНОВКИ.....</b>	<b>79</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>81</b>
Додаток А Тези конференцій .....	84

## ВСТУП

**Актуальність теми.** Зростання структурної і поведінкової складності розробки комп'ютерних систем вимагає залучення сучасних методів і засобів для підвищення ефективності та автоматизації процесів життєвого циклу. Впровадження здобутків і технологій у галузі інтернету речей, опрацювання великих масивів інформації, інтелектуального аналізу даних повинно забезпечувати можливість вирішувати ці проблеми. Однак важливо відмітити і той факт, що основна увага при розробці комп'ютерних систем, приділяється її практичній реалізації та розвитку процесів, які орієнтовані на імплементацію функціональних властивостей системи. Зважаючи на той факт, що процес виявлення, аналізу та фіксації вимог є фундаментом при реалізації будь-якої системи, необхідно проводити його оптимізацію та автоматизацію. Підтвердженням цього факту є те, що у дослідженні журналу СІО задекларовано висновок про те, що 71% проектів розробки комп'ютерних систем зазнають невдач, які пов'язані з поганим управлінням вимогами. Сумна правда полягає ще й в тому, що керівники багатьох із цих проектів, ймовірно, вірили, що вони ефективно забезпечили керування вимогами. З цього випливає, що етапу розробки вимог приділяють менше уваги у порівнянні з іншими процесами життєвого циклу.

Одним з прикладів незадовільного аналізу вимог до комп'ютерних систем є провалена спроба виходу на орбіту зонда NASA Mars Climate Orbiter вартістю 125 млн. доларів. Сумнозвісна помилка виникла через несумісність специфікацій – систему контролю орієнтації було визначено в імперативних одиницях, а навігаційне обладнання та програмне забезпечення використовувало метричні одиниці. У цьому випадку, вимоги були визначені не правильно і використані не у повній мірі.

Під час аналізу звітів про причини невдачі проекту часто можна побачити такі фактори, як «неадекватне розуміння потреб користувачів»,



«нечіткий обсяг» і «зміна вимог», усі з яких є основними аспектами інженерії вимог.

Опосередкованим доказом актуальності досліджень процесів етапу розробки вимог до комп'ютерних систем є те, що кількість звернень у пошуковій системі Google із запитом «управління вимогами» становить майже 350 мільйонів звернень. Незважаючи на те, що в проектах дедалі більше застосовуються гнучкі процеси, управління вимогами залишається ключовим.

Важливим при організації вимог до комп'ютерних систем є визначення їх приналежності до класу функціональних або нефункціональних вимог, що в подальшому впливає на побудову архітектури системи і тестування. Даний процес вимагає залучення аналітиків вимог і від якості виконання їх роботи залежить успішність виконання проекту. Для підвищення ефективності та спрощення процедури класифікації вимог щодо їх приналежності до одного з класів без залучення фахівців необхідно впроваджувати технології інтелектуального кластерного аналізу вимог. Тому актуальною задачею при реалізації проектів комп'ютерних систем є розробка і впровадження методів і засобів кластерного аналізу для підвищення процесу розробки вимог та інших процесів життєвого циклу.

Виходячи з обґрунтованої вище актуальності наряду досліджень, **мета кваліфікаційної роботи** полягає у розробці методів і засобів підвищення ефективності та автоматизації процесів виявлення функціональних та нефункціональних вимог із застосуванням кластерного аналізу.

**Об'єкт дослідження:** процес оптимізації виявлення функціональних та нефункціональних вимог при розробці комп'ютерних систем .

**Предмет дослідження:** методи і засоби виявлення та класифікації вимог до комп'ютерних систем на основі кластерного аналізу.

Для досягнення цілі роботи, поставлено та розв'язано наступні науково-практичні **задачі**:

- дослідження процесів збору, виявлення та аналізу вимог до комп'ютерних систем;
- аналіз основних властивостей і типів вимог до комп'ютерних систем;
- дослідження методів кластерного аналізу вимог, представлених у вигляді текстової інформації;
- обґрунтування математичних методів і моделей визначення кластерів при аналізі вимог до комп'ютерних систем;
- розробка методу оптимізації розробки комп'ютерних систем із застосуванням кластерного аналізу;
- розробка програмної моделі для виявлення кластерів функціональних та нефункціональних вимог до комп'ютерних систем;
- обґрунтування метрик якості кластерного аналізу вимог.

**Методи дослідження:** Для розв'язання задач кваліфікаційної роботи використано наступні методи: аналіз та узагальнення – при аналітичному огляді процесів розробки комп'ютерних систем та особливостей інженерії вимог; математичної статистики і машинного навчання – при розробці методу кластеризації функціональних та нефункціональних вимог до комп'ютерних систем; проектування та програмування – при розробці програмної моделі методу кластеризації вимог; експеримент та вимірювання – при оцінюванні точності і повноти результатів застосування методу кластерного аналізу.

**Наукова новизна отриманих результатів.** Наукова новизна результатів дослідження полягає в наступному:

- уперше запропоновано метод агломеративної ієрархічної кластеризації для визначення груп функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу забезпечити автоматизацію

та підвищити ефективність процесу визначення та аналізу вимог на відповідному етапі життєвого циклу;

– уперше запропоновано процедури і побудовано програмний конвеєр опрацювання функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу практично реалізувати технології машинного навчання в процесі їх аналізу тим самим скоротивши часові витрати на виконання проекту та підвищивши якість його виконання.

**Практичне значення одержаних результатів.** Застосування запропонованого методу і його програмної реалізації при кластеризації функціональних та нефункціональних вимог до комп'ютерних систем дало змогу на основі кластерів вимог спростити процес проектування архітектури систем.

**Публікації.** Результати кваліфікаційної роботи апробовані на XII Міжнародній науково-практичній конференції молодих учених та студентів (6-7 грудня 2023 р.) та XI науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року) як тези конференцій.

1. Тиш Є.В., Шалапай Р.І. Типи вимог до комп'ютерних систем і методи їх виявлення. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 437.

2. Тиш Є.В., Шалапай Р.І. Їєрархічна кластеризація для визначення сукупності функціональних та нефункціональних вимог комп'ютерних систем. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2023. С. 193.

**Структура роботи.** Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 89 арк. формату А4, графічна частина – 8 аркушів формату А1.

## РОЗДІЛ 1

### АНАЛІЗ МЕТОДІВ І ПІДХОДІВ В ОБЛАСТІ ВИЗНАЧЕННЯ ТА АНАЛІЗУ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ

#### 1.1. Аналіз основних понять при визначенні вимог до комп'ютерних систем

Усі проекти починаються з формулювання вимог. Вимоги – це опис того, як має працювати комп'ютерна система. Вимога, як правило, відноситься до деяких аспектів нового або покращеного продукту чи послуги. Широко цитований стандарт IEEE 610.12-1990 [1] визначає вимогу як:

- «Стан або здатність, які необхідні користувачеві для вирішення проблеми або досягнення мети.
- «Умова або здатність, якій повинна відповідати або володіти система чи компонент системи, щоб задовольнити контракт, стандарт, специфікацію чи інші офіційно встановлені документи».
- «Задокументоване представлення стану або можливості, як попередніх означеннях».

Таким чином, вимоги включають не лише потреби користувачів, а й потреби, що впливають із загальних організаційних, державних і галузевих стандартів. Очевидно, що вимога — це набір потреб, що висувають користувачі та інших зацікавлені сторони (загальна організація, спільнота, державні органи та галузеві стандарти), які мають бути виконані.

В ідеалі вимоги не залежать від дизайну, показу «що» повинна робити система, а не «як» це має бути зроблено. Однак це не завжди можливо на практиці, тобто значення «що» і «як» відрізняються [2]. Вимоги можна класифікувати багатьма способами, наприклад, як показано в табл. 1.1.

## Приклад типів вимог

№	Тип вимог	Опис
1.	Функціональні	Дає відповідь на питання, що буде виконувати система
2.	Нефункціональні вимоги	Обмеження щодо типів рішень, які відповідатимуть функціональним вимогам, наприклад, точність, продуктивність, безпека та можливість модифікації
3.	Вимоги до рівня цілей	Вимоги, які безпосередньо пов'язані з бізнес-цілями
4.	Вимоги до рівня домену	Вимоги, які стосуються предметної області
5.	Вимоги до рівня продукту	Вимоги, які висуваються до системи, як до продукту
6.	Вимоги до рівня проектування	Визначають, що буде спроектовано
7.	Основні вимоги	Вимоги, одержані від зацікавлених сторін
8.	Похідні вимоги	Вимоги, які наслідуються від основних вимог
9.	Бізнес-вимоги на протипагу технічним вимогам	Визначається реальність імплементації бізнес-вимог технологічною спроможністю
10.	Вимоги до продукту на протипагу вимогам до процесу реалізації	Бізнес-потреби щодо того, як люди взаємодіятимуть із системою
11.	Вимоги відносно ролей користувачів	Вимоги клієнта, вимоги користувачів, ІТ-вимоги та вимоги безпеки.

Хоча в літературі проводиться розмежування між різними типами вимог, на практиці не завжди легко визначити такі відмінності [3].

Наприклад, вимога користувача, пов'язана з безпекою може бути класифікована як нефункціональна вимога. Однак під час реалізації можуть розвиватися інші вимоги, які відрізняються функціональністю, наприклад авторизація користувача [4]. Розуміючи основи того, що представляє собою вимога, наступним кроком є організація процесу, який використовується для управління та розробки вимог.

## 1.2. Аналіз процесів інженерії вимог при реалізації комп'ютерних систем

Розробка вимог має відношення до усіх процесів життєвого циклу, які оперують вимогами. Це в основному включає збір, документування та керування вимогами. Із зростаючим усвідомленням важливості вимог у процесі розробки комп'ютерних систем, інженерія вимог все більше стає сферою, якій приділяють увагу.

Загальними видами діяльності з розробки вимог є виявлення, інтерпретація та структурування (аналіз та документація), узгодження, верифікація та перевірка, управління змінами та відстеження вимог.

Існує кілька доступних моделей процесів для опису процесу розробки вимог. Процес розробки вимог часто зображується різними способами, включаючи лінійну модель, інкрементну, нелінійну модель і спіральні моделі. У [5] запропоновано концептуальну лінійну модель процесу розробки вимог, яка вказує на ітерації між видами діяльності. З іншого боку у [6], недоліком вважають суто лінійну модель, яка не вказує на накладення або повторення діяльності. Хоча деякі дослідники схильні зображати процес розробки вимог як лінійну модель, нелінійні моделі також були використані Loucoroulos і Karakostas [6] і зображують процес розробки вимог як ітеративний і циклічний за своєю природою.

Альтернативно, спіральна модель представляє послідовність дій, що виконуються в ітераціях, які призводять до поступового прогресування

моделі розробки вимог [5]. Однак вона має наслідки для моделі процесу розробки вимог. Спіральний підхід вимагатиме обробки вимог у кожному раунді спіральної моделі, яка подібна до ідей, представлених у [5]. Вони забезпечують другу модель процесу розробки вимог, яка зображує ті самі дії з розробки вимог, що й у їхній лінійній моделі, що відбувається лише у спіральному представленні.

Дії з лінійної моделі процесу повторюються в ітераціях, утворюючи спіраль. Наприкінці кожної ітерації приймається рішення щодо прийняття документа вимог або виконання подальшої ітерацію. Результати досліджень процесів розробки вимог на практиці вказали на те, що представлені в літературі систематичні та поетапні моделі розробки вимог не обов'язково відображають процеси її формування на практиці. Мартін та інші вчені, які досліджували процес розробки вимог у кожному конкретному випадку виявили, що зазвичай проекти оброблялися за лінійною моделлю з деякою ітерацією дій. Більшість проектів, які вони досліджували, йшли загалом лінійним процесом до фази прототипування, яка потім призвела до ітераційного процесу.

Розробка вимог представляє собою процес, за допомогою якого визначається мета реалізації комп'ютерної системи. Зацікавлені сторони та їхні потреби, а також системні обмеження та їх взаємодія ідентифікуються та документуються таким чином, щоб їх можна було використовувати для подальшого аналізу, використання та впровадження. Для досягнення цієї мети потрібно виявити, зрозуміти, сформулювати, проаналізувати та дійти згоди щодо того, в чому полягає проблема, чому цю проблему потрібно вирішити та хто має відповідати за вирішення цієї проблеми. Процес формування вимог, згідно [1], є спіральним процесом і включає чотири основні фази, а саме:

- дослідження домену для його розуміння та виявлення вимог,
- оцінювання та узгодження вимог щодо можливості їх практичної реалізації,



- специфікація і документація вимог
- забезпечення якості вимог.

Активності та відповідні дії процесу формування вимог показані на рис. 1.1.



Рис. 1.1. Структура процесів при формуванні вимог до комп'ютерних систем

Сукупність процесів і результатів їхнього виконання щодо формування вимог, які показані на рис. 2.1, є доволі загальними і потребують більшої деталізації в залежності від типу проекту комп'ютерної системи та процесів, які характерні для конкретної команди розробників. Окрім цього, характерною особливістю процесів виявлення вимог до КС є значна частка ручної роботи, яка складно піддається автоматизації. Це стосується виявлення і точної ідентифікації й поділу на функціональні та нефункціональні вимоги, управління змінами у вимогах, контроль і моніторинг виконання вимог на подальших етапах розвитку проекту.

Схематично процеси опрацювання функціональних та нефункціональних вимог представлено на рис. 1.2.

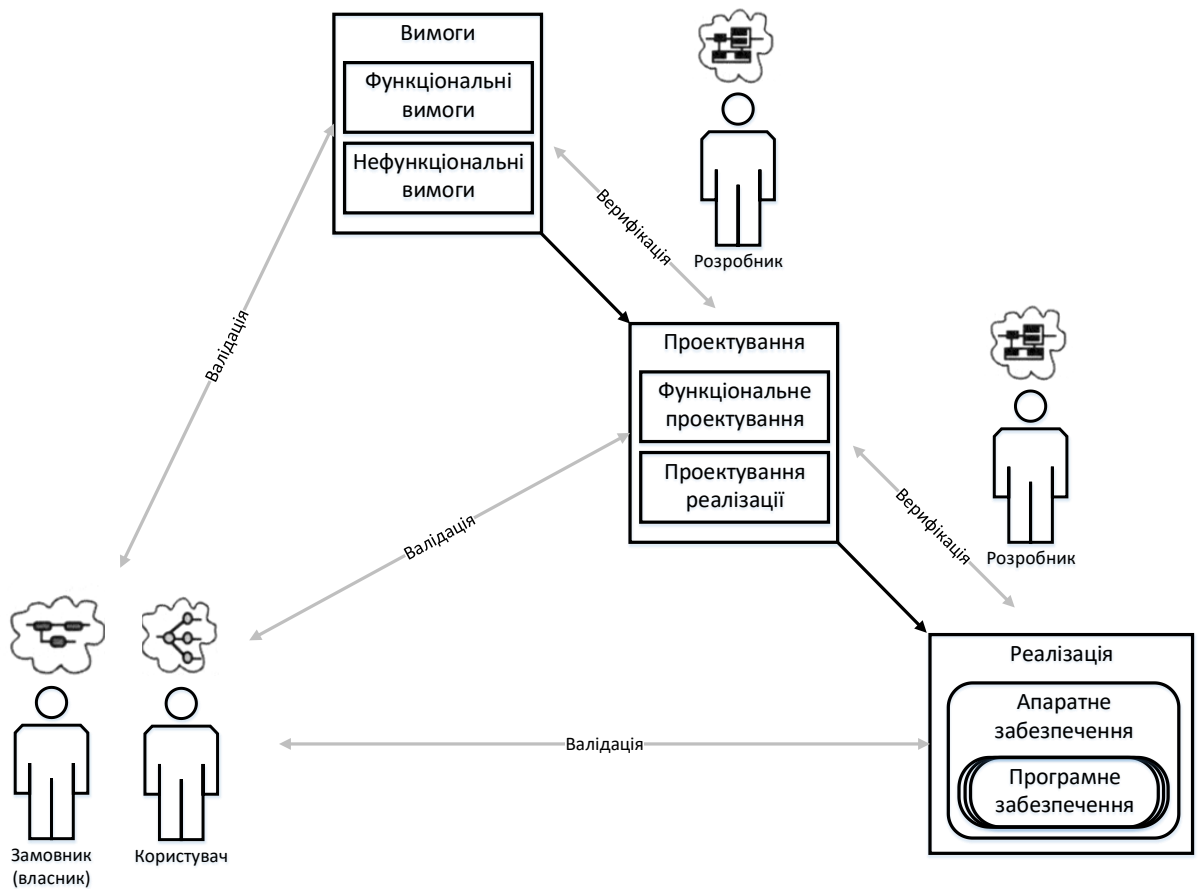


Рис. 1.2. Процеси опрацювання функціональних та нефункціональних вимог до комп'ютерних систем

На рис. 1.2 показано, що маючи функціональні та нефункціональні вимоги, або ще як їх називають вимоги якості, процес реалізації комп'ютерної системи передбачає відображення їх на етап побудови архітектурних рішень та подальшої практичної імплементації. При цьому процеси верифікації покладаються на розробників КС, а валідація вимог та їх реалізації у вигляді властивостей КС виконується із залученням замовників та користувачів системи. Однак, у даній схемі не наведено принципів визначення щодо приналежності вимог до однієї з груп: функціональних чи нефункціональних. Це ще раз підтверджує актуальність

задач щодо автоматизації процесів виявлення та бінарної класифікації чи кластеризації.

Для більш глибокого розуміння, якими властивостями і характеристиками володіють функціональні та нефункціональні вимоги необхідно дослідити принципи їх класифікації і фактори, які на це впливають.

