

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ)

Факультет інформаційних технологій

ПОГОДЖЕНО

Декан факультету (Директор ННІ)

інформаційних технологій

(назва факультету (ННІ))

Ігор БОЛБОТ

(підпис)

(ім'я ПРІЗВИЩЕ)

“ ___ ” _____ 20__ р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри

комп'ютерних наук

(назва кафедри)

Белла ГОЛУБ

(підпис)

(ім'я ПРІЗВИЩЕ)

“ ___ ” _____ 20__ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Програмне забезпечення оцінки не функціональних вимог до програмного проекту

Спеціальність 121 «Інженерія програмного забезпечення»

(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

к. ф.-м. н., доцент

(науковий ступінь та вчене звання)

Віктор КИРИЧЕНКО

(підпис)

(ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

к.е.н., ст.викладач

(науковий ступінь та вчене звання)

Дмитро НІКОЛАЄНКО

(підпис)

(ім'я ПРІЗВИЩЕ)

Виконав

(підпис)

Дмитро МИСЛИВИЙ

(ім'я ПРІЗВИЩЕ здобувача)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ) Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук
К.Т.Н., доцент. ГОЛУБ Б. Л.
(науковий ступінь, вчене звання) (підпис) (ПІБ)
“ ” 20 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Мисливому Дмитру Ігоровичу

(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

(код і назва)

Освітня програма Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Програмне забезпечення оцінки не функціональних вимог до програмного проекту

затверджена наказом ректора НУБіП України від “ 01 ” листопада 2024р. №1963 «С»

Термін подання завершеної роботи на кафедру 20.11.2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: набори даних, що містять нефункціональні вимоги до програмного забезпечення, зокрема показники зручності використання, а також відгуки й оцінки користувачів, отримані з опитувань та відкритих джерел.

Перелік питань, що підлягають дослідженню:

1. Дослідити фактори, що впливають на результативність виконання завдань користувачами, та визначити основні перешкоди у взаємодії із системою.
2. Визначення стандартів і методичних підходів до оцінки usability (ISO/IEC 25010, ISO 9241 та інші).
3. На основі зібраних даних сформулювати нефункціональні вимоги для покращення зручності використання програмних продуктів.
4. Проведення експериментального дослідження та аналіз результатів оцінки зручності використання.
5. Визначити найкращі практики управління нефункціональними вимогами, зокрема у сфері usability.

Перелік графічного матеріалу (за потреби):

1. Модель вимірювання зручності використання.
2. Класифікація методів оцінки зручності використання.
3. Етап збору даних.
4. Архітектура програмної системи

Дата видачі завдання “ 01 ” листопада 2024 р.

Керівник магістерської кваліфікаційної роботи

НИКОЛАЄНКО Д. В.

(підпис)

(прізвище та ініціали)

Завдання прийняв до виконання

МИСЛИВИЙ Д. І.

(підпис)

(прізвище та ініціали студента)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ I ЗАБЕЗПЕЧЕННЯ ЗРУЧНОСТІ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ.....	10
1.1 Поняття функціональних і нефункціональних вимог та опис.....	10
1.2 Поняття зручності використання програмного продукту	11
1.3 Вимірювання та оцінка зручності використання програмних продуктів.....	14
1.4 Управління зручністю використання програмних продуктів.....	20
РОЗДІЛ II МЕТОД УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ.....	25
2.1 Метод управління зручністю використання програмних продуктів	25
2.2 Кореляційно-регресійний аналіз показників зручності використання.....	29
2.3 Кількісний та якісний аналіз узгодженості експертів	33
2.4 Визначення ваги метрик, показників та атрибутів ЗВ.....	37
РОЗДІЛ III ПРОГРАМНА СИСТЕМА УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ.....	41
3.1 Функціональні вимоги до програмної системи та вибір архітектурного стилю.....	41
3.2 Основні характеристики та варіанти використання системи	44
3.3 Підсистеми аналізатора.....	52
РОЗДІЛ IV ЗАСТОСУВАННЯ ПРОГРАМНОЇ СИСТЕМИ УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ.....	58
4.1 Вибір критеріїв зручності використання та опис досліджуваного програмного продукту	58
4.2 Емпірична оцінка та варіанти забезпечення ЗВ	62
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ	74