На рис. 1.3 проілюстровано фактори, які мають визначальний вплив на встановлення приналежності вимог до класу функціональних чи нефункціональних вимог

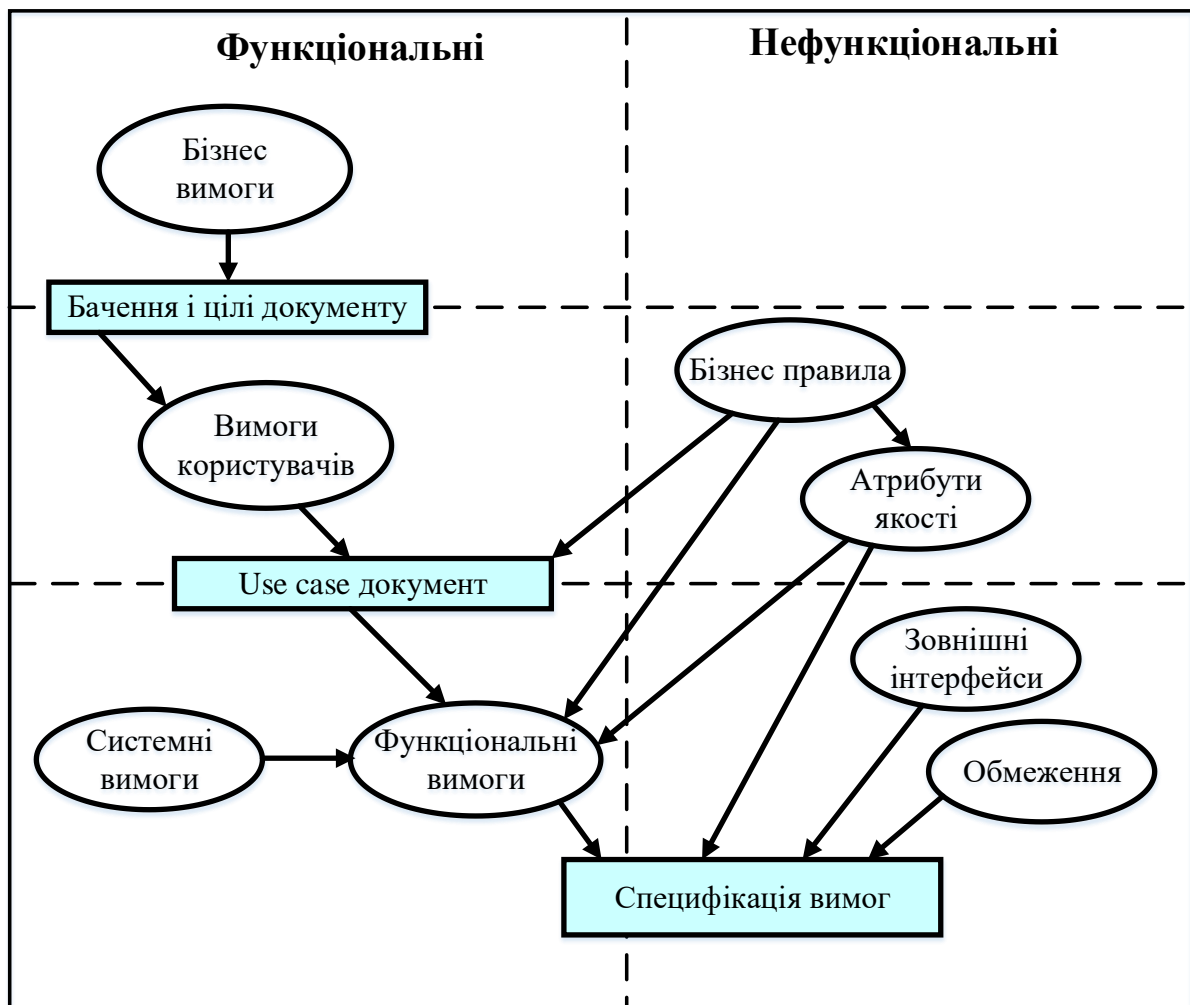


Рис. 1.3. Фактори, які визначають і впливають на встановлення приналежності вимог до певного класу

На основі аналізу бізнес вимог можна встановити частину необхідних функціональних вимог для користувачів, проте все ж вони висловлюють і свої власні потреби. Бізнес вимоги та вимоги користувачів формують базис функціональності, який фіксується у потенційних сценаріях імплементації комп'ютерної системи. Нефункціональні вимоги також впливають на сценарії виконання функцій комп'ютерною системою, зокрема через наявні у бізнес правила, які впливають також на функціональні вимоги та атрибути якості системи. Серед системних вимог, виділяють частину, яка має безпосереднє відношення до виконання функцій системою. Нефункціональні вимоги можуть містити обмеження та особливості взаємодії із зовнішніми програмно-апаратними комплексами через відповідні інтерфейси.

Як видно з рис. 1.3, кластер функціональних вимог визначають в першу чергу бізнес вимоги, які формують бачення і цілі реалізації комп'ютерної системи.

Розробку вимог часто тривіалізують як діяльність, яку виконують аналітики з добрих намірів, перш ніж вони почнуть виконувати справжню роботу зі специфікації продукту. В цілому процеси інженерії вимог можна представити, як проілюстровано на рис. 1.4.



Рис. 1.4. Процеси інженерії вимог при створенні комп'ютерних систем

Іноді корисно розглядати вимоги як вирішення проблеми, а не як рішення, що підкреслює, що вимога не є аспектом дизайну або пропозицією

щодо того, як реалізувати рішення. Вимога має містити те, що ми хочемо вирішити, а не як Звичайно, хоча це визначення слід сприймати серйозно, це не жорстке і швидке правило; іноді корисно знати, яке рішення передбачає користувач як частину своїх вимог, а міркування дизайну часто створюють нові або переглянуті вимоги. Крім того, проекти часто мають обмеження, які диктують аспекти рішення, такі як забезпечення відповідності стандартам або обмеження використання певних технологій. Але вказуючи обмеження, ми просто ставимо огорожі навколо рішення, ми не надаємо саме рішення.

### 1.3. Управління процесом розробки вимог до комп'ютерних систем

Ефективне управління процесом розробки комп'ютерних систем сприяє стійкій конкурентній перевазі компаній, що займаються інформаційними технологіями. Це означає, що менеджери повинні враховувати вимоги клієнтів і бізнес-вимоги, а також технологічні можливості, які можуть бути різними або збігатися. Важливо дотримуватися бюджету, скорочення часу життєвого циклу та досягнення цільових показників продуктивності продукту, щоб переконатися, що вимоги до системи узгоджуються з бізнес-цілями. Ці проблеми не є унікальними для розробки комп'ютерних систем, а насправді типові для складних системних продуктів. В епоху Інтернету відбулися значні зміни в бізнес-середовищі, створивши складніші вимоги до технологій, які підтримують бізнес-інформаційні системи.

Відповідно, розуміння, аналіз, моделювання та управління вимогами стали однаково складними. Щоб реалізувати високоякісні комп'ютерні системи вчасно та в рамках бюджету, важливо мати належним чином структуровані та контрольовані специфікації вимог, які є зрозумілими, вичерпними та послідовними. Рівні вимог, які часто використовуються на практиці представлені у вигляді табл. 1.2.

Таблиця 1.2

## Класифікація вимог за трьома рівнями

	Стратегічне управління	Тактичне управління	Операційне управління
Вимоги організаційного рівня	<ul style="list-style-type: none"> <li>– бізнес стратегія;</li> <li>– конкурентоспроможність;</li> <li>– технологія;</li> <li>– маркетинг;</li> <li>– економічна цінність товару;</li> </ul>	<ul style="list-style-type: none"> <li>– заплановані переваги товару</li> </ul>	<ul style="list-style-type: none"> <li>– технологічний поштовх на ринку</li> </ul>
Вимоги на рівні продукту	<ul style="list-style-type: none"> <li>– вимоги до системи у конкретній версії;</li> <li>– вимоги до архітектури системи</li> </ul>	<ul style="list-style-type: none"> <li>– управління ресурсами;</li> <li>– реалізація конкретної версії.</li> </ul>	<ul style="list-style-type: none"> <li>– управління змінами;</li> <li>– внесення син-таксичних або семантичних змін;</li> <li>– змінюваність вимог.</li> </ul>
Вимоги на рівні проекту	<ul style="list-style-type: none"> <li>– планування проекту;</li> <li>– техніко-економічне обґрунтування</li> </ul>	<ul style="list-style-type: none"> <li>– управління проектами</li> </ul>	<ul style="list-style-type: none"> <li>– перевірка того, які вимоги перейдуть до наступного випуску</li> </ul>

Процес розробки вимог є одним із головних чинників, які впливають на успіх проектів з реалізації комп'ютерних систем. Це особливо підтверджується на глобальному конкурентному ринку, де терміни виходу

на ринок і задоволення вимог зацікавлених сторін є ключовими факторами успіху.

Таким чином, удосконалення процесу розробки вимог може значно збільшити ймовірність успіху комп'ютерної системи.

Сучасні підходи до розробки комп'ютерних систем часто змішують бізнес проблеми з впровадженням ІТ для формування монолітних систем, які не реагують на зміни краще, ніж їхні попередники. Отже, ІТ-системи в цій галузі повинні бути динамічними та швидко адаптуватися до свого середовища.

Поточна розширена перспектива ІТ-продуктів у бізнесі має різні наслідки для управління процесами розробки комп'ютерних систем, тобто вимоги до них не повинні розглядатися виключно в проектах її частин. Базуючись на трирівневій моделі прийняття управлінських рішень, а саме стратегічних, тактичних і оперативних, у [2] проілюстровано, як проводити аналіз процесу розробки вимог і процесів прийняття рішень, що лежать в його основі, використовуючи класичні рамки прийняття рішень. У роботі варто дотримуватися подібної точки зору, тобто управління вимогами до комп'ютерних систем залежить від діяльності, орієнтованої на організацію, продукт і процес, і ними потрібно керувати на стратегічному, тактичному та операційному рівнях.

Табл. 1.2. ілюструє класифікацію вимог до КС у матрицях 3\*3, де кожна комірка містить декілька прикладів дій або рішень щодо вимог. Ці три рівні можна коротко описати так.

Вимоги на рівні організації: команда вищого керівництва організації може мати стратегічні та довгострокові цілі щодо частки ринку. Цілі та стратегії на рівні організації неминуче впливатимуть на те, які продукти організація повинна розробляти. Таким чином, вимоги, що висуваються до продуктів, спочатку повинні бути оцінені на рівні організації, щоб переконатися, що вимоги узгоджені з цілями та стратегіями організації. Однією з головних проблем, з якими стикаються успішні розробки

комп'ютерних систем, є визначення того, як кінцевий продукт підтримуватиме бізнес-цілі.

Вимоги на рівні продукту. Вимоги до комп'ютерних систем повинні бути узгоджені з бізнес-цілями організації, що їх розробляє. Одне з важливих питань полягає в тому, як збалансувати занепокоєння клієнтів і розробників. Методи моделювання цілей у розробці вимог служать механізмом, за допомогою якого можна зв'язати вимоги зі стратегічними цілями, закріпленими в контексті загальної моделі бізнес-стратегії. Вимоги зазвичай є як функціональними, так і нефункціональними. Керівництво продуктом має гарантувати, що вимоги відповідають цілям і завданням продукту. Це може означати вибір вимог до продукту, які найкраще відповідають загальним цілям і стратегіям організації.

Вимоги на рівні проекту. Вимоги на рівні системи, як продукту, мають бути запаковані в частини, які входять до конкретних проектів або випусків комп'ютерної системи. Важливо, щоб вимоги були пріоритезовані та вибрані на основі їх виконання як продукту, так і організаційних цілей і стратегій.

Вимоги можуть бути обрані для реалізації на основі того, чи відповідають вони потребам конкретного та важливого клієнта, чи вони потенційно відкривають новий сегмент ринку для організації. Ці вимоги визначають умови, за яких буде виконуватися проект, включаючи питання, пов'язані з плануванням проекту, управлінням ризиками, бюджетом і вартістю.

Зростання стратегічної важливості ІТ означає, що інструменти, методи та процеси повинні бути інтегровані з вимогами КС, щоб вони відповідали стратегічним цілям бізнесу та бізнес-моделі організацій, які вони підтримують. Зміна бізнесу є частиною розвитку системи. У міру того, як системи стають більш інтегрованими та залучають більше користувачів з різним походженням, розробники комп'ютерних систем змушені розуміти



наслідки своїх рішень щодо аналізу витрат і вигод, особливо на ранніх етапах життєвого циклу.

У літературі з системної інженерії та менеджменту, зокрема в літературі з управління ризиками, наголошується на важливості зусиль з планування проекту, планування графіків, планування витрат та оцінки ризиків у розробці продукту як найважливіших при його створенні, які відповідають вимогам клієнтів і відповідають стратегічним бізнес-цілям.

#### 1.4. Висновки до розділу

До основних результатів, отриманих у даному розділі, належать:

1. При дослідженні основних понять у сфері інженерії вимог визначено основні їх типи і класифікацію, які широко використовуються на практиці при реалізації комп'ютерних систем і встановлено, що актуальним завданням сучасного розвитку комп'ютерної інженерії є автоматизація та оптимізація процесів класифікації функціональних та нефункціональних вимог.

2. Проведено аналіз процесів і моделей інженерії вимог у результаті якого встановлено, що найбільш важливими з них є виявлення та управління вимогами, а для забезпечення ефективності їх виконання необхідно зменшити суб'єктивний вплив при визначенні груп функціональних та нефункціональних вимог.

3. Проаналізовано підходи щодо управління вимогами до комп'ютерних систем на етапах їх життєвого циклу і підтверджено важливу роль відповідних процесів, дій та активностей, оскільки вимоги є фундаментом для подальших стадій розробки комп'ютерних систем.

## РОЗДІЛ 2

### МЕТОДИ КЛАСТЕРНОГО АНАЛІЗУ ПРИ ВИЗНАЧЕННІ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ

#### 2.1. Методи та моделі формування кластерів із застосуванням алгоритмів машинного навчання

Перш, ніж перейти до застосування методів кластеризації для групування функціональних та нефункціональних вимог, необхідно проаналізувати означення процесу кластеризації. Загалом, кластеризація це задача поділу не розмічених даних або точок даних на різні кластери таким чином, щоб подібні точки даних потрапляли в той самий кластер. Простими словами, мета процесу кластеризації полягає в тому, щоб відокремити групи зі схожими ознаками та розподілити їх у кластери.

Для більш практичного розуміння, можна привести приклад з реального життя. Припустимо, що керівник організації, яка займається задачею в оренду нерухомості, хоче зрозуміти пріоритети своїх клієнтів для розширення бізнесу. На практиці в ручному режимі провести аналіз даних щодо кожного клієнта та розробити унікальну бізнес-стратегію для кожного з них неможливо. Однак, можна об'єднати всіх своїх клієнтів у, скажімо, 10 груп на основі їхніх купівельних звичок і використовувати окрему стратегію для клієнтів у кожній із цих 10 груп. Це і є кластеризацією.

Загалом розрізняють кластеризацію двох типів:

- жорстка кластеризація (hard);
- м'яка кластеризація (soft).

У випадку hard кластеризації кожен елемент вхідних даних або повністю належить до кластеру, або ні. При виявленні функціональних та нефункціональних вимог до комп'ютерних систем передбачається два основних кластери, хоча в кожному з них, можуть існувати підкластери. У

наведеному вище прикладі з реального життя, кожен клієнт однозначно входить до однієї з 10 груп.

Soft кластеризація передбачає те, що замість розміщення кожного елемента вхідних даних лише в одному кластері розраховується ймовірність того, що цей елемент з певною ймовірністю належатиме до кількох кластерів. Наприклад, із наведеного вище сценарію, для кожного клієнта визначається ймовірність приналежності до кожного з 10 кластерів.

Оскільки завдання кластеризації є доволі не тривіальним, а мітки кластерів чи класів відсутні, то існує багато методологій щодо її реалізації. Кожна методологія дотримується різного набору правил для визначення «подібності» між елементами даних. Насправді відомо понад 100 алгоритмів кластеризації. Але деякі з алгоритмів використовуються більш широко.

Моделі зв'язності представляють собою моделі, які базуються на уявленні про те, що елементи даних, розташовані ближче одні до одних в просторі даних, мають більшу схожість, ніж елементи даних, розташовані на більшій відстані. У цих моделях можна застосовувати два підходи.

У першому підході вони починають з групування всіх елементів даних в окремі кластери, а потім агрегування їх у міру зменшення відстані.

У другому підході всі елементи даних групуються в один кластер, а потім розділяються зі зростанням відстані. Крім того, вибір функції відстані є суб'єктивним. Ці моделі дуже легко інтерпретувати, але їм не вистачає масштабованості для обробки великих наборів даних. Прикладами таких моделей є ієрархічні алгоритми кластеризації та їх варіанти.

Центроїдні моделі представляють собою алгоритми з ітераціями, у яких подібність елементів даних обчислюється на основі відстані елемента до центроїда або центру кластерів. Алгоритм кластеризації k-Means є популярним алгоритмом, який потрапляє в цю категорію. У цих моделях немає параметрів кластера, тому важливо мати попередні знання про набір даних. Ці моделі виконуються ітеративно, щоб знайти локальний оптимум.

Моделі розподілу – це такі моделі кластеризації, які базуються на уявленні про те, наскільки ймовірно, що всі точки даних у кластері належать до одного розподілу (наприклад, нормального, Гаусового). Ці моделі часто страждають від перенавчання. Популярним прикладом цих моделей є алгоритм максимізації очікування (EM), який використовує багатовимірний нормальний розподіл.

Моделі щільності реалізують пошук у просторі даних в області з різною щільністю елементів даних. Вони виділяють різні області щільності елементів та відносять їх до одного кластеру. Популярними прикладами моделей щільності є DBSCAN і OPTICS. Ці моделі особливо корисні для ідентифікації кластерів довільної форми та виявлення викидів, оскільки вони можуть виявляти та відокремлювати елементи даних, розташовані в розріджених областях простору даних, а також елементи, які належать до щільних областей.

## 2.2. Алгоритм кластеризації K-means

K-means — це ітеративний алгоритм кластеризації, який спрямований на пошук локальних максимумів у кожній ітерації. Цей алгоритм складається з 5 кроків.

Перший крок полягає у встановленні бажаної кількості кластерів, які необхідні для вирішення задачі. При розробці методу кластеризації функціональних та нефункціональних вимог до комп'ютерних систем кількість кластерів становить два, тобто  $k = 2$ . Для наочності і простоти представлення елементів функціональних та нефункціональних вимог припустимо, що їх загальна кількість становить 5. У декартовій площині зобразимо елементи, які інтерпретують вимоги у вигляді п'яти точок, як показано на рис. 2.1.

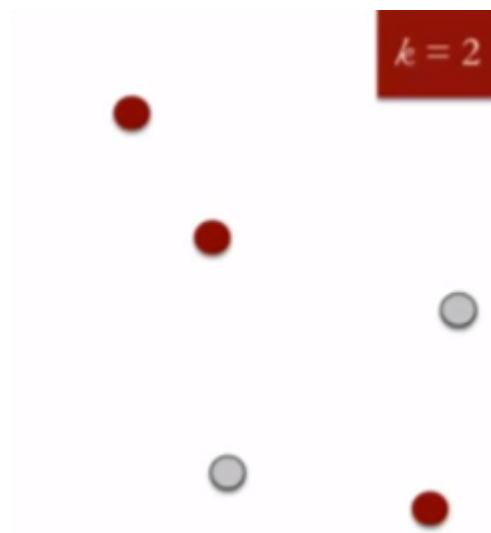


Рис. 2.1. Візуалізація вимог у декартовому просторі

Другий крок полягає у випадковому визначенні приналежності кожного елемента даних до конкретного кластера. У даному випадку червоним кольором позначено елементи даних, які відносяться до кластера функціональних вимог, а сірим – до нефункціональних.

Третій крок передбачає обчислення центра кластера або по-іншому центроїда, який для елементів, позначених червоним кольором позначено відповідно червоним хрестиком, а для сірих – сірим хрестиком (рис. 2.2).



Рис. 2.2. Обчислення центрів кластера

На четвертому кроці повторно призначають кожному елементу даних найближчий центроїд кластера: потрібно зауважити, що нижня точка даних належить до червоного кластера, навіть якщо вона ближча до центроїда сірого кластера. Таким чином, встановлюється приналежність даних до сірого кластера (рис. 2.3.).



Рис. 2.3. Встановлення приналежності даних до кластера на основі відстані до його центру

П'ятий крок передбачає повторне обчислення центрів кластерів, тобто знаходження центроїдів для кластерів функціональних та нефункціональних вимог. Результат повторного обчислення центрів кластера показано на рис. 2.4.

Повторне виконання четвертого і п'ятого кроків алгоритму відбуваються до тих пір, поки спостерігається покращення групування даних. Це означає, що поки не буде забезпечено досягнення глобального оптимуму, тобто коли більше не буде «перемикання» елементів даних між двома кластерами при двох послідовних повторях. Це позначатиме завершення алгоритму, якщо це не зазначено явно.

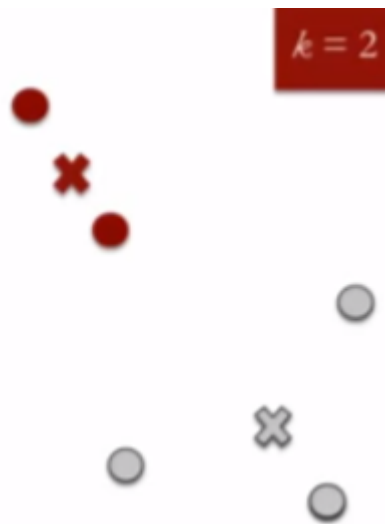


Рис. 2.4. Повторне обчислення центрів кластерів

Таким чином, K-means представляє собою найпопулярніший і найпростіший алгоритм кластеризації, головною метою якого є пошук кластерів для заданих даних. Метод кластеризації K-Means є методом кластеризації на основі центроїда, де «K» означає – кількість кластерів на які потрібно поділити елементи даних. Для того, щоб визначити як обрати «K», потрібно володіти знаннями домену. Математично цільову функцію при застосуванні методу K-means можна описати наступним чином:

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2, \quad (2.1)$$

де  $J$  – цільова функція;

$k$  – кількість кластерів;

$n$  – кількість випадків встановлення приналежності;

$c_j$  – центр кластера  $j$ ;

$x_i^{(j)}$  –  $i$ -ий випадок приналежності елемента  $x$  до кластера  $j$ .

Вираз  $\left\| x_i^{(j)} - c_j \right\|^2$  інтерпретує функцію відстані до центра кластера.

### 2.3. Алгоритм ієрархічної кластеризації

Ієрархічна кластеризація, як випливає з назви – це алгоритм, який формує ієрархію кластерів. Цей алгоритм починається з усіх елементів даних, які розглядаються як окремі кластери. Після цього два найближчі кластери об'єднуються в один. Зрештою, цей алгоритм завершується, коли залишається лише один кластер. Результати ієрархічної кластеризації можна показати за допомогою дендрограми. Дендрограму можна інтерпретувати як показано на рис. 2.5.

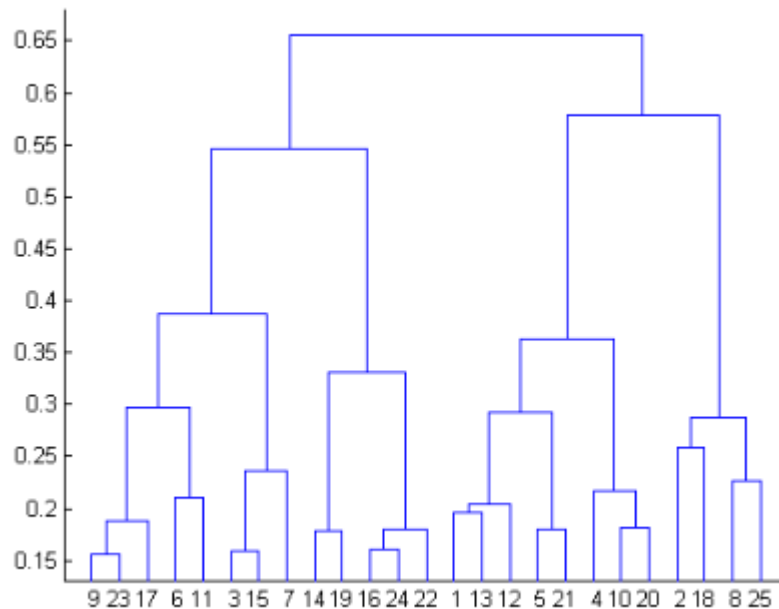


Рис. 2.5. Представлення ієрархічної кластеризації у вигляді дендрограми

На рис. 2.6. показано вхідні елементи даних у вигляді 25 точок, кожна з яких представляє окремий кластер. Потім два найближчі кластери об'єднуються, доки не залишиться лише один кластер у верхній частині.

Висота на дендрограмі, на якій зливаються два кластери, представляє відстань між двома кластерами в просторі даних. Рішення щодо кількості кластерів, які найкраще відображають різні групи, можна вибрати, аналізуючи дендрограму. Найкращий вибір кількості кластерів



визначається кількістю вертикальних ліній у дендрограмі, розрізаних горизонтальною лінією, яка може перетинати максимальну відстань по вертикалі, не перетинаючи кластер.

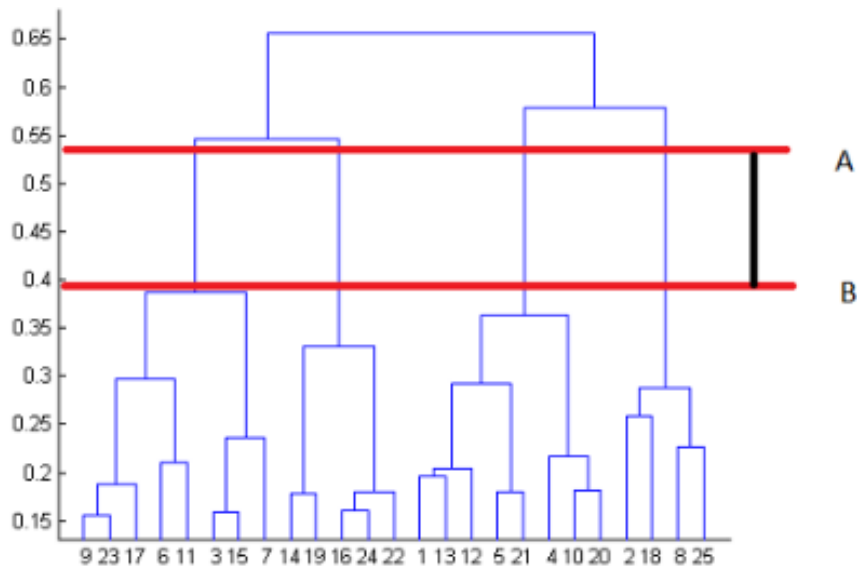


Рис. 2.6. Формування кластерів на основі відстані між ними

У наведеному на рис. 2.6 прикладі, найкращий вибір кількості кластерів буде 4, оскільки червона горизонтальна лінія на дендрограмі нижче покриває максимальну вертикальну відстань АВ.

Дві важливі речі, які потрібно знати про ієрархічну кластеризацію. Цей алгоритм реалізовано з використанням підходу знизу вгору. Також можна застосувати низхідний підхід, починаючи з того, що усі точки даних, задають один кластер. Після цього реалізується рекурсивне розбиття до тих пір, поки кожній точці даних не буде призначено окремий кластер.

Рішення про об'єднання двох кластерів приймається на основі близькості цих кластерів. Існує кілька критеріїв для визначення близькості двох кластерів.

Для обчислення відстані між кластерами може використовуватися Евклідова метрика, яка описується формулою:

$$\|a - b\| = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (2.2)$$

де  $a$  і  $b$  – кластери;

$a_i$  і  $b_i$  – елементи відповідно кластера  $a$  і  $b$ ;

$n$  – кількість елементів кластера.

Метрика квадрату Евклідової відстані обчислюється за формулою:

$$\|a - b\| = \sum_{i=1}^n (a_i - b_i)^2 \quad (2.3)$$

При розрахунку відстані між кластерами може застосовуватися Манхеттенська відстань, яка обчислюється за формулою:

$$\|a - b\|_1 = \sum |a_i - b_i| \quad (2.4)$$

Також ефективним, в залежності від виміру просторів даних, можуть бути метрики максимуму відстані (метрика Чебишева) та відстань Махаланобіса. Формула для обчислення максимум відстані приведена у формулі (4.5).

$$\|a - b\|_{Infinity} = \max_i |a_i - b_i| \quad (2.5)$$

Ієрархічна кластеризація не може добре опрацьовувати великі дані, але k-means може. Це пояснюється тим, що часова складність k-means є лінійною, тобто  $O(n)$ , тоді як складність ієрархічного є квадратичною, тобто  $O(n^2)$ . Оскільки опрацювання даних починається з випадкового вибору кластерів, результати, отримані багаторазовим виконанням алгоритму, можуть відрізнятися в кластеризації k-means.

Під час ієрархічної кластеризації результати можна відтворити. Виявлено, що k-means добре працює, коли форма кластерів є

гіперсферичною (як коло у 2D або сфера у 3D). «K» означає, що кластеризація вимагає попереднього знання кількості кластерів, на які потрібно розділити дані. Однак при ієрархічній кластеризації можна зупинитися на будь-якій кількості кластерів, інтерпретуючи дендрограму.

#### 2.4. Машинне навчання при опрацюванні вимог до комп'ютерних систем

Машинне навчання – це наука програмування комп'ютерів, метою якої є навчання на основі даних. По суті, машинне навчання передбачає створення математичних моделей, які допомагають зрозуміти дані [5].

Наразі машинне навчання використовується в кількох контекстах, таких як:

- аналіз зображень продуктів на виробничій лінії для їх автоматичної класифікації;
- виявлення пухлин при скануванні мозку;
- автоматична класифікація статей та новин;
- виявлення шахрайства;
- створення чат-ботів чи особистих помічників та ін.

Сьогодні інтенсивно досліджується можливість впровадження машинного навчання (ML) у процеси розробки комп'ютерних систем та їх компонентів, зокрема, для управління процесами розробки, так і для генерації програмного коду системи [3].

Для проведення цих досліджень сховища даних, що походять із процесу розробки комп'ютерних систем, як-от обговорення на форумі, історія обслуговування, відгуки користувачів і коментарі користувачів у соціальних мережах, стали багатим джерелом даних для використання в ML у поєднанні з аналізом тексту для добування корисної інформації, яка може бути використана в майбутньому.

На сьогодні уже проведено велику кількість досліджень, пов'язаних із комп'ютерною і програмною інженерією в симбіозі з машинним навчанням. Автори виявили, що ML впливає на діяльність із виявлення, аналізу, перевірки та управління вимогами.

Для досягнення мети кваліфікаційної роботи щодо кластеризації функціональних та нефункціональних вимог пропонується використовувати техніки нормалізації тексту, методи добування ознак і алгоритми машинного навчання для створення моделей кластеризації і класифікації вимог до комп'ютерних систем з використанням бази даних PROMISE\_exp та деяких інших.

Нормалізація тексту визначається як процес, який складається з серії кроків, які слід виконати, щоб очистити і стандартизувати текстові дані у форму, яка може використовуватися іншими системами опрацювання природної мови (NLP) і аналітичними системами та програмами як вхідні дані [9].

Одним із цих кроків є токенізація, яка складається з поділу тексту на набори токенів, і ці лексеми можуть бути реченнями або окремими словами, залежно від цілей дослідника. Часто сама токенізація також є частиною нормалізації тексту. Окрім токенізації, частинами процесу нормалізації є кілька інших методів, таких як:

- перетворення регістру;
- виправлення орфографії;
- видалення нерелевантних слів та інших непотрібних термінів, таких як артиклі та займенники, корені й лематизація.

У кваліфікаційній роботі база даних, що містить вимоги до комп'ютерних систем, пройшла процес нормалізації, під час якого документи в корпусі були токенізовані, усі тексти були перетворені на нижній регістр, нерелевантні слова були видалені, а сполучені слова в дієслівних часах були перетворені на їхні вихідні форми з використанням лематизації.

Алгоритми машинного навчання працюють з числовим простором ознак, очікуючи вхідних даних як двовимірний масив, де рядки є екземплярами, а стовпці є функціями [10].

Для застосування машинного навчання до тексту, потрібно перетворити екземпляри вимог, які представляються окремими документами, у векторний вигляд, щоб можна було використовувати числові значення.

Процес кодування документів у числовому просторі ознак називають добуванням ознак або, простіше кажучи, векторизацією, і є важливим першим кроком до аналізу з урахуванням специфіки мови [12].

Модель векторного простору (Vector Space Model (VSP)) – це концепція та модель, яка дуже корисна при опрацюванні текстових даних, зокрема, у пошуку інформації та ранжуванні документів.

Модель векторного простору представляє собою алгебраїчну та математичну модель для представлення текстів у вигляді числових векторів. Деякі з цих моделей, які використовуються в дослідженнях, пов'язаних з аналізом тексту, це: Bag of Words («сумка слів»), TF-IDF та Averaged Word Vectors [14].

У кваліфікаційній роботі буде використовуватися підхід «сумки слів», TF-IDF і  $\chi^2$ . Ці три методи є доволі простими у використанні та ефективно функціонують на невеликих наборах даних, а контекст залежить від домену.

«Сумка слів» є, мабуть, одним із найпростіших, але найпотужніших методів добування функцій із текстових документів [2]. Суть цієї моделі полягає у перетворенні текстових документів на вектори таким чином, щоб кожен з них перетворювався на вектор, який представляє частоту всіх різних слів, які присутні у векторному просторі документа для цього конкретного випадку [14].

Підсумовуючи, використовуючи BoW, вимога « $j$ » виражається вектором  $X_j = (x_{1,j} \dots x_{i,j} \dots x_{n,j})$ , в якому  $x_{i,j}$  позначає обчислену вагу ознаки « $i$ » частотою терміна « $i$ » у вимозі « $j$ », а « $n$ » позначає кількість

термінів у словнику. Після цього класифіковані вручну вимоги, які виражаються векторами, діють як вхідні дані контрольованих алгоритмів машинного навчання і, отже, використовуються для навчання класифікаторів [15].

Періодичність термінів — зворотна частота документів (TF-IDF) TF-IDF — це модель, яка поєднує два показники:

- необроблене значення частоти терміна в конкретному документі;
- обернена частота документа для кожного терміну, яка обчислюється діленням загальної кількості документів у корпусі за частотою використання документів для кожного терміну та застосування логарифмічного масштабування до результату.

Обернену частоту документа можна математично представити такою формулою:

$$IDF_i = \frac{T_{Req}}{T_{Req\_with\_term\_i}} \quad (2.6)$$

де  $T_{Req}$  – загальна кількість вимог до комп'ютерної системи;

$T_{Req\_with\_term\_i}$  – кількість вимог з терміном  $i$ .

Поєднуючи дві метрики, вектор ознак TF-IDF може бути математично визначений як:

$$TF\_IDF(term_{i,j}) = tf_{ij} \times idf_i \quad (2.7)$$

де  $tf_{ij}$  – частота зустрічання терміну  $i$  у вимозі  $j$ ;

$idf_i$  – обернений показник до частоти для терміну  $i$  та вимоги  $j$ .

Провівши дослідження щодо можливостей кластеризації текстових вимог, наступний етап роботи полягає у розробці методу виявлення функціональних та нефункціональних вимог за допомогою кластерного аналізу.

## 2.5. Метод кластеризації вимог до комп'ютерних систем

Для проведення кластеризації вимог до комп'ютерних систем необхідно розробити метод, який би враховував їх текстову природу та забезпечував здатність групування подібних вимог між собою. Для цього запропоновано метод, в основі якого лежить алгоритм, продемонстрований на рис. 2.7.

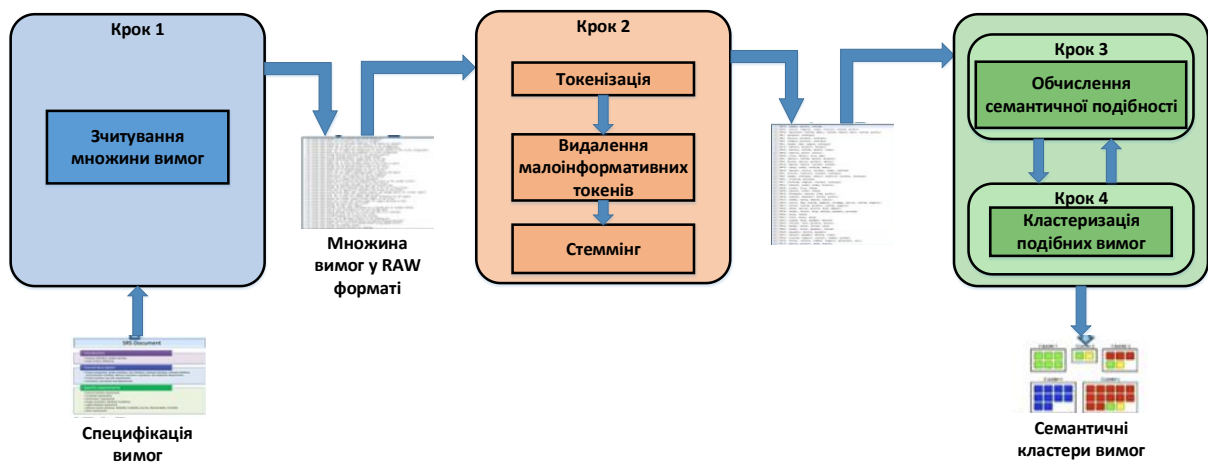


Рис. 2.7. Метод кластеризації функціональних та нефункціональних вимог

Перший крок при кластеризації функціональних і нефункціональних вимог передбачає аналіз специфікації вимог, шляхом його зчитування і перетворення у формат, придатний для опрацювання комп'ютером.