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПП – програмні продукти

ЗВ – зручність використання

ПС – програмна система

ПЗ – програмне забезпечення

ОС – операційна система

БД – база даних

ПО – предметна область

ІС – інформаційна система

UML (Unified Modeling Language) – уніфікована мова моделювання

GUI (Graphical user interface) – графічний інтерфейс користувача

ООП – об'єктно-орієнтоване програмування

СД – сховище даних

ВСТУП

Зручність використання (usability) є однією з ключових характеристик будь-якого продукту, орієнтованого на взаємодію з людиною. Це повною мірою стосується і ПЗ. Рівень зручності використання ПЗ визначає, з одного боку, обсяг ресурсів (зокрема, часу та зусиль), необхідних для досягнення поставлених цілей із належною точністю та повнотою, що безпосередньо впливає на продуктивність користувача. З іншого боку, цей показник є важливим чинником конкурентоспроможності ПП, адже впливає на рішення користувачів щодо його вибору. Таким чином, забезпечення зручності використання є актуальним завданням як для користувачів, так і для розробників. Її значущість підкреслюється тим, що даний показник включено до моделі якості ПЗ і представлений у міжнародних стандартах ISO/IEC 25010:2011, ISO 9241-11 та ISO/IEC 25060:2010.

Проблематику ЗВ досліджували багато іноземних науковців, серед яких N. Bevan, B. Boehm, A. Cooper, Sh. Laskowski, J. McCall, J. Nielsen, A. Holzinger, J. Scholtz. Серед українських учених питання ЗВ розглядаються опосередковано, через категорію якості ПЗ, у працях Ф. Андона, Г. Ковалю, Б. Конорева, Т. Коротун, К. Лаврищевої, В. Сулова, І. Туркіна, О. Харченка.

З огляду на важливість ЗВ, необхідно забезпечувати її досягнення та підтримку протягом життєвого циклу ПЗ. Для цього під час розроблення та супроводу продукту виконуються спеціальні дії, що потребують ефективного управління. Оскільки ЗВ є складовою загальної якості ПЗ, процес управління нею інтегровано у систему управління якістю. Водночас досягнення цього показника ускладнюється низкою факторів: різницею у сприйнятті зручності між користувачами та розробниками; необхідністю оптимізації трудовитрат; а також браком методів управління ЗВ в сучасних технологіях розробки.

Розбіжності між уявленнями розробників і користувачів щодо ЗВ зумовлені суб'єктивним характером цього поняття. Незважаючи на конкретність функцій ПП, способи подання вхідних і вихідних даних можуть значно

відрізнятись. Важливо враховувати не лише естетичний аспект інтерфейсу, а й легкість взаємодії з системою, доступність функцій, можливість контролю над процесами та досягнення цілей користувача з ефективністю й задоволеністю. При цьому користувачі не завжди усвідомлюють різницю між бажаним і дійсно зручним інтерфейсом. Тому під час управління ЗВ необхідно приймати обґрунтовані рішення щодо зміни конкретних показників, базуючись на оцінках реальних користувачів.

Оптимізація процесу досягнення ЗВ пов'язана з потребою уникнути перевищення трудовитрат над економічним ефектом від поліпшень. У процесі створення або модернізації ПЗ часто існує кілька можливих варіантів удосконалення інтерфейсу чи взаємодії, кожен із яких може позитивно вплинути на користувацьке сприйняття. Тому важливо визначити такий варіант, який забезпечить максимальне поліпшення при мінімальних витратах, що безпосередньо впливає на загальну вартість розроблення.

Недостатня інтеграція методів управління ЗВ до процесів життєвого циклу ПЗ, особливо в умовах ітераційних моделей розробки, створює додаткові труднощі. Це часто перешкоджає своєчасному виявленню проблем і призводить до зростання витрат на їх усунення на пізніших етапах.

Наявні методики здебільшого охоплюють окремі процеси – планування або контроль ЗВ, тому особливо актуальним є створення комплексного підходу до управління цим показником. Саме розробці програмного засобу управління зручністю використання присвячена дана магістерська робота.

Метою дослідження є вирішення науково-технічного завдання управління ЗВ у процесі створення та супроводу програмних продуктів шляхом розроблення відповідного методу, реалізованого у вигляді програмного засобу.

Для досягнення поставленої мети в роботі сформульовано й розв'язано такі завдання:

- дослідити предметну галузь і проаналізувати існуючі методи забезпечення зручності використання;

- розробити метод управління зручністю використання ПЗ, який дозволяє досягти рівня, що відповідає очікуванням користувачів;
- створити методику дослідження узгодженості оцінок зручності, отриманих від користувачів та експертів;
- поставити та вирішити оптимізаційну задачу мінімізації трудомісткості робіт для досягнення заданого рівня зручності;
- розробити методику визначення наявності та типу залежності між показниками зручності використання;
- спроектувати архітектуру та прототип ПС управління ЗВ;

Об'єкт дослідження – процеси управління зручністю використання програмних продуктів.

Предмет дослідження – моделі, методи і засоби управління ЗВ.