Оскільки функціональні вимоги детально описують, що повинна робити система і що мають реалізувати розробники, формулювання цих вимог документується в добре структурованому форматі в документі під назвою специфікація вимог розділ «Функції системи». Блок функціональних вимог описують згідно відповідного розділу стандарту IEEE 830.

Таким чином, цей крок аналізує текстове представлення вимог у відповідному розділі специфікації і передбачає його пошук та зчитування кожної вимоги цього розділу. Всі решта вимоги відносяться до блоку нефункціональних вимог.

Для забезпечення найкращої відповідності вимог щодо їх приналежності до певної групи, як важливий крок для ідентифікації семантичних кластерів, текст кожного твердження нормалізується. Така нормалізація досягається виконанням трьох завдань: токенізація, видалення стоп-слів та формування кореня. Ці завдання виконуються в такому порядку.

При проведенні токенізації кожна функціональна вимога розділяється на окремі оператори за допомогою розділювачів коми та крапки. Потім кожен оператор ділиться на токени на основі пробілу. Крім того, кожне складне слово поділяється на прості лексеми (наприклад, `interDepartment` поділяється на `inter` і `department`) з використанням відповідного реєстру.

Рідкісні лексеми є більш інформативними, ніж часті лексеми у вільному тексті. Це пов'язано з тим, що часті лексеми є малоінформативними та не володіють розрізнявальною силою [20, 21]. При виконанні цього завдання видаляються два типи частих лексеми. Перший тип представляє стоп-слова, такі як, «а», «for», «the», «of», «to» тощо. Другий тип представляє набір токенів із високою частотою в усіх операторах вимог. Наприклад, токени «система» та «користувач» існують у всіх операторах функціональних вимог для системи E-Shop. Таким чином, ці маркери мають високу частоту і повинні бути видалені у випадку формування підгруп у функціональних вимогах.

Зведення токенів до їх кореня під час пошуку інформації відоме як стеммінг [20]. Наприклад, «стискання» і «стиснути» зводяться до «стиск». Серед набору найбільш популярних та ефективних алгоритмів стеммінгу для англійської мови варто виділити алгоритм Портера, який в подальшому буде застосований при формуванні кластерів функціональних та нефункціональних вимог, а також для визначення груп подібних функціональних вимог.

Оскільки, у кваліфікаційній роботі необхідно реалізувати автоматизовану процедуру і засіб кластеризації функціональних та



нефункціональних вимог на основі аналізу тексту, то групування вимог відбувається за допомогою виявлення семантичних кластерів із застосуванням метрик для визначення схожості контексту.

Для формування таких кластерів пропонується наступна евристика для визначення міри подібності: «Семантична подібність вказує на відповідність тексту між лексемами, отриманими з операторів функціональних та нефункціональних вимог. Такі токени записують важливі знання домену, які представляють функціональні та нефункціональні можливості. Таким чином, коли два або більше операторів функціональних вимог використовують багато токенів, очікується, що ці вимоги пов'язані з одним завданням домену, особливо коли аналітики використовують той самий словник функціональних вимогах.

Семантична подібність між двома функціональними чи нефункціональними вимогами обчислюється на основі моделі векторного простору, який частково розглянутий раніше. Це добре відомий метод пошуку інформації. У векторному просторі кожна вимога представляється у вигляді вектора. Семантична подібність між двома вимогами може бути визначена за допомогою косинусної міри подібності

Для двох заданих вимог дана метрика використовується для визначення того, скільки релевантної семантичної інформації спільно використовується між їхніми відповідними векторами токенів:

$$SemSimilaraty(R_i, R_j) = \frac{\vec{R}_i \cdot \vec{R}_j}{\|\vec{R}_i\| \|\vec{R}_j\|} \quad (2.8)$$

де  $R_i$  та  $R_j$  – вимоги з набору функціональних та нефункціональних;

$\vec{R}_i$  – векторне представлення  $i$ -ої вимоги;

$\vec{R}_j$  – векторне представлення  $j$ -ої вимоги;

$\|\vec{R}_i\|$  – довжина вектора  $\vec{R}_i$ ;

$\|\vec{R}_j\|$  – довжина вектора  $\vec{R}_j$ .

На завершальному етапі кластеризації функціональних та нефункціональних вимог застосовується алгоритм для групування подібних вимог. Серед різних можливих алгоритмів потрібно обрати алгоритм кластеризації, який би відповідав меті роботи.

У роботі пропонується алгоритм кластеризації заснований на агломеративній ієрархічній кластеризації [16]. Однак цей крок не обмежується лише даним алгоритмом, тому може бути використаний будь-який інший алгоритм, що підтримує аналіз виявлення груп подібних за певними ознаками елементів.

Як правило, ієрархічна кластеризація передбачає на початку одиночні кластери, такі що кожен елемент даних розглядається як окремий кластер. Після цього два найбільш схожі кластери об'єднуються при кожній наступній ітерації розрахунку відстані між кластерами. Коли з'являється декілька кластерів вимог, які мають однакову схожість з виділеним кластером, алгоритм розглядає перший кластер.

Таке бінарне злиття продовжується при кожній наступній ітерації, поки не буде отримано один великий кластер. Такий кластер представляє собою дерево дендрограми, яке включає всі кластери-кандидати цих об'єктів. У кваліфікаційній роботі одиночні кластери спочатку складаються з окремих вимог, а пізніше з кількох кластерів вимог, сформованих під час кожної ітерації із використанням рекурсії.

## 2.6. Обґрунтування метрик оцінювання якості при кластеризації та класифікації вимог до комп'ютерних систем

Для оцінювання якості семантичної кластеризації при виявленні функціональних та нефункціональних вимог необхідно обґрунтувати вибір відповідних метрик. Для цього варто скористатися двома добре відомими критеріями в області інформаційного пошуку. Такими показниками є

точність (Precision) і повнота (Recall). Область визначення метрик точності і повноти лежать в діапазоні від 0 до 1

Для кожного визначеного кластера проводиться оцінювання його якості, що передбачає виконання наступних кроків.

злиття ідентифікованого кластеру (у даному випадку вимоги) з усіма фактично наявними кластерами комп'ютерної системи. Фактичні кластери групуються вручну та документуються аналітиками у специфікації вимог для кожної окремої комп'ютерної системи. Нехай  $X$  ідентифікований кластер. Кластер, який має велику кількість збігів у термінах з вимогою кластера  $X$ , називається еталонним кластером.

обчислення точності і запис значення на основі еталонного кластера відповідно до формул 2.9 та 2.10.

$$Precision = \frac{|{\{REF\_CLUSTER\} \cap \{IDE\_CLUSTER\}}|}{|{\{IDE\_CLUSTER\}}|} \quad (2.9)$$

$$Recall = \frac{|{\{REF\_CLUSTER\} \cap \{IDE\_CLUSTER\}}|}{|{\{REF\_CLUSTER\}}|} \quad (2.10)$$

де  $REF\_CLUSTER$  і  $IDE\_CLUSTER$  представляють вимогу до комп'ютерної системи як посилання та ідентифікований кластер відповідно.

Якщо точність дорівнює 1, це означає, що всі вимоги ідентифікованого кластера є підмножиною еталонного кластера. Якщо значення повноти дорівнює 1, то це стосується всіх вимог еталонного кластера підмножини ідентифікованого кластера.

Тому і точність, і повноту слід використовувати разом як додаткові частини для оцінювання правильності ідентифікованого кластера. Для того, щоб встановити кількість кластерів, які існують у функціональних та нефункціональних вимогах потрібно порівняти кількість ідентифікованих кластерів із фактичною кількістю кластерів.

Таке порівняння забезпечує використання метрики розриву кластеризації. Значення цієї метрики розраховується за допомогою формули:

$$GAP\_C = |\#I\_CLUSTER - \#A\_CLUSTER| \quad (2.11)$$

де  $\#I\_Cluster$  – кількість ідентифікованих кластерів;

$\#A\_CLUSTER$  – кількість фактичних кластерів.

Ідеальне значення  $GAP\_C$  дорівнює 0, тобто це означає, що кількість виявлених кластерів відповідає кількості фактичних кластерів.

Як підсумок, результати в основному оцінюються за допомогою трьох метрик: *Precision*, *Recall* і  $GAP\_C$ . Ці показники працюють разом як інтегрована група. Для оцінки кожного ідентифікованого кластера використовуються параметри точності і повноти окремо, тоді як  $GAP\_C$  використовується для оцінки кількості ідентифікованих кластерів.

Часто зручно об'єднувати точність і повноту в одну метрику. Для цього можна використати F1-міру, яку досить просто застосувати у випадку порівняння двох класифікаторів [8].

Оцінка F1 є середнім зваженим значенням точності та повноти. У той час як при обчисленні звичайного середнього значення використовується однакова вага для усіх значення, зважене середнє надає значно більшої ваги низьким значенням.

У результаті класифікатор отримає високу оцінку F1, лише якщо повнота і точність високі. Формула для обчислення F1-міри має наступний вигляд:

$$F1 = \frac{2TP}{2TP+FP+FN} \quad (2.12)$$

Точність представляє собою частку правильно класифікованих спостережень. Це більш загальний показник, оскільки він обчислює

кількість правильних класифікацій у цілому. Це хороший показник, коли цільові класи в даних (майже) збалансовані:

$$Acc = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.13)$$

$\chi^2$ (Хі-квадрат) є загальним статистичним тестом, який вимірює відхилення від очікуваного розподілу, якщо припустити, що поява ознаки фактично не залежить від значення класу [13]. Він вимірює відсутність незалежності між терміном  $t$  і класом  $c$ . Дана міра обчислюється за формулою:

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \quad (2.14)$$

де  $N$  – загальна кількість вимог;

$A$  – кількість разів, коли  $t$  і  $c$  зустрічаються одночасно;

$B$  – кількість разів, коли  $t$  трапляється без  $c$ ,

$C$  – кількість разів, коли  $c$  відбувається без  $t$ ;

$D$  – кількість разів, коли ні  $c$ , ні  $t$  не зустрічаються.

Таким чином, розроблено метод та обґрунтовано метрики для оцінювання якості кластеризації функціональних та нефункціональних вимог до комп'ютерних систем, які в подальшому будуть реалізовані у вигляді програмного забезпечення.

## 2.7. Висновки до розділу

Основні наукові та практичні результати, які отримані у даному розділі полягають в наступному:

1. Досліджено сучасні алгоритми і методи кластерного аналізу у сфері опрацювання текстової інформації і встановлено, що найбільш

оптимальним рішенням за точністю і оптимальністю використання ресурсів є алгоритми k-means, які доцільно застосовувати при кластеризації функціональних та нефункціональних вимог до комп'ютерних систем.

2. Запропоновано процедуру і метод агломеративної ієрархічної кластеризації для визначення груп функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу забезпечити автоматизацію та підвищити ефективність процесу визначення та аналізу вимог на відповідному етапі життєвого циклу.

3. Обгрунтовано методи перетворення текстових даних, якими представляються функціональні та нефункціональні вимоги до комп'ютерних систем, у векторний простір семантичних ознак, що дало змогу в подальшому застосувати методи машинного навчання.

4. Запропоновано та обгрунтовано доцільність застосування метрик точності і повноти для визначення якості кластеризації вимог до комп'ютерних систем, а також введено метрику порівняння, яка дає змогу визначити відхилення кількості ідентифікованих кластерів, визначених за допомогою методу агломеративної ієрархічної кластеризації та фактично наявних кластерів.

### РОЗДІЛ 3

## ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ КЛАСТЕРИЗАЦІЇ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ

3.1. Аналіз та препроцесинг наборів даних для проведення кластеризації вимог

Відмінність між функціональними та нефункціональними вимогами до комп'ютерних систем не є чітко розмежованою. Нефункціональні вимоги, наприклад ті, що пов'язані з безпекою, можуть породжувати багатофункціональні вимоги, пов'язані з відповідними послугами [8]. Тому пропонується ряд спроб автоматизувати ідентифікацію такого розрізнення.

Для кращого розуміння та аналізу переваг кластеризації вимог у вигляді набору семантичних кластерів, пропонується спочатку розглянути приклад комп'ютерної системи у вигляді електронного магазину з продажу електроніки. Ця система забезпечує виконання функціональних вимог і функцій, які дають змогу реалізовувати онлайн-продажі, формувати розповсюдження та маркетинг електроніки. Крім того, вона фокусується на компанії, зацікавлених сторонах і програмах. Ця система підтримує виконання 62 функціональних вимог. На рис. 3.1 показано функціональні вимоги та відповідні їм групи, задокументовані вручну в документі специфікації вимог до електронного магазину [20].

На етапі розробки вимог колекція неструктурованих вимог програмного забезпечення вручну групується в узгоджені кластери, а потім формується специфікація, як показано на рис. 3.2.

Наприклад, функціональні вимоги [«Система повинна відображати детальну інформацію про вибрані продукти», «Система повинна надавати параметри перегляду, щоб переглянути деталі продукту»] вручну групуються разом у кластер із назвою «деталі продукту».

Виконання такого завдання займає багато часу та може викликати помилки, особливо у великому проекті, оскільки функціональні вимоги організовані в різні підсистеми, які складають комп'ютерну систему.

- 1 The system shall display all the products that can be configured.
  - 2 The system shall allow user to select the product to configure.
  - 3 The system shall enable user to add one or more component to the configuration.
  - 4 The system shall notify the user about any conflict in the current configuration.
  - 5 The system shall allow user to update the configuration to resolve conflict in the current configuration.
  - 6 The system shall provide browsing options to see product details.
  - 7 The system shall allow user to confirm the completion of current configuration.
  - 8 The system shall display detailed product categorization to the user.
  - 9 The system shall enable user to enter the search text on the screen.
  - 10 The system shall enable user to select multiple options on the screen to search.
  - 11 The system shall display all the matching products based on the search.
  - 12 The system shall display only 10 matching result on the current screen.
  - 13 The system shall enable user to navigate between the search results.
  - 14 The system shall notify the user when no matching product is found on the search.
  - 15 The system shall allow user to create profile and set his credential.
  - 16 The system shall display detailed information of the selected products.
  - 17 The system shall authenticate user credentials to view the profile.
  - 18 The system shall allow user to update the profile information.
  - 19 The system shall display both the active and completed order history in the customer profile.
  - 20 The system shall display the most frequently searched items by the user in the profile.
  - 21 The system shall allow user to register for newsletters and surveys in the profile.
  - 22 The system shall provide online help, FAQ's customer support, and site map options for customer support.
  - 23 The system shall allow user to select the support type he wants.
  - 24 The system shall allow user to enter the customer and product information for the support.
  - 25 The system shall display the customer support contact numbers on the screen.
  - 26 The system shall allow user to select the order from the order history.
  - 27 The system shall display the detailed information about the selected order.
  - 28 The system shall allow user to enter the contact number for support personnel to call.
  - 29 The system shall display the online help upon request.
  - 30 The system shall display detailed invoice for current order once it is confirmed.
- .....

Рис. 3.1. Приклад неструктурованих функціональних вимог

- .....
- ### 3. Specific Requirements
- The specific requirements are –
- #### 3.1 Functionality
- This subsection contains the requirements for the e-store. These requirements are organized by the features discussed in the vision document. Features from vision documents are then refined into use case diagrams and to sequence diagram to best capture the functional requirements of the system. All these functional requirements can be traced using tractability matrix.
- ##### 3.1.1 Self Configured to Ordered Products.
- 3.1.1.1 The system shall display all the products that can be configured.
  - 3.1.1.2 The system shall allow user to select the product to configure.
  - 3.1.1.3 The system shall display all the available components of the product to configure
  - 3.1.1.4 The system shall enable user to add one or more component to the configuration.
  - 3.1.1.5 The system shall notify the user about any conflict in the current configuration.
  - 3.1.1.6 The system shall allow user to update the configuration to resolve conflict in the current configuration.
  - 3.1.1.7 The system shall allow user to confirm the completion of current configuration
- ##### 3.1.2 Provide comprehensive product details.
- 3.1.2.1 The system shall display detailed information of the selected products.
  - 3.1.2.2 The system shall provide browsing options to see product details.
- ##### 3.1.3 Detailed product Categorizations
- The system shall display detailed product categorization to the user.
- ##### 3.1.4 Provide Search facility.
- The system shall enable user to enter the search text on the screen.  
 The system shall enable user to select multiple options on the screen to search.  
 The system shall display all the matching products based on the search  
 The system shall display only 10 matching result on the current screen.  
 The system shall enable user to navigate between the search results.  
 The system shall notify the user when no matching product is found on the search.
- ##### 3.1.5 Maintain customer profile.
- .....

Рис. 3.2. Приклад функціональних вимог структурованих у кластери



Процес і способи групування функціональних вимог рідко зустрічаються у літературі з розробки вимог. Соммервіл у [15] класифікує їх на два рівні: вимоги користувача та системні вимоги. У [19] та ін. вручну класифікували лише п'ять класів функціональних вимог. Ці класи включають вхідні дані, вихідні дані, збереження даних, правила застосування та дії.

На відміну від груп функціональних вимог, більшість існуючих підходів зосереджено на класифікації нефункціональних вимог. Крім того, класи цих вимог є стандартизованими і дуже деталізованими. Наприклад, нефункціональні вимоги у [15] класифікуються за трьома основними типами: зовнішні вимоги, вимоги до продукту та організаційні вимоги.

Потім вимоги до продукту класифікуються на чотири категорії: вимоги до ефективності, зручності використання, надійності та безпеки. Крім того, вимоги до ефективності детально описані щодо властивостей простору та продуктивності. Решта основних груп нефункціональних вимог постійно деталізуються на нижчі рівні.

Отже, отримана схема класифікації є ієрархією, що складається з трьох рівнів. Пропонується велика кількість підходів для автоматизації процесу ідентифікації та класифікації нефункціональних вимог.

У [17] запропоновано метод на основі онтології для виявлення та класифікації речень вимог на чотири типи (ремонтпридатність, надійність, портативність, безпека та зручність використання/корисність) за допомогою машинного класифікатора опорних векторів.

У [18] запропоновано методику, засновану на правилах, для ідентифікації та класифікації тексту вимог у корпусі PROMISE на вісім типів нефункціональних вимог. Їх техніка використовувала лінгвістичні зв'язки між висловлюваннями вимог для виділення тематичного контексту.

Варто відмітити, що нефункціональні вимоги визначають бажані характеристики якості системи, яка буде розроблена, і часто впливають на архітектуру системи більше, ніж функціональні вимоги.

Ще одне визначення нефункціональних вимог визначення, згідно з [19], полягає в тому, що: нефункціональна вимога відповідає набору обмежень, накладених на систему, що розробляється, встановлюючи, наприклад, наскільки вона приваблива, корисна, швидка або надійна.

Нефункціональні вимоги формують категорію вимог, яка представляє собою набір необхідних загальних атрибутів системи, включаючи портативність, надійність, ефективність, людську інженерію, можливість тестування, зрозумілість і модифікованість. Нефункціональні вимоги можна розділити на підкатегорії.

Вимоги якості іноді класифікуються як вимоги до продукції, організаційні вимоги або вимоги, пов'язані з процесом, і зовнішні вимоги. Різниця між нефункціональними та функціональними вимогами в розробці комп'ютерних систем повинна залежати від того, «як» і «що» системи виконують або які пропонують ресурси [20]. Репозиторій PROMISE розглядає 11 підкатегорій для цього типу вимог. Тип і опис кожної підкатегорії приведено у табл. 3.1.

*Таблиця 3.1*

### **Кластери нефункціональних вимог**

Назва кластеру	Опис
Доступність (A)	Описує, наскільки система доступна для користувача в певний момент часу
Стійкість до відмов (FT)	Ступінь, до якого система, продукт або компонент працює належним чином, незважаючи на наявність несправностей апаратного чи програмного забезпечення.
Відповідність стандартам (L)	Сертифікати або ліцензії, якими система повинна володіти і відповідати.
Look & Feel (LF)	Описує стиль зовнішнього вигляду системи

Продовження табл. 3.1

Назва кластеру	Опис
Ремонтопридатність (MN)	Ступінь ефективності, з якою продукт або система можуть бути модифіковані під час їх супроводу
Працездатність (O)	Ступінь, до якого продукт або система має атрибути, які полегшують роботу та контроль
Продуктивність (PE)	Продуктивність відносно кількості використаних ресурсів (апаратних і програмних).
Портативність (PO)	Ступінь ефективності, з якою систему, продукт або компонент можна перенести з одного апаратного або програмного забезпечення в інше робоче або користувацьке середовище.
Масштабованість (SC)	Ступінь, до якого продукт або система може бути ефективно адаптована до іншого апаратного чи програмного забезпечення або обладнання, що розвивається або інших операційних середовища чи середовища використання.
Безпека (SE)	Ступінь, до якого продукт або система захищає інформацію та дані, щоб особи або інші продукти чи системи мали ступінь доступу до даних відповідно до їх типів і рівнів авторизації.
Зручність використання (USA)	Ступінь, до якого продукт або система можуть використовуватися певними користувачами для досягнення визначених цілей з ефективністю, дієвістю, і задоволеністю в певному контексті використання.

Для проведення експериментів за допомогою розробленого у другому розділі методу запропоновано використовувати мову програмування Python, а в якості одного з даних сетів обрати базу даних «PROMISE\_exp».

Цей репозиторій є розширеною версією оригінального набору даних PROMISE, що є загальнодоступним репозиторієм машинного навчання UCI.

Оригінальний репозиторій складається з попередньо розміченого набору з 255 функціональних вимог та 370 нефункціональних. Нефункціональні вимоги розділені на 11 різних типів. Розширена версія [3] складається з 969 вимог, і її склад представлено у табл. 3.2.

Розширення відбулося за допомогою цілеспрямованого веб-пошуку документів, що містять записи про вимоги до комп'ютерних систем та програмного забезпечення.

Таблиця 3.2

**Розподіл за типами вимог у наборі даних «PROMISE\_exp»**

Тип вимоги	Кількість вимог
Функціональні вимоги	444
Доступність	31
Відповідність стандартам	15
Look & Feel (LF)	49
Ремнотопридатність	24
Працездатність	77
Продуктивність	67
масштабованість	22
Безпека	125
Зручність використання	85
Стійкість до відмов	18
Портативність	12
Загалом	969

На рис. 3.3 наочно показано, наскільки незбалансованими є класи вимог. Єдиний випадок, коли групи вимог добре розподілені, це коли всі підкласи нефункціональних вимог згруповані в один унікальний клас під

час аналізу бінарною класифікації або кластеризації. При цьому забезпечується такий поділ, що до класу функціональних вимог належить 444 вимоги, а для класу нефункціональних – 525.

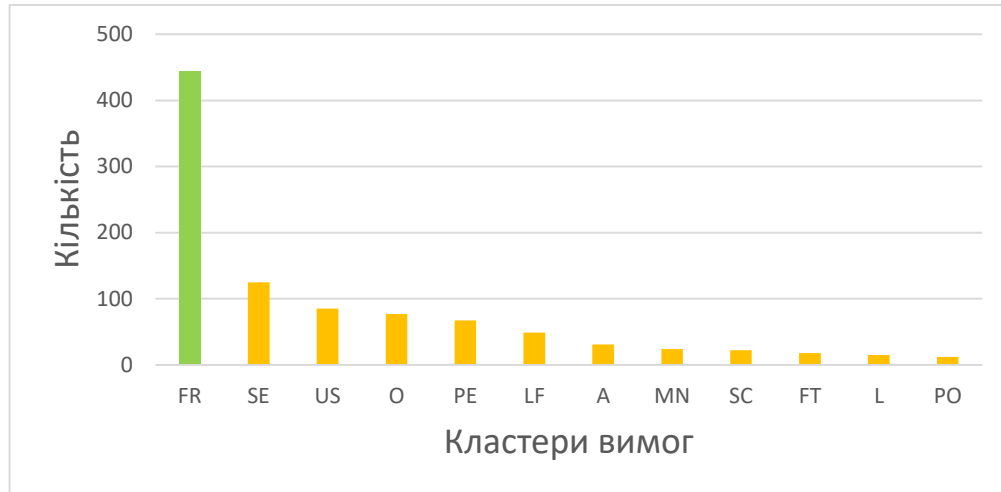


Рис. 3.3. Кількість елементів у класах вимог

Процедуру попереднього опрацювання даних у наборі даних від «Promise\_exp», а також весь конвеєр при роботі з розміченими даними представлено на рис. 3.4.

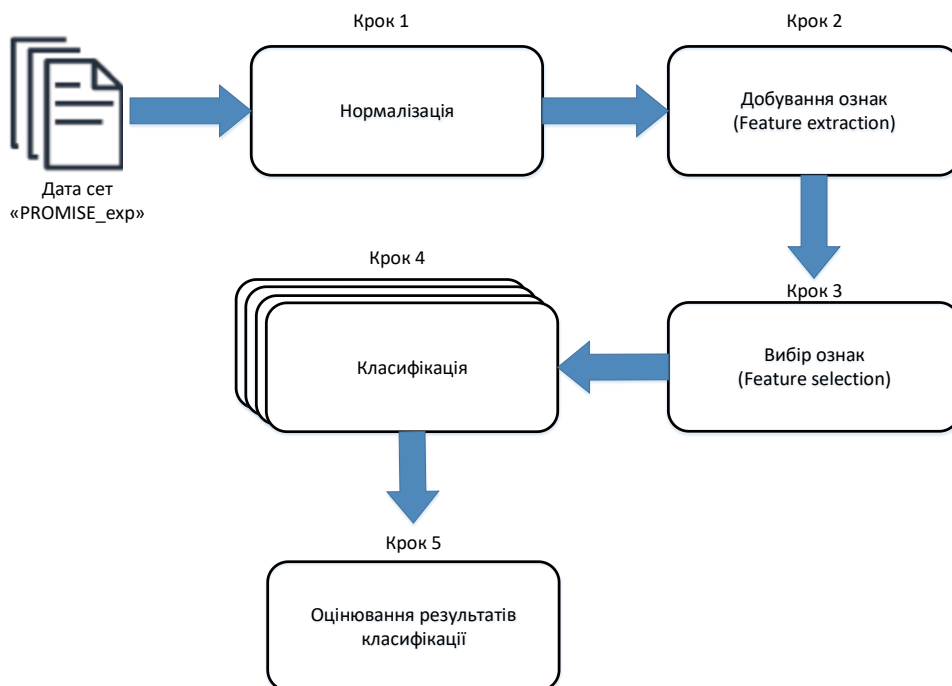


Рис. 3.4. Конвеєр при класифікації вимог у дата сеті з мітками даних

Усі документи в базі даних пройшли процес нормалізації, який є першим кроком у конвеєрі, показаному на рис. 3.4. У процесі нормалізації застосовувалась бібліотека NLTK, компоненти якої дають змогу ефективно опрацьовувати природну мову. На рис. 3.5 показано фрагмент вимог, які містяться у базі даних перед процесом нормалізації (очищення тексту).

	Text	Class
1	The system shall refresh the display every 60 s.	PE
2	The application shall match the color of the schema set forth by Department of Homeland Security.	LF
3	If projected the data must be readable. On a 10 × 10 projection screen 90% of viewers must be able to read Event/Activity data from a viewing distance of 30.	US
4	The product shall be available during normal business hours. As long as the user has access to the client PC the system will be available 99% of the time during the first six months of operation.	A
5	If projected the data must be understandable. On a 10 × 10 projection screen 90% of viewers must be able to determine that Events or Activities are occurring in current time from a viewing distance of 100.	US
...	...	...
965	The system should be portable to various operating environments.	PO

Рис. 3.5. Фрагмент корпусу вимог перед проведення процедури нормалізації

При виконанні нормалізації усі букви у тексті вимог було приведено до нижнього регістру. Потім слова, які є сполучними або які взагалі не мали значення при опрацюванні вимог, було видалено з документів. Ці слова називаються стоп-словами і, зазвичай, є найпоширенішими, коли виконується об'єднання кількох документів. Такі артиклі як «the», «a» і тому подібні, вважаються нерелевантними словами. У результаті, одержано дані у вигляді, який показаний на рис. 3.6.

Останнім кроком у процесі нормалізації було перетворення сполучених слів у їх кореневу форму, наприклад, слово «користувачі» було змінено на «користувач», а «вказаний» — на «вказати».

	Text	Class
1	System shall refresh display every second.	PE
2	Application shall match color schema set forth department homeland security.	LF
3	Project data must readable projection screen viewer must able read event activity data view distance.	US
4	Product shall available normal business hour long user access client pc system available time first six month operation.	A
5	Project data must understandable projection screen viewer must able determine event activity occur current time view distance.	US
...	...	...
965	System portable various operate environment.	PO

Рис. 3.6. Нормалізований текст вимог

Код, використаний для реалізації процесу нормалізації, проілюстровано на рис. 3.7.

```
#1 Remove blank rows .
Corpus [ 2 'text ']. dropna ( inplace = True )
#3 Change all the text to lower case .
Corpus [ 4 'text ' ] = [ entry . lower () for entry in Corpus ['text ']]
#5 Tokenization
Corpus [ 6 'text ' ] = [ word_tokenize ( entry ) for entry in Corpus ['text ']]
#7 Remove Stop words , Non - Numeric and perform Word Lemmenting .
tag_map = defaultdict ( 8 lambda : wn. NOUN )
tag_map [ 9 'J' ] = wn. ADJ
tag_map [ 10 'V' ] = wn. VERB
tag_map [ 11 'R' ] = wn. ADV
12
for 13 index , entry in enumerate ( Corpus ['text ']):
14
Final_words = [] 15
word_Lemmatized = WordNetLemmatizer () 16
for 17 word , tag in pos_tag ( entry ):
#18 Below condition is to check for Stop words and consider only
alphabets 19 if word not in stopwords . words ('english ')
and 20 word . isalpha ():
word_Final = word_Lemmatized . lemmatize ( word , tag_map [ tag [0]]) 21
Final_words . append ( word_Final ) 22
Final_words = 23 ' ' . join ( Final_words )
Corpus .loc [index , 24 'text ' ] = str ( Final_words )
```

Рис. 3.7. Фрагмент програмного коду для реалізації процедури нормалізації тексту вимог до комп'ютерних систем

Наступний крок стосується векторного представлення вимог до комп'ютерних систем.

### 3.2. Результати векторного представлення вимог до комп'ютерних систем

Як зазначалось раніше, для представлення вимог у векторному просторі запропоновано скористатися підходом Bag of Words та TF-IDF. Після виконання нормалізації всі документи пройшли процес векторизації.

Цей етап необхідний, щоб можна було використовувати інформацію, отриману з документів, у моделях машинного навчання відповідно до етапу «Добування ознак», як показано на рис 3.4.

Дві форми добування ознак на основі частотного підходу та «сумки слів» будуть порівняні на етапі класифікації, де буде можливість спостерігати, які з цих методів покращили продуктивність використаних алгоритмів.

Для того, щоб оцінити, які слова у тексті вимог мали найбільший вплив на результати визначення класів (кластерів), обчислено суму рядків кожної матриці, отриманої у результаті векторизації. У табл. 3.3 наведено 10 слів з найвищим балом для кожної техніки.

*Таблиця 3.3*

#### **Найбільш важливі слова при векторизації тексту вимог**

Рейтинг слів	Bag of Words	TF-IDF
1	must	must
2	information	information
3	user	sensitive
4	website	website
5	security	registered
6	registered	security



Продовження табл. 3.3

Рейтинг слів	Bag of Words	TF-IDF
7	secure	secure
8	sensitive	user
9	high	third
10	agreement	issue

У табл. 3.3 можна помітити, що слова ті самі, але з 3-ї позиції порядок важливості змінюється у BoW та TF-IDF.

### 3.3. Результати групування вимог до комп'ютерних систем за відомими мітками класу

Нормалізований і векторизований корпус використовувався для навчання та тестування продуктивності з використанням чотирьох алгоритмів: k найближчих сусідів, метод опорних векторів, мультикласовий наївний Баєсівський класифікатор і лінійна регресія. Застосування цих підходів відбуваються на четвертій стадії конвеєра, який представлений на рис. 3.4.

Ці алгоритми були використані для класифікації вимог до комп'ютерних систем за трьома різними типами деталізації:

- класифікація на рівні функціональних та нефункціональних вимог (бінарна класифікація);
- класифікація на рівні підкласів нефункціональних вимог;
- класифікація функціональних вимог та підкласів нефункціональних вимог.

У алгоритмах SVM, MNB і LR використовувався гіперпараметр під назвою «class\_weight», що враховує значення міток класу для автоматичного коригування ваг обернено пропорційно частотам класів у вхідних даних.

Усі гіперпараметри, як з алгоритмів векторизації, так і з класифікації, були обрані за допомогою функції під назвою «GridSearchCV». Ця функція виконує вичерпний пошук за вказаними значеннями параметрів для оцінювача та вибирає найкращу комбінацію на основі певного параметра оцінки, у даному випадку F-міри. Іншими словами, ця функція перевіряє всі можливі комбінації між параметрами та повертає комбінацію, яка досягла найкращого результату. Ця виграшна комбінація є комбінацією, яка використовується в процесі векторизації/класифікації.

Далі необхідно обчислити ефективність та якість процесу класифікації вимог чотирьох алгоритмів за допомогою крос-валідації, яка виконується на завершальному етапі конвеєра.

При крос-валідації формується послідовність відповідей, де кожна підмножина даних використовується як навчальний набір і в якості набору перевірки. Такий підхід є ефективним, оскільки дані не збалансовані, тобто кількість даних для різних типів вимог суттєво відрізняється.

У цьому експерименті корпус був розділений на десять підмножин (10-кратних), з яких 9 використовувалися для навчання алгоритмів, що відповідає 90 відсоткам бази, а 1 використовувався для виконання тестів, що відповідає 10 відсоткам бази даних вимог.

На основі перехресної перевірки обчислено точність, повноту і F-міру результатів. На рис. 3.8 показано лістинг програмного коду, що передбачає створення функції для виконання перехресної перевірки.

Корпус вимог передається в якості параметрів функції, а також вказуються алгоритми класифікації (модель), об'єкти, які забезпечують векторизацію вимог до комп'ютерних систем, і кількість підмножин вимог. Таким чином, програмний код добуває функції, навчає та перевіряє дані, виводячи в кінці показники продуктивності, отримані за допомогою тесту.

```

def 1 kfoldcv (data , classifier , vectorizer , k_best , k = 10):
2kf = KFold ( n_splits =k, shuffle = True , random_state = 60) 3
kf. get_n_splits ( data )
4report_test = pd. Series () 5
report_prediction = pd. Series ()
6 7
for train_indices , test_indices in kf. split ( data ):
train_text = data . iloc [ train_indices ][ 8 'text ' ]
train_class = data . iloc [ train_indices ][ 9 'class ' ]
10
test_text = data . iloc [ test_indices ][ 11 'text ' ]
test_class = data . iloc [ test_indices ][ 12 'class ' ]
13
pipeline = Pipeline ([ ( 14 'vect ' , vectorizer ) ,
(15'chi ' , SelectKBest ( score_func =chi2 , k= kbest ) ) ,
(16'clf ' , classifier )])
17
model = pipeline . fit ( train_text , train_class ) 18
19
predictions = model . predict ( test_text ) 20
21
report_test = np. concatenate (( report_test , test_class ) , axis =0) 22
report_prediction = np. concatenate (( report_prediction , predictions ) , axis =0) 23
24
print 25 ( metrics . classification_report ( report_test , report_prediction , digits = 2))

```

Рис. 3.8. Реалізація процесів класифікації та крос-валідації

При проведенні експериментів використовувалась бібліотека Scikit-learn, яка поєднує у собі декілька алгоритмів машинного навчання для роботи з розміченими та нерозміченими даними. Цей інструмент було обрано, оскільки він містить алгоритми як основні, так і додаткові інструменти, які реалізують VoW, TF-IDF і крос-валідацію.

### 3.4. Імплементация алгоритмів кластеризації функціональних вимог до комп'ютерних систем

Реалізація алгоритмів кластеризації функціональних виконується шляхом побудови дерева дендрограми, яке є представленням та візуалізацією кластерів. Дендрограма формується за допомогою методу агломеративної ієрархічної кластеризації, що запропонований у дургому розділі кваліфікаційної роботи.

У кваліфікаційній роботі виконується адаптація цього методу для побудови дерева дендрограми на заданому наборі функціональних вимог та з врахуванням основі алгоритму, який показано на рис. 3.9.