Методи дослідження. Для реалізації поставлених цілей застосовано такі наукові підходи: аналіз і синтез – під час вивчення існуючих методів забезпечення ЗВ; аналогію, формалізацію, моделювання, згортку та багатокритеріальну оптимізацію – при побудові математичних моделей; статистичні методи аналізу даних і виявлення залежностей – під час оптимізації чисельних рішень моделі; об'єктно-орієнтований аналіз і програмування – для проектування програмного засобу; анкетування та експертне оцінювання – для емпіричного підтвердження результатів.

Деякі організації, окрім загальноновизнаних стандартів в галузі якості ПЗ та забезпечення ЗВ, використовують власні *керівництва* зі зручності використання. Вони розробляються шляхом аналізу експертами великої кількості результатів досліджень, які часом протирічать один одному. Написання рекомендацій щодо створення ПП залежить, зокрема, від цільового призначення, типу функціональних користувачів та середовища, в якому даний ПП буде використовуватися. Керівництва з використання в основному орієнтовані на статичний вигляд інтерфейсу користувача. Аспекти дизайну оцінюються відособлено один від одного та безвідносно до контексту використання під час виконання завдання в реальному середовищі. З метою вирішення цієї проблеми

було розроблено *метод когнітивного аналізу*. Для проведення такого аналізу необхідно мати детальний опис інтерфейсу користувача, завдання, дані щодо демографічного стану функціональних користувачів, дані щодо контексту використання ПП, дані щодо успішної послідовності дій, які мають виконуватися. Спочатку виконується аналіз ПП, потім, згідно з визначеними для завдання діями, оцінюється, наскільки користувач буде здатний вибрати потрібну дію, зможе визначити, що вибрана дія вірна і чи буде помітно прогрес у вирішенні завдання. Аналіз ЗВ ґрунтується на інформації про те, що має знати користувач перед виконанням завдання та що він має засвоїти під час його виконання.

Наукова новизна роботи полягає в розробці нових, уточненні і доповненні наявних науково-методичних положень та практичних рекомендацій з управління зручністю використання програмних продуктів: *вперше*:

- на основі процесного підходу створено метод управління ЗВ, який охоплює планування, контроль, забезпечення та управління змінами, використовуючи формальний апарат теорії прийняття рішень. Застосування цього методу дозволяє здійснювати управління з урахуванням ресурсних обмежень, зокрема трудомісткості;

- розроблено математичну модель забезпечення ЗВ, отриману з моделі оцінки шляхом уведення цільової функції ресурсу та обмежень, що дає змогу визначати оптимальні коригувальні дії у випадку невідповідності встановленим вимогам;

подальшого розвитку набуло:

- вирішення задачі контролю ЗВ шляхом побудови моделі оцінки на основі методу вкладених скалярних згорток, що забезпечує ефективне відстеження відповідності встановленим вимогам.

Практичне значення отриманих результатів полягає в тому, що впровадження запропонованих моделей та програмного засобу у процес розроблення ПЗ підвищує ефективність управління проектами за критерієм ЗВ

та сприяє зростанню конкурентоспроможності ПП. На базі створеного методу та моделей спроектовано та реалізовано програмну систему Управління зручністю використання програмних продуктів.

РОЗДІЛ І

ЗАБЕЗПЕЧЕННЯ ЗРУЧНОСТІ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ

1.1 Поняття функціональних і нефункціональних вимог та опис

У процесі створення нової інформаційної системи або під час адаптації вже наявного рішення розробники неминуче стикаються з потребою формулювання вимог до майбутнього продукту. Тому доцільно детально проаналізувати, що являють собою функціональні та нефункціональні вимоги, яку роль вони відіграють і як саме визначаються у професійній практиці.

Незважаючи на велику кількість підходів і рекомендацій щодо опису властивостей ПЗ чи інформаційних систем, усі вимоги традиційно поділяють на дві основні групи: **функціональні** та **нефункціональні**. Передусім важливо окреслити різницю між ними.

Функціональні вимоги визначають, які саме можливості та дії повинна забезпечувати система. Вони описують поведінку продукту, перелік його функцій та очікувану взаємодію користувачів із системою.

Нефункціональні вимоги, навпаки, характеризують умови, в яких система має працювати, та її якісні параметри. Вони уточнюють властивості ПЗ, що впливають на зручність, ефективність та стабільність його використання. Фактично нефункціональні вимоги задають критерії, за якими оцінюється якість роботи ПЗ.[4, с. 38]

Після розмежування понять варто розглянути, що саме входить до групи нефункціональних вимог і які аспекти вони охоплюють. У загальному вигляді до них відносять якісні характеристики продукту, такі як надійність, продуктивність, масштабованість, безпека та інші властивості, що визначають рівень зрілості та ефективності розробки. Разом із тим нефункціональні вимоги охоплюють і низку додаткових аспектів:

- **Обмеження.** Тобто умови, що обмежують вибір будь-яких рішень щодо втілення в життя окремих вимог (або наборів вимог). Вони звужують

різноманітність вибору інструментів, стратегій, засобів при розробці структури (архітектури), так і зовнішнього вигляду інформаційного якого ПП.