Алгоритм, наведений на рис. 3.9 починається зі створення кластера для кожної окремої функціональної вимоги. Потім два найбільш подібні кластери об'єднуються в кожній ітерації. У даній роботі ця подібність відноситься до семантичної схожості, яка обчислюється між векторами термів двох порівнюваних кластерів за допомогою формули(2.1).

```

Input: FRs // Functional Requirements
Output: dendgr // Dendrogram Tree
1 stack clusters ← FRs
2 while (|clusters| > 1) do
3   (Cl1, Cl2) ← mostSimilarClusters(clusters)
4   Pop(Cl1, clusters) // Cl1: Cluster 1
5   Pop(Cl2, clusters) // Cl2: Cluster 2
6   Cl3 ← Merge(Cl1, Cl2)
7   Push(Cl3, clusters)
8 end
9 dendgr ← clusters
10 return dendgr

```

Рис. 3.9. Алгоритм побудови дерева дендрограми

Наступний крок передбачає, що два найбільш схожі кластери замінюються новим кластером, який є результатом злиття попередніх. Цей процес триває до тих пір, поки не буде отримано єдиний кластер. Такий кластер представляє дендрограму, яка містить набір вкладених кластерів. На рис 3.10 показано приклад дерева дендрограми.

На нижньому рівні створюється кластер для кожної функціональної вимоги. На верхньому рівні всі функціональні вимоги належать до єдиного великого кластера.

Внутрішні вузли (внутрішні кластери) відносяться до нових кластерів, які утворюються шляхом злиття кластерів, які з'являються як нащадки нових.

Оскільки дерево дендрограми є ієрархією вкладених кластерів, то коли цю ієрархію обрізають у певній точці на основі попередньо визначених критеріїв, одержують набір кластерів (рис. 3.10). Кожен з них є семантичним кластером-кандидатом. Тому на цьому кроці запропоновано алгоритм, який наведений на рис. 3.11. Дана послідовність кроків заснована на пошуку в глибину для визначення відповідної точки «різання». Входом цього алгоритму є дерево дендрограми, а виходом – набір семантичних кластерів. Алгоритм порівнює значення семантичної подібності кожного вузла (батьківського вузла), починаючи з кореня, з його дочірніми вузлами (його лівим і правим вузлами).

Якщо значення подібності батьківського вузла менше, ніж середнє значення подібності його нащадків, алгоритм переходить до наступних вузлів ієрархії. В іншому випадку батьківський вузол ідентифікується як семантичний кластер, додається до накопичувача (semCluster), і алгоритм переходить до наступного вузла в стеку (купі). Відповідно до цього в ході обходу визначаються найбільш відповідні семантичні кластери.

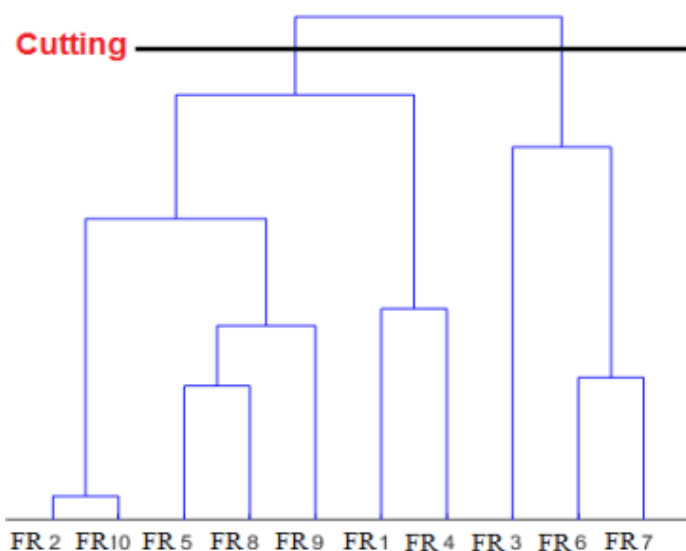


Рис. 3.10. Приклад формування дендрограми

Рис. 3.11 є простим прикладом для демонстрації уявної картини виконання алгоритму, що представлений на рис. 3.10. На цьому рисунку

горизонтальна лінія перетинає ієрархію в двох точках (так звані точки розрізу). Таким чином, виділено два семантичних кластери. Перший кластер містить сім функціональних вимог [ФВ2, ФВ10, ФВ5, ФВ8, ФВ9, ФВ1, ФВ4], тоді як другий кластер складається з трьох – [ФВ3, ФВ6, ФВ7].

Ефективність запропонованого підходу оцінюється за допомогою специфікації вимог чотирьох комп'ютерних систем з відкритим кодом програмного забезпечення, які відносяться до різних доменів і мають різний розмір. Це система електронної комерції «E-Store», система формування подій і нагадувань «WASP», система управління університетом «UUIS» і медична система управління даними пацієнтів щодо їх ментального здоров'я «MHC-PM».

```

Input: dendgr // a dendrogram tree of FRs
Output: semClusters // array of semantic clusters
1 stack pile
2 pile.push(root(dendgr))
3 while (|pile| > 0) do
4   | parent ← pile.pop()
5   | Cl1 ← getLeftCluster(parent, dendgr)
6   | Cl2 ← getRightCluster(parent, dendgr)
7   | avg ← (Similarity(Cl1), Similarity(Cl2))/2
8   | if (Similarity(parent) > avg) then
9     |   add(parent, semClusters)
10  | else
11  |   pile.push(Cl1)
12  |   pile.push(Cl2)
13  | end
14 end
15 return semClusters

```

Рис. 3.11. Ідентифікація кластерів на основі аналізу дендрограми

«E-Store» забезпечує можливість реалізації процесів онлайн-продажів електроніки і додаткових маркетингових і рекламних акцій.

Система «WASP» представляє собою платформу сервісів з проведення різноманітних подій та управління ними.

Комп'ютерна система «UUIS» є єдиною системою управління університетом отримати доступ до інтегрованої інвентаризації та керувати нею.

Система «МНС-PM» – це система управління пацієнтами, які мають ментальні розлади. У табл. 3.4 наведено статистичну інформацію щодо кожної системи в контексті фактичної кількості кластерів, їх розміру і кількості функціональних вимог.

Таблиця 3.4

#### Статистичні дані щодо кластерів функціональних вимог

Комп'ютерні системи	Кількість кластерів	Розмір кластерів			Кількість функціональних вимог
		Мін.	Сер.	Макс.	
«E-Store»	20	1	3	7	62
«WASP»	14	1	4,7	8	66
«UUIS»	11	1	2,3	4	25
«MHP-PM»	6	1	3,2	6	19

Застосовуючи метод ієрархічної кластеризації запропонований у розділі 2 одержано результати, які можна оцінити за допомогою метрик точності і повноти. У табл. 3.5 наведено результати розрахунку середніх значень цих критеріїв для кожної з комп'ютерних систем.

Таблиця 3.5

#### Середнє значення метрик якості кластеризації

Комп'ютерні системи	Середня точність (Precision)	Середня повнота (Recall)
«E-Store»	0,80	0,61
«WASP»	0,83	0,54

## Продовження таблиці 3.5

Комп'ютерні системи	Середня точність (Precision)	Середня повнота (Recall)
«UUIS»	0,72	0,60
«МНР-РМ»	0,78	0,57

Значення, наведені в табл. 3.5 вказують на те, що функціональні вимоги цих ідентифікованих кластерів семантично згруповані разом, щоб утворити семантичні кластери. Дійсно, значення точності мають високі значення, які коливаються в залежності від типу системи від 0,72 до 0,83, а значення повноти знаходяться в прийнятному діапазоні від 54 до 61.

На рис. 3.12 графічно представлено середні значення точності та повноти ідентифікованих кластерів у кожній комп'ютерній системі з відповідною кількістю функціональних вимог.

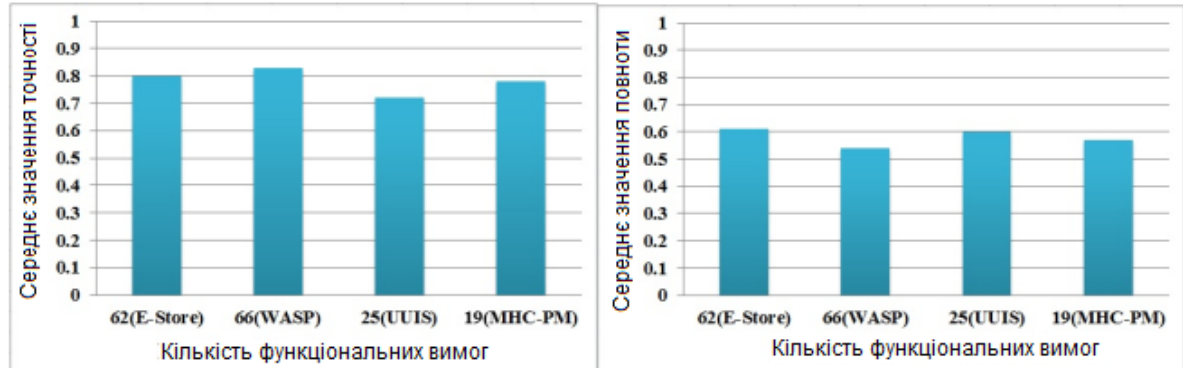


Рис. 3.12. Візуалізації метрик кластеризації функціональних вимог

Як видно з рис. 3.12, середні значення для точності та повноти відносно близькі одні до одних. Це є показником того, що запропонований підхід працює ефективно незалежно від кількості функціональних вимог.

У табл. 3.6 продемонстровано статистичні дані для значень точності і повноти ідентифікованих кластерів для кожної комп'ютерної системи. Помітно, що стандартне відхилення цих значень є низьким. Крім того, максимальні значення для визначених метрик досягають ідеального



значення. Як наслідок, це є доказом того, що підхід забезпечує визначення семантичних кластерів, які завжди мають високі значення точності та повноти при різних розмірах комп'ютерних систем.

Таблиця 3.6

### Статистичні показники ідентифікованих кластерів

Комп'ютерні системи	Точність (Precision)			Повнота (Recall)		
	Сер.	Похибка	Макс	Сер.	Похибка	Макс
«E-Store»	0,80	0,28	1	0,61	0,29	1
«WASP»	0,83	0,23	1	0,54	0,31	1
«UUIS»	0,72	0,29	1	0,60	0,29	1
«MHP-PM»	0,78	0,25	1	0,57	0,35	1

Виходячи з одержаних експериментальних результатів, запропонований підхід дозволяє ідентифікувати семантичні кластери функціональних вимог з високою мірою точності і повноти та незалежно від кількості функціональних вимог.

Важливо також сформулювати статистичні дані про ідентифіковані кластери. У табл. 3.7 для кожної комп'ютерної системи наведено статистичну інформацію щодо кількості ідентифікованих кластерів та розмірів цих кластерів (мінімальний, середній та максимальний розміри).

Таблиця 3.7

### Статистичні дані щодо ідентифікованих кластерів та їх розмірів

Комп'ютерні системи	Кількість кластерів	Розмір кластерів		
		Мін.	Сер.	Макс.
«E-Store»	19	2	3,3	11
«WASP»	18	1	3	13
«UUIS»	9	1	2,8	8

## Продовження таблиці 3.7

Комп'ютерні системи	Кількість кластерів	Розмір кластерів		
		Мін.	Сер.	Макс.
«МНР-РМ»	6	1	3,2	6

Порівнюючи табл. 3.7 з табл. 3.5 (статистика щодо фактичних кластерів), можна відзначити, що кількість ідентифікованих кластерів близька до фактичної кількості кожної комп'ютерної системи. Крім того, розмір визначених кластерів подібний до їхніх відповідних фактичних кластерів. Отже, ці статистичні дані є показником ефективності запропонованого методу. Візуалізація зв'язку між кількістю ідентифікованих кластерів і кількістю функціональних вимог показана на рис. 3.13. Крім того, ця діаграма показує зв'язок між кількістю ідентифікованих кластерів і кількістю фактичних кластерів.

На основі рис. 3.13 стає зрозуміло, що кількість ідентифікованих кластерів дуже близька до кількості фактичних кластерів у різних комп'ютерних системах. Цей факт стає очевидним, аналізуючи табл. 3.8, яка описує незначні відхилення між кількістю фактичних та ідентифікованих кластерів для кожного програмного продукту.

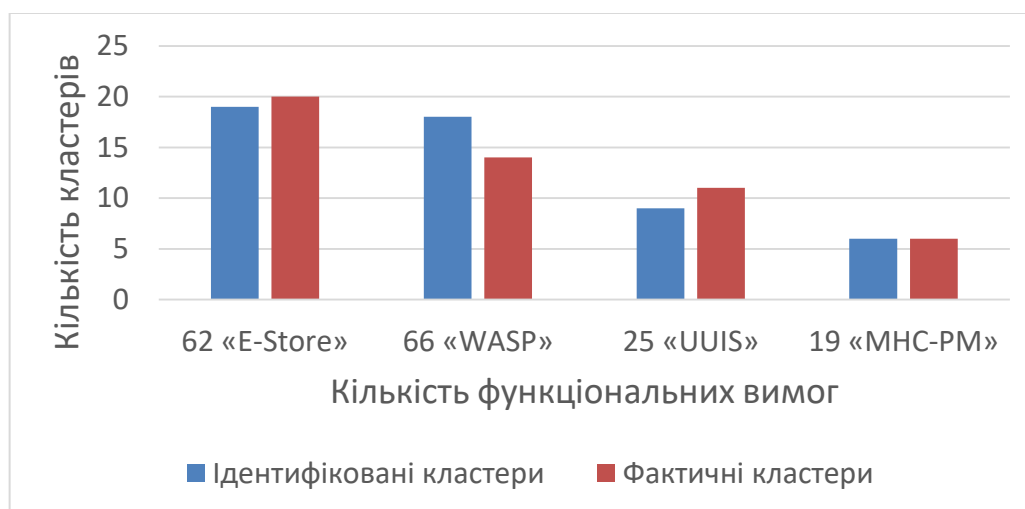


Рис. 3.13. Порівняння фактичної кількості кластерів з ідентифікованою кількістю

### 3.5. Висновки до розділу

Основні наукові та практичні результати, які отримані у даному розділі полягають в наступному:

1. Проведено аналіз та препроцесинг наборів даних, які містять функціональні та нефункціональні вимоги до комп'ютерних систем, що дало змогу провести їх очищення і в подальшому підвищити якість кластеризації вимог.

2. На практиці застосовано програмні методи векторизації текстових даних, що дозволило в подальшому визначити кластери в межах функціональних та нефункціональних вимог із використанням запропонованого методу ієрархічної кластеризації.

3. Запропоновано процедури і побудовано програмний конвеєр опрацювання функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу практично реалізувати технології машинного навчання в процесі їх аналізу тим самим скоротивши часові витрати на виконання проекту та підвищивши якість його виконання.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 4.1. Охорона праці

У кваліфікаційній роботі магістра досліджено технології виявлення функціональних та нефункціональних вимог до комп'ютерних систем. Обов'язковим елементом дослідження є визначення та аналіз вимог з охорони праці і техніки безпеки при розробці програмного засобу інтелектуального аналізу і виявлення приналежності до вимог до одного із кластерів і проведенні експериментальних досліджень, що супроводжується використанням комп'ютерної техніки. Дотримання норм і правил охорони праці є важливим аспектом у контексті дотримання норм організації робочого місця, забезпечення комфортних та зручних умов праці осіб, які беруть участь у процесі розробки та експлуатації, а це вимагає дослідження та дотримання вимог з охорони праці.

В Україні розроблено й діють ряд нормативних документів, які визначають вимоги і правила щодо використання комп'ютерної техніки, приміщень з екранними пристроями та ін. Основним нормативним документом при використанні комп'ютерної техніки є НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Він регламентує, що приміщення для експлуатації комп'ютерної техніки повинно розміщуватися в північній або північно-східній частині будівлі. Площа одного робочого місця повинна становити щонайменше 6 м<sup>2</sup>, об'єм — щонайменше 20 м<sup>3</sup>, відстань між робочими столами — щонайменше 2,5 м у ряду і 1,2 м між рядами. Стіни приміщень потрібно фарбувати у пастельні тони з коефіцієнтом відбиття 0,5-0,6 [26].

З метою зменшення напруження очей потрібно, щоб відстань між краями сусідніх точок зображення на моніторі не перевищувала гранично оптимальний розмір літеро-цифрових знаків – 16-20, складних знаків – 35-

40. Оптимальні співвідношення параметрів літер і цифр такі: ширина знака – 0,75 їх висоти, товщина ліній при зворотному контрасті –  $1/6-1/8$ , відстань між знаками — 0,25-0,5 висоти знака, між словами – 0,75-1, між рядками – 0,5-1 [26].

Для профілактики загальної втоми і особливо зорового аналізатора важливе значення має організація режиму праці та відпочинку. Загальна тривалість робочого дня не повинна перевищувати 8 год. Частота і тривалість перерв залежать від типу та інтенсивності виконуваних робіт. Під час робіт, які виконуються з великим навантаженням, рекомендуються перерви на 10-15 хв. через кожну годину, а при неінтенсивній і монотонній роботі — на 10-15 хв. через кожні дві години. Кількість мікропауз (тривалістю до хвилини) потрібно регулювати індивідуально.

Зміст регламентованих перерв може бути різний: виробнича гімнастика (вправи для очей, гімнастика, спрямована на корекцію вимушеної робочої пози, поліпшення венозного кровообігу, часткову дисфункцію рухової активності), альтернативна допоміжна робота, приймання їжі тощо.

Для того, щоб особи, які займаються виявленням функціональних та нефункціональних вимог до комп'ютерних систем меншою мірою втомлювались і зберігали високий рівень працездатності, потрібно раціонально організувати їхні робочі місця. Зокрема, робоче місце має відповідати основним антропометричним даним людини. Крісло або стілець на робочому місці повинні мати висоту сидіння 40-50 см від рівня підлоги, а також відповідний кут нахилу спинки.

Монітори потрібно розміщувати на висоті рівня очей (висота від підлоги до нижнього краю екрана має становити 95-100 см) на відстані 60-70 см від оператора (відстань від краю столу — 50-70 см). Кут зору працюючого щодо екрану має дорівнювати  $10-20^\circ$ , але не більше  $40^\circ$ , кут між верхнім краєм монітора і рівнем очей користувача має становити менш як  $10^\circ$ . Найдоцільніше розміщувати екран перпендикулярно до лінії погляду

користувача. Кут нахилу екрана по вертикалі має становити  $0-30^\circ$  [26]. З цією метою сучасні монітори комплектують підставкою з поворотним кронштейном, що дає змогу регулювати кут нахилу монітора і горизонтально обертати його навколо вертикальної осі. Висоту екрана від поверхні підлоги регулюють змінюючи висоту робочої поверхні столу. Іноді монітори встановлюють на спеціальні підставки, що уможлиблює його переміщення у просторі у вертикальному та горизонтальному напрямках.

У приміщеннях, де виконуються роботи на ПК, повинно бути передбачене природне і загальне штучне освітлення. Робочі місця користувачів потрібно розміщувати так, щоб у поле зору не потрапляли вікна і освітлювальні прилади (монітори потрібно розміщувати під кутом  $90-105^\circ$  до вікон і на відстані 2,5-4 м від стін і віконних прорізів). У поле зору користувача не повинні потрапляти поверхні, що відбивають світло. Покриття столу має бути матовим з коефіцієнтом відбиття 0,25-0,4.

Для штучного освітлення приміщення рекомендується застосовувати світильники матового світла з розсіювачами, а спектральний склад ламп має наближатися до спектру сонячного світла (наприклад, люмінесцентні типу ЛБ). Оптимальна освітленість робочих місць — 400-500 лк.

У разі ураження електричним струмом необхідно терміново звільнити потерпілого від дії електричного струму (через відключення електроживлення в кімнаті, загального електроживлення на розподільному щиті або іншим способом). Викликати швидку медичну допомогу (подзвонивши за міським телефоном 103). Надати першу медичну допомогу потерпілому, враховуючи наступне:

- якщо потерпілий знепритомнів, але дихає, його необхідно рівно і зручно вкласти, розстебнути одяг, створити приплив свіжого повітря і забезпечити повний спокій;
- при відсутності ознак життя до прибуття лікарів потерпілому необхідно робити штучне дихання.

Бізнес аналітики та усі задіяні у процесі аналізу вимог до комп'ютерних систем при виконанні відповідних робіт несуть відповідальність за порушення вимог з охорони праці і правил техніки безпеки.

При дослідженні технологій виявлення функціональних та нефункціональних вимог до комп'ютерних систем було дотримано усіх вище наведених вимог нормативних документів щодо охорони праці і техніки безпеки при експлуатації комп'ютерної техніки.

4.2. Особливості роботи та розлади здоров'я користувачів комп'ютерів, що формується під впливом роботи за комп'ютером

У професійних операторів частіше зустрічаються порушення органів зору, опорно-рухового апарату, центральної нервової, серцево-судинної, імунної та статевої систем, захворювання шкіри. Зафіксована значна кількість скарг операторського персоналу на загальне недомагання, передчасне стомлювання, головний біль, порушення функцій органів зору, які здійснювали несприятливий психофізіологічний вплив на самопочуття та працездатність операторів [24].

Сучасна професія користувача ВДТ належить до розумової праці, яка характеризується:

- високою напруженістю зорових функцій;
- одноманітною позою;
- великою кількістю стереотипних висококоординованих рухів, що виконуються лише м'язами кистей рук на фоні малої загальної рухової активності;
- значним нервовоемоційним компонентом, особливо в умовах дефіциту часу;
- роботою з великими масивами інформації, що викликає активізацію уваги та інших вищих психічних функцій.

Крім того, при роботі з дисплеями на електронно-променевих трубках виникає вплив на користувача цілої низки факторів фізичної природи — електростатичні поля, радіочастотне та рентгенівське випромінювання тощо. Діяльність професіоналів можна поділити на три групи:

– діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження, наприклад, робота операторів комп'ютерного набору.

– діяльність, яка пов'язана із здійсненням логічних операцій, що постійно повторюються – робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва.

– діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму, наприклад, робота інженера-програміста, диспетчерів руху залізничного транспорту, аеропортів тощо.

У користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень досить часто (40 – 70%) виникають психологічні та поведінкові порушення (нервозність, роздратування, тривога, нерішучість, замкнутість тощо) [24].

Серед користувачів ВДТ в США і Європі значного поширення набуло специфічне захворювання, яке отримало назву синдром комп'ютерного стресу (СКС). СКС супроводжується головним болем, запаленням очей, алергією, роздратованістю, млявістю і депресією.

Інформаційне перевантаження користувачів ВДТ супроводжується низкою специфічних захворювань, які називають інформаційними. Першим симптомом їх є головний біль. Дослідження, проведені в США, Німеччині, Швейцарії та інших країнах, показали, що робота з обслуговування ВДТ супроводжується підвищеним напруженням зору, інтенсивністю і монотонністю праці, збільшенням статичних навантажень, нервово-психічним напруженням, впливом різного виду випромінювань та ін.

Внаслідок цього серед операторів ВДТ, як зазначають фахівці Всесвітньої організації охорони здоров'я, частіше, ніж в інших групах



працюючих, трапляються такі професійні захворювання, як передчасна стомлюваність, погіршення зору, м'язові і головні болі, психічні й нервові розлади, хвороби серцево-судинної системи, онкологічні захворювання та ін.

Вважається, що стан організму операторів ВДТ визначається комплексним впливом факторів трудового процесу і середовища, значення яких є неоднаковим. На операторів з малим стажем роботи на ВДТ домінуючий вплив чинять фактори середовища, а на операторів зі стажем понад 5 років - фактори трудового процесу.

Розлади здоров'я користувачів, що формуються під впливом роботи за комп'ютером.

Комп'ютерний зоровий синдром (КЗС) – комплекс порушень здоров'я, який може виникати у користувачів персональних комп'ютерів (ПК). Діагноз ставлять, якщо людина, що працює за ПК протягом двох годин, висловлює хоча б дві з десяти скарг: головний біль, слезотеча, різь, туман, двоїння, свербіж, важкість в очах, фотофобія, миготіння знаків на екрані, нудота. У користувачів ПК дуже поширені кон'юнктивіти і блефарити, патогенетично пов'язані з КЗС.

Синдром розвивається при умові, що робоче місце організовано неправильно – у користувача незручне крісло, відсутні попітри для паперів, підставки для ніг та кистей рук, не встановлена висота і нахил монітора відносно очей, відстань від очей до екрана. За таких умов тіло людини при роботі займає вимушене положення: спина статично напружена, шия витягнута, плечі жорстко фіксовані. Напружені м'язи погіршують кровотік у сонних артеріях, а недостатнє кровозабезпечення головного мозку веде до очманіння, появи головного болю.

Перенапруження скелетно-м'язової системи.

Діяльність користувачів комп'ютерів характеризується тривалою багатогодинною (8 год. і більше) працею в одноманітному напруженому сидячому положенні, малою руховою активністю при значних локальних

динамічних навантаженнях, що припадають лише на кисті рук. Такий характер роботи може призвести до появи низки хворобливих симптомів, що об'єднані загальною назвою — синдром довготривалих статичних навантажень (СДСН).

Узагальнюючи статистичні дані можна зробити висновок про те, що СДСН може проявлятися втомую, скутістю, болем, судомою, онімінням та ін., локалізуватися у різних 100 частинах тіла (шия, спина, руки, ноги та ін.) і виникати індивідуально з різною частотою (ніколи, рідко, епізодично, щоденно). Робоче положення "сидячи" забезпечується статичною працею значної кількості м'язів, що дуже втомлює. При такому положенні тіла м'язи ніг, плечей, шиї та рук довгий час перебувають у скороченому стані. Оскільки м'язи не розслабляються, в них погіршується кровообіг.

#### Ураження шкіри.

Частота шкірних уражень корелюється з низькою відносною вологістю на робочих місцях операторів та частим виникненням електростатичних зарядів. Електростатичне поле, яке генерується дисплеєм комп'ютера, посилює електростатичний заряд на тілі оператора, а відтак зростає електростатичне поле біля нього. Підвищення відносної вологості повітря у приміщенні в поєднанні з вилученням килимових покриттів, в яких нагромаджуються статичні заряди, сприяли зниженню шкірних висипань на обличчі. Обладнання заземлення, встановлення сіткового екрана з металевого дроту між дисплеєм і оператором у деяких випадках знижувало частоту захворювань шкіри.

#### Розлади центральної нервової системи (ЦНС).

Виробнича діяльність операторів ВДТ має свої особливості, під впливом яких можуть формуватись розлади здоров'я. До найважливіших факторів, характерних для роботи операторів ВДТ, що впливають на погіршення стану їх ЦНС належать: інформаційне перевантаження мозку в поєднанні з дефіцитом часу, тривожне очікування інформації, особливо тієї, що викликає необхідність прийняти рішення; велике зорове та нервово-

емоційне напруження; гіподинамія; монотомія; висока відповідальність за кінцевий результат. Під впливом цих факторів виникають зміни у співвідношенні процесів збудження та гальмування в корі головного мозку.

#### 4.3. Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань

Захищаючись від зовнішнього іонізуючого опромінювання при роботах із закритими джерелами випромінювання, тобто такими, які виключають можливість потрапляння радіоактивних речовин у навколишнє середовище, перш за все необхідно не допустити переопромінення працівників. Основним способами захисту від цього є:

- зменшення активності джерела, з яким контактують працівники під час конкретного технологічного процесу – досягається шляхом використання речовин із меншою активністю;

- зменшення часу контакту з джерелом випромінювання – досягається шляхом вдосконалення організації робіт і технологічного виробничого процесу та проведення попередніх тренінгів працівників;

- збільшення відстані між людиною і джерелом – використовується, як правило, при контакті з точковим джерелом випромінювання шляхом використання дистанційних універсальних маніпуляторів та інших автоматизованих пристроїв;

- розташування між людиною і джерелом захисного екрану (стаціонарного, пересувного, розбірного, настільного тощо), тобто пристрою, який зменшує інтенсивність випромінювання до безпечного рівня [24].

Для виготовлення екранів, а також для захисту працівників в стаціонарних спорудах, використовується бетон, чавун, сталь, алюміній, скло, свинець та інші матеріали.

Від дії рентгенівських променів застосовують екрани зі сталевого листа товщиною 0,5-1 мм або алюмінію товщиною 3 мм, спеціальної гуми. Оглядові вікна виконують з плексигласу товщиною 30 мм або з покритого оловом скла товщиною 9 мм. Для захисту шкіри від забруднень радіоактивними речовинами та запобігання їх попаданню всередину організму, захисту від альфа і бетавипромінювання передусім застосовуються засоби індивідуального захисту (ЗІЗ) від радіації. Отже, засоби захисту від радіації використовуються у тих випадках, коли інші заходи недостатньо ефективні: при переході через зони збільшеної інтенсивності випромінювання, при ремонтних та налагоджувальних роботах у аварійних ситуаціях, під час короткочасного контролю та при зміні інтенсивності опромінення. З урахуванням зазначеного прогнозу на території області може виникнути складна радіаційна обстановка наслідки якої вимагатимуть від органів виконавчої влади, органів місцевого самоврядування, суб'єктів господарювання, на які покладено виконання завдань щодо захисту населення і територій від надзвичайних ситуацій, оперативного реагування та дій [24].

Місцеві органи виконавчої влади, органи місцевого самоврядування, суб'єкти господарювання здійснюють для забезпечення захисту людей від впливу іонізуючих випромінювань наступні заходи:

- приймають згідно з законодавством України рішення щодо застосування на підвідомчій території заходів втручання у разі радіаційних аварій;
- організовують проведення в установленому порядку щорічні обстеження з метою оцінки стану захисту людини від впливу іонізуючих випромінювань та ведення екологічного паспорта підвідомчої території;
- здійснюють організаційне керівництво системою обліку та контролю доз опромінення населення на підвідомчій території;
- організовують контроль за виконанням заходів щодо захисту людини від впливу радіонуклідів, що містяться у будівельних матеріалах;

- затверджують відповідні плани щодо захисту населення від радіаційних аварій та їх наслідків;
- забезпечують постійну готовність засобів оповіщення населення на підвідомчій території про виникнення радіаційної аварії;
- організовують контроль за виконанням заходів щодо захисту населення від радіаційних аварій та їх наслідків;
- забезпечують населення, в місцях його проживання, інформацією щодо рівнів опромінення людини та заходів захисту від впливу іонізуючих випромінювань, що виконуються на підвідомчій території;
- розроблюють та впроваджують програми захисту людей від впливу іонізуючих випромінювання;
- здійснюють оповіщення населення у разі виникнення радіаційної аварії та інформування про рятувальні та профілактичні заходи у зв'язку з цим.

Для виконання вищезазначених заходів залучаються органи управління, сили і засоби обласної територіальної та функціональних підсистем єдиної державної системи цивільного захисту (далі – ЄДС ЦЗ), порядок дій яких визначено Планом реагування на надзвичайні ситуації, пов'язаних з викидом радіоактивних речовин.

Режими захисту робітників і службовців на суб'єктах господарювання вводяться в дію рішенням керівників об'єктів. Незалежно від місця розміщення суб'єкту господарювання (в населеному пункті або за його межами) на його території вводиться в дію свій режим захисту з урахуванням рівнів радіації, виміряних на об'єкті, і реального ступеню захисту працівників і службовців.

При виникненні комунальної радіаційної аварії окрім термінових робіт щодо стабілізації радіаційного стану (включаючи відновлення контролю над джерелом) місцеві органи виконавчої влади, органи місцевого самоврядування, суб'єкти господарювання одночасно здійснюють заходи, спрямовані на:

- зведення до мінімуму кількості осіб з населення, які зазнають аварійного опромінення;
- запобігання чи зниження індивідуальних і колективних доз опромінення населення;
- запобігання чи зниження рівнів радіоактивного забруднення продуктів харчування, питної води, сільськогосподарської сировини і сільгоспугідь, об'єктів довкілля (повітря, води, ґрунту, рослин тощо), а також будівель і споруд.

Для населення, робітників та службовців суб'єктів господарювання, які можуть потрапити в зону випадіння радіоактивних опадів, доцільно завчасно, виходячи з конкретних місцевих умов, розрахувати варіанти режимів радіаційного захисту [23]. З урахуванням вищезазначеного, режими радіаційного захисту вводяться в дію місцевими органами виконавчої влади, органами місцевого самоврядування, суб'єктами господарювання з метою захисту людей від впливу іонізуючого випромінювання у разі загрози або виникнення надзвичайних ситуацій, пов'язаних з радіаційними аваріями.

#### Висновок.

Проведено аналіз особливостей роботи користувачів комп'ютерів і встановлено, що при неправильній організації робочих місць, недотриманні вимог мікроклімату та при не коректній організації графіку робочого режиму відбуваються розлади здоров'я. Це призводить до розвитку, в першу чергу, синдрому комп'ютерного зору, проблем із м'язами, викривлення хребта і т.п. Тому важливо дотримуватися визначених гігієнічних норм і правил для зменшення негативного впливу комп'ютерної техніки. Забезпечення захисту працівників суб'єкта господарювання від іонізуючих випромінювань полягає у дотриманні встановлених на підприємстві вимог і правил, і впровадженні технологій зменшення рівня іонізуючого випромінювання. Дотримання таких правил дозволяє зменшити або нівелювати негативний вплив випромінювання на організм людини.

## ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

4. При дослідженні основних понять у сфері інженерії вимог визначено основні їх типи і класифікацію, які широко використовуються на практиці при реалізації комп'ютерних систем і встановлено, що актуальним завданням сучасного розвитку комп'ютерної інженерії є автоматизація та оптимізація процесів класифікації функціональних та нефункціональних вимог.

5. Проведено аналіз процесів і моделей інженерії вимог у результаті якого встановлено, що найбільш важливими з них є виявлення та управління вимогами, а для забезпечення ефективності їх виконання необхідно зменшити суб'єктивний вплив при визначенні груп функціональних та нефункціональних вимог.

6. Проаналізовано підходи щодо управління вимогами до комп'ютерних систем на етапах їх життєвого циклу і підтверджено важливу роль відповідних процесів, дій та активностей, оскільки вимоги є фундаментом для подальших стадій розробки комп'ютерних систем.

7. Досліджено сучасні алгоритми і методи кластерного аналізу у сфері опрацювання текстової інформації і встановлено, що найбільш оптимальним рішенням за точністю і оптимальністю використання ресурсів є алгоритми k-means, які доцільно застосовувати при кластеризації функціональних та нефункціональних вимог до комп'ютерних систем.

8. Запропоновано процедуру і метод агломеративної ієрархічної кластеризації для визначення груп функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу забезпечити автоматизацію та підвищити ефективність процесу визначення та аналізу вимог на відповідному етапі життєвого циклу.

9. Обґрунтовано методи перетворення текстових даних, якими представляються функціональні та нефункціональні вимоги до

комп'ютерних систем, у векторний простір семантичних ознак, що дало змогу в подальшому застосувати методи машинного навчання.

10. Запропоновано та обгрунтовано доцільність застосування метрик точності і повноти для визначення якості кластеризації вимог до комп'ютерних систем, а також введено метрику порівняння, яка дає змогу визначити відхилення кількості ідентифікованих кластерів, визначених за допомогою методу агломеративної ієрархічної кластеризації та фактично наявних кластерів.

11. Проведено аналіз та препроцесинг наборів даних, які містять функціональні та нефункціональні вимоги до комп'ютерних систем, що дало змогу провести їх очищення і в подальшому підвищити якість кластеризації вимог.

12. На практиці застосовано програмні методи векторизації текстових даних, що дозволило в подальшому визначити кластери в межах функціональних та нефункціональних вимог із використанням запропонованого методу ієрархічної кластеризації.

13. Запропоновано процедури і побудовано програмний конвеєр опрацювання функціональних та нефункціональних вимог до комп'ютерних систем, що дало змогу практично реалізувати технології машинного навчання в процесі їх аналізу тим самим скоротивши часові витрати на виконання проекту та підвищивши якість його виконання.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Requirements Engineering: Setting the Context. URL: [https://www.wohlin.eu/rm\\_chapter05.pdf](https://www.wohlin.eu/rm_chapter05.pdf) (дата звернення: 14.09.2023).
2. Kharchenko A., Galay I., Yatsyshyn V. The method of quality management software. 2011 Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH 2011 . Polyana. 2011. pp. 82-84.
3. Yatsyshyn V., Pastukh O., Lutskiy A., Tsymbalistyy V., Martsenko N. A Risks management method based on the quality requirements communication method in agile approaches / Information technologies: theoretical and applied problems, 2022, pp. 1-10.
4. Harchenko A., Bodnarchuk I., Yatsyshyn V. The modeling and optimization of software engineering processes. Modern Problems of Radio Engineering, Telecommunications and Computer Science - Proceedings of the 11th International Conference, TCSET'2012. Lviv - Slavske , 2012. p. 326.
5. Тиш Є.В., Шалапай Р.І. Типи вимог до комп'ютерних систем і методи їх виявлення. Матеріали XII міжнародної науково-практичної конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (6-7 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 437.
6. Тиш Є.В., Шалапай Р.І. Їєрархічна кластеризація для визначення сукупності функціональних та нефункціональних вимог комп'ютерних систем. Матеріали XI науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (13-14 грудня 2023 року). Тернопіль: ТНТУ. 2022. С. 193.
7. Yasniy O., Pastukh O., Didych I., Yatsyshyn V., Chykhira I. Application of machine learning for modeling of 6061-T651 aluminum alloy stress–strain diagram. Procedia Structural Integrity. 48. 2023. pp. 183–189.