- **Бізнес-правила.** Сюди відносяться керівні правила, політика, принципи, положення, якимось обмежують певні аспекти бізнесу. Наприклад, вони можуть визначати склад і правила виконання будь-яких бізнес-проектів.

- **Пропозиції щодо реалізації.** У групу входять конкретні пропозиції, які оцінюють можливість застосування певних архітектурних і технологічних рішень.

- **Зовнішні інтерфейси.** Опис ключових аспектів взаємодії продукту з іншими системами та операційної середовищем. Насамперед, це вимоги до API системи або продукту, а також вимоги до API інших систем, з якими планується інтеграція розроблюваного продукту.

- **Пропозиції з перевірки, тестування розроблюваного ПЗ.** Це ряд доповнень до вимогам, що вказує, як те чи інше вимога повинно бути перевірено на практиці.

- **Юридичні вимоги.** До ліцензування продукту, наявності патенту та ін. Слід наголосити, що нефункціональні вимоги визначаються та фіксуються на початкових етапах розробки, оскільки саме вони формують основу для подальшого проектування та реалізації ПП. [1, с. 341-342]

1.2 Поняття зручності використання програмного продукту

У науковій літературі існує чимало спроб сформулювати визначення зручності використання ПЗ, проте ці трактування часто відрізняються або суперечать одне одному. Тому в даному дослідженні доцільно скористатися визначеннями, закріпленими в міжнародних стандартах ISO 9241-11 та ISO/IEC 25010:2011 (який прийшов на зміну ISO/IEC 9126-1:2001) [18].

Відповідно до цих стандартів, **зручність використання (usability)** розуміють як рівень, до якого ПП може бути ефективно, економно та із задоволенням використаний визначеною групою користувачів для досягнення конкретних цілей у певному контексті застосування.

У стандарті ISO/IEC 25010:2011, що входить до сімейства SQuaRE (ISO/IEC 25000 – ISO/IEC 25099) [20], зручність використання розглядається з двох точок зору:

1. у моделі якості ПП (**product quality model**) – як одна з його характеристик;
2. у моделі якості у використанні (**quality in use model**) – опосередковано, через результативність взаємодії користувача з продуктом.

У моделі якості ПП визначено вісім ключових характеристик:

- функціональна придатність (*functional suitability*),
- продуктивність (*performance efficiency*),
- сумісність (*compatibility*),
- зручність використання (*usability*),
- надійність (*reliability*),
- безпека (*security*),
- супроводжуваність (*maintainability*),
- переносимість (*portability*).

Ця модель охоплює властивості, які можна оцінити кількісно під час розробки ПЗ, а також у процесі його експлуатації на конкретній операційній системі або апаратній платформі.

У стандарті ISO/IEC 25010:2011 [20] для опису підхарактеристик ЗВ застосовується формулювання «продукт або система». Оскільки система розглядається як продукт або як набір сервісів, що нею надаються, у подальшому тексті використовується узагальнений термін «продукт» – ПП у складі програм, процедур, документації та відповідних даних.

Згідно зі стандартом, зручність використання включає шість підхарактеристик:

- *розпізнавання сумісності* (*appropriateness recognisability*) – ступінь розпізнавання користувачами сумісності продукту з їх потребами;
- *можливість навчання* (*learnability*) – ступінь, у якому продукт може бути використано визначеними користувачами для досягнення поставлених

цілей в навчанні використовувати продукт з економічністю, ефективністю, вільною від ризику та задоволеністю у визначеному контексті використання;

- *операбельність* (operability) – ступінь, у якому продукт має властивості, що дозволяють легко оперувати ним та здійснювати контроль;
- *захист від помилок користувача* (user error protection) – ступінь, у якому продукт захищає користувача від помилок;
- *естетика користувацького інтерфейсу* (user interface aesthetics) – ступінь, у якому користувацький інтерфейс надає зручну та якісну та приємну взаємодію для користувача;
- *доступність* (accessibility) – ступінь, у якому продукт може бути використаний людьми з різноманітними здібностями та можливостями для досягнення поставленої мети у визначеному контексті використання. [5, с. 272]

Друга модель, представлена в ISO 9241-11, включає п'ять характеристик:

- **ефективність** – точність і змістовність