8. Chan C.H., Sun A., Lim E.P. Automated online news classification with personalization. In Proceedings of the 4th International Conference of Asian Digital Libraries, Bangalore, India, 10–12 December 2001. 2001. p. 8.
9. Shirgave S., Awati C., More R., Patil S. A Review On Credit Card Fraud Detection Using Machine Learning. *Int. J. Sci. Technol. Res.* 2019. pp. 1217–1220.
10. Tiha A. Intelligent Chatbot Using Deep Learning. Ph.D. Thesis, University of Memphis, Memphis, TN, USA, 2018.
11. Abdul Salam M. Sentiment Analysis of Product Reviews Using Bag of Words and Bag of Concepts. *Int. J. Electron.* 2019. pp. 49–60.
12. Fei N., Zhang Y. Movie genre classification using TF-IDF and SVM. In Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City, Shanghai, China, 20–23 December. 2019. pp. 131–136.
13. Erkan A., Gungor T. Sentiment Analysis Using Averaged Weighted Word Vector Features. arXiv 2020, arXiv:2002.05606.
14. Forman G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *J. Mach. Learn. Res.* 2003. pp. 1289–1305.
15. Kotsiantis S. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* 2007. Pp. 249–269.
16. Vergara D., Hernández S., Jorquera F. Multinomial Naive Bayes for real-time gender recognition. In Proceedings of the 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA), Bucaramanga, Colombia, 31 August–2 September. 2016. pp. 1–6.
17. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design. *Scientific Journal of TNTU (Tern.)*. Vol. 109. No 1. 2023. pp. 54–65.
18. Pastukh O., Yatsyshyn V. Brain-computer interaction neurointerface based on artificial intelligence and its parallel programming using high-

performance calculation on cluster mobile devices. Scientific Journal of TNTU (Tern.). Vol 112. No 4. pp. 26–31.

19. Yatsyshyn V., Pastukh O., Zharovskyi R., Shabliy N. Software tool for productivity metrics measure of relational database management system. Mathematical Modeling. No 1 (48). 2023. P. 7-17.

20. Menzies, T., Krishna, R., Pryor, D. The Promise Repository of Empirical Software Engineering Data. URL: Available online: <http://openscience.us/repo/requirements/requirements-other/wasp.html> (дата звернення: 10.09.2023).

21. Mental Health Care Patient Management System. URL: <https://bscs143.files.wordpress.com/2015/11/requirement-mhc-pms.docx> (дата звернення 15.09.2023).

22. National Research Council of Italy. Natural Language Requirements Dataset. URL: <http://fmt.isti.cnr.it/nlreqdataset/> (дата звернення: 15.09.2023). Катренко Л.А., Катренко А.В. Охорона праці в галузі комп'ютерингу. Львів: Магнолія-2006. 2012. 544 с.

23. Желібо Є. Безпека життєдіяльності. К.: 2001. 483 с.

24. Стадник І.Я., Зварич Н.М. Оцінка хімічної обстановки при аваріях на хімічно небезпечних об'єктах викидом (виливом) небезпечних хімічних речовин та застосуванні хімічної зброї. ТНТУ. 2020. 36 С.

Додаток А

Тези конференцій

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
Тернопільський національний технічний університет імені Івана Пулюя (Україна)  
Університет імені П'єра і Марії Кюрі (Франція)  
Маріборський університет (Словенія)  
Технічний університет у Кошице (Словаччина)  
Вільнюський технічний університет ім. Гедімінаса (Литва)  
Міжнародний університет цивільної авіації (Марокко)  
Наукове товариство ім. Т.Шевченка

# **АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ**

**Збірник**  
тез доповідей

**ХІІ Міжнародної науково-практичної  
конференції молодих учених та студентів**  
6-7 грудня 2023 року



**УКРАЇНА**  
**ТЕРНОПІЛЬ – 2023**

38.	<b>Т. Крамар</b> ДЕЦЕНТРАЛІЗОВАНЕ АВТОМАТИЧНЕ ПІДКЛЮЧЕННЯ ПУНКТИВ НЕЗЛАМНОСТІ ПІД ЧАС ВІДКЛЮЧЕНЬ У ЗИМІ 2023 В ПРИФРОНТОВИХ ЗОНАХ УКРАЇНИ	415
39.	<b>Б. Б. Млинко, О. П. Стефанюк</b> АНАЛІЗ ВИКОРИСТАННЯ ІГРОВИХ РУШІВ ДЛЯ СТВОРЕННЯ ЦИФРОВИХ ДВІЙНИКІВ НА ОСНОВІ СИСТЕМНОГО ПІДХОДУ	417
40.	<b>Н. М. Коцюк, В. Д. Тимошук, Ю. О. Момоток, Н. С. Луцки</b> СИСТЕМА РЕЗЕРВУВАННЯ ТРАФІКУ НА ОСНОВІ МІКРОТІК	419
41.	<b>В. В. Василюшин, В. Д. Тимошук, Н. Ю. Кігчак, Н. С. Луцки</b> АНАЛІЗ ХАРАКТЕРИСТИК ТА ЗАСТОСУВАННЯ МІКРОКОНТРОЛЕРІВ ATTINY85, ATMEGA8, RP2040	420
42.	<b>А. М. Ковтко, Н. В. Лешук, І. Р. Козбур, І. В. Коноваленко</b> АНАЛІЗ ЕФЕКТИВНОСТІ СИСТЕМ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНИХ ПРОДУКТІВ	421
43.	<b>О. Ю. Замора, А. В. Немеришин, І. Р. Козбур, О. Р. Дмитрів</b> АНАЛІЗ МЕРЕЖЕВИХ СИСТЕМ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ З ВИКОРИСТАННЯМ ПРОТОКОЛІВ МНОЖИННОГО ДОСТУПУ	423
44.	<b>М. В. Дрогобицький, Н. С. Луцки, А. М. Паламар</b> КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО КОНТРОЛЮ РІВНЯ ШУМУ НАВКОЛИШНЬОГО СЕРЕДОВИЩА	425
45.	<b>І. В. Лялик, А. М. Паламар</b> КОМП'ЮТЕРНА СИСТЕМА ДИСТАНЦІЙНОГО КОНТРОЛЮ ІНТЕНСИВНОСТІ УЛЬТРАФІОЛЕТОВОГО ВИПРОМІНЮВАННЯ	426
46.	<b>А. М. Паламар, Д. С. Соєн, В. П. Волоський</b> КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ВІДДАЛЕНОГО СПОСТЕРЕЖЕННЯ ЗА РІВНЕМ НАСИЧЕННЯ КИСНЕМ КРОВІ ЛЮДИНИ	427
47.	<b>М. В. Криховецький</b> МЕТОДИ ВИЯВЛЕННЯ ДРОНІВ НА БАЗІ НЕЙРОННИХ МЕРЕЖ	428
48.	<b>Д. І. Муштин</b> МОБІЛЬНА МЕТЕОСТАНЦІЯ ДЛЯ ОБПРИСКУВАЧА	431
49.	<b>Л. Є. Мосій, І. В. Струтинська, Г. В. Козбур</b> РОЛЬ КОМП'ЮТЕРНО-ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ЦИФРОВІЙ ТРАНСФОРМАЦІЇ ЕКОНОМІКИ.	432
50.	<b>О. Є. Подвисоцький, Н. Б. Стадник</b> МЕТОДИ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ В РОЗУМНОМУ БУДИНКУ	435
51.	<b>А. М. Паламар, Р. О. Романчук</b> КОМП'ЮТЕРНА СИСТЕМА ДЛЯ ВІДДАЛЕНОГО КОНТРОЛЮ РІВНЯ ЗАБРУДНЕННЯ ПОВІТРЯ ПИЛОМ	436
52.	<b>Є. В. Тит, Р. І. Шалапай</b> ТИПИ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ І МЕТОДИ ЇХ ВИЯВЛЕННЯ	437
53.	<b>А. М. Луцків, С. В. Макогон</b> НЕЙРОМЕРЕЖЕВІ ПІДХОДИ ДО ПЕРЕТВОРЕННЯ ТЕКСТОВИХ ПОВІДОМЛЕНЬ В АУДИОПОТІК	438
54.	<b>В. В. Яцишин канд. І. М. Кучма</b> ПОБУДОВА ОНТОЛОГІЙ ЯК СПОСІБ ЕФЕКТИВНОГО	439

УДК 004.912

Є. В. Тиш канд. техн. наук, Р. І. Шалапай

(Тернопільський національний технічний університет імені Івана Пулюя)

## ТИПИ ВИМОГ ДО КОМП'ЮТЕРНИХ СИСТЕМ І МЕТОДИ ЇХ ВИЯВЛЕННЯ

Ie. V. Tysh PhD., R. I. Shalapai

### TYPES OF COMPUTER SYSTEMS REQUIREMENTS AND METHODS OF THEIR DETECTION

Для більш глибокого розуміння, якими властивостями і характеристиками володіють функціональні та нефункціональні вимоги необхідно дослідити принципи їх класифікації і фактори, які на це впливають. На рис. 1 проілюстровано фактори, які мають визначальний вплив на встановлення приналежності вимог до класу функціональних чи нефункціональних вимог. На основі аналізу бізнес вимог можна встановити частину необхідних функціональних вимог для користувачів, проте все ж вони висловлюють і свої власні потреби. Бізнес вимоги та вимоги користувачів формують базис функціональності, який фіксується у потенційних сценаріях імплементації комп'ютерної системи. Нефункціональні вимоги також впливають на сценарії виконання функцій комп'ютерною системою, зокрема через наявні у бізнес правила, які впливають також на функціональні вимоги та атрибути якості системи. Серед системних вимог, виділяють частину, яка має безпосереднє відношення до виконання функцій системою. Нефункціональні вимоги можуть містити обмеження та особливості взаємодії із зовнішніми програмно-апаратними комплексами через відповідні інтерфейси.

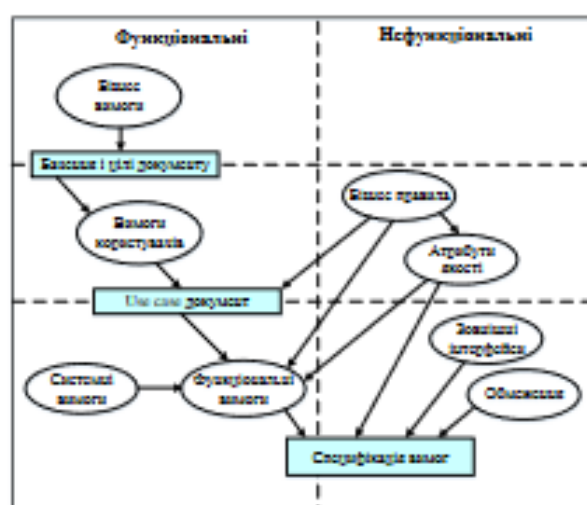


Рисунок 1. Фактори, які визначають і впливають на встановлення приналежності вимог до певного класу

Як видно з рис. 1, кластер функціональних вимог визначають в першу чергу бізнес вимоги, які формують бачення і цілі реалізації комп'ютерної системи. Розробку вимог часто тривіалізують як діяльність, яку виконують аналітики з добрих намірів, перш ніж вони почнуть виконувати справжню роботу зі специфікації продукту.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

XI НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



13-14 грудня 2023 року

ТЕРНОПІЛЬ  
2023

<b>Б.С.Таранів, О.Р.Цєбрів</b> РОЗРОБКА СКРИПТУ ДЛЯ ІНТЕГРАЦІЇ СКЛАДОВИХ ЕЛЕМЕНТІВ ІНЕРЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ «ARDUPILOT» <b>B.S.Tarasiv, O.R.Tsebriv</b> DEVELOPMENT OF A SCRIPT FOR INTEGRATION COMPONENT ELEMENTS OF THE INERTIAL SYSTEM USING «ARDUPILOT» TECHNOLOGIES	182
<b>М.В.Темениський</b> ПОРІВНЯННЯ БАЗ ДАНИХ MONGODB ТА POSTGRESQL В КОНЕКСТІ РОЗРОБКИ СУЧАСНИХ ВЕБ-ЗАСТОСУНКІВ <b>M.V.Temenytskyi</b> COMPARISON OF MONGODB AND POSTGRESQL DATABASES IN THE CONTEXT OF MODER WEB APPLICATIONS DEVELOPMENT	184
<b>В.Тимошчук, Т.Чех, А.Фіалка, Н.Луцук</b> МЕТОДИ ВІРТУАЛІЗАЦІЇ В КЛАСТЕРАХ ВИСОКОЇ ДОСТУПНОСТІ <b>V. Tymoshchuk, T. Chekh, A. Fialka, N. Lutsyk</b> METHODS OF VIRTUALIZATION IN HIGH AVAILABILITY CLUSTERS	186
<b>В.Тимошчук, В.Васильшин, І.Мудрий, Н.Луцук</b> ОГЛЯД ТА ПОРІВНЯННЯ ПРОТОКОЛІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ В IOT <b>V. Tymoshchuk, V. Vasylyshyn, I. Mudryi, N. Lutsyk</b> OVERVIEW AND COMPARISON OF INFORMATION TRANSFER PROTOCOLS IN IOT	188
<b>Ткачук Р.М., Ткачук Р.А.</b> ЗАБЕЗПЕЧЕННЯ ІНДИВІДУАЛЬНОГО ПІДБОРУ КЛАПАНІВ ДЛЯ ВИВОДУ ВНУТРІШНЬООЧНОЇ РІДИНИ ПРИ ЛІКУВАННІ ГЛАУКОМИ <b>Tkachuk R.M., Tkachuk R.A.</b> PROVISION OF INDIVIDUAL SELECTION OF VALVES FOR THE REMOVAL OF INTRAOCULAR FLUID IN THE TREATMENT OF GLAUCOMA	189
<b>Д.А.Урбан</b> СУЧАСНІ МОДЕЛІ ТА АЛГОРИТМИ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ДИНАМІКИ СОЦІАЛЬНИХ МЕДІА <b>D. A. Urban</b> MODERN MODELS AND ALGORITHMS FOR AUTOMATED ANALYSIS AND FORECASTING OF SOCIAL MEDIA DYNAMICS	190
<b>Лілія Хвостівська, Назар Паламар, Сергій Сторож</b> ПРОГРАМНИЙ ЗАСІБ ВЕЙВЛЕТ-ВИЯВЛЕННЯ РАДІОСИГНАЛІВ В МАТЕРИНЬСЬКОМУ БАЗИСІ МЕКСИКАНСЬКОГО КАПЕЛЮХА <b>Liliia Khvostivska, Nazar Palamar, Serhii Storozh</b> SOFTWARE WAVELET DETECTION OF RADIO SIGNALS IN THE MEXICAN HAT MOTHER BASE	191
<b>Д.Р.Чарковський, Н.Б.Стадник</b> МЕТОДИ ДЕТЕКТУВАННЯ ТЕКСТОВИХ ОБЛАСТЕЙ НА ЗОБРАЖЕННЯХ <b>D.R. Charkovskyi, N.B. Stadyuk</b> METHODS FOR DETECTION OF TEXT REGIONS IN IMAGES	192
<b>Євгенія Туш, Руслан Шалапай</b> ІЄРАРХІЧНА КЛАСТЕРИЗАЦІЯ ДЛЯ ВИЗНАЧЕННЯ СУКУПНОСТІ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ КОМП'ЮТЕРНИХ СИСТЕМ <b>Ievhenia Tush, Ruslan Shalapaу</b> HIERARCHICAL CLUSTERIZATION FOR DETERMINING FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS OF COMPUTER SYSTEMS	193



УДК 004.94

Євгенія Тиш канд. техн. наук, Руслан Шалапай

Тернопільський національний технічний університет імені Івана Пулюя

## ІЄРАРХІЧНА КЛАСТЕРИЗАЦІЯ ДЛЯ ВИЗНАЧЕННЯ СУКУПНОСТІ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ КОМП'ЮТЕРНИХ СИСТЕМ

Ievhenia Tysh PhD., Ruslan Shalapai

### HIERARCHICAL CLUSTERIZATION FOR DETERMINING FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS OF COMPUTER SYSTEMS

Ієрархічна кластеризація, як впливає з назви – це алгоритм, який формує ієрархію кластерів. Цей алгоритм починається з усіх елементів даних, які розглядаються як окремі кластери. Після цього два найближчі кластери об'єднуються в один. Зрештою, цей алгоритм завершується, коли залишається лише один кластер. Результати ієрархічної кластеризації можна показати за допомогою дендрограми. Дендрограму можна інтерпретувати як показано на рис. 1.

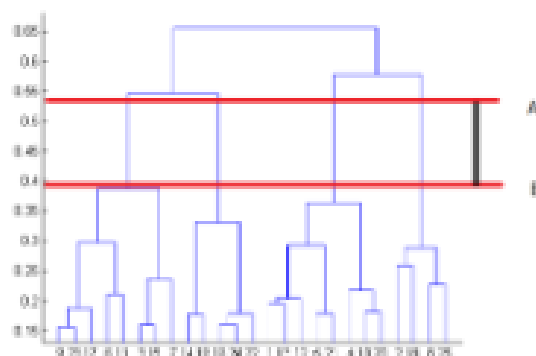


Рис. 1. Формування кластерів на основі відстані між ними

На рис. 1 показано вхідні елементи даних у вигляді 25 точок, кожна з яких представляє окремий кластер. Потім два найближчі кластери об'єднуються, доки не залишиться лише один кластер у верхній частині.

Бисота на дендрограмі, на якій зливаються два кластери, представляє відстань між двома кластерами в просторі даних. Рішення щодо кількості кластерів, які найкраще відображають різні групи, можна вибрати, аналізуючи дендрограму. Найкращий вибір кількості кластерів визначається кількістю вертикальних ліній у дендрограмі, розрізаних горизонтальною лінією, яка може перетинати максимальну відстань по вертикалі, не перетинаючи кластер.

У наведеному на рис. 1 прикладі, найкращий вибір кількості кластерів буде 4, оскільки червона горизонтальна лінія на дендрограмі нижче покриває максимальну вертикальну відстань АВ.

Дві важливі речі, які потрібно знати про ієрархічну кластеризацію. Цей алгоритм реалізовано з використанням підходу знизу вгору. Також можна застосувати низхідний підхід, починаючи з того, що усі точки даних, задають один кластер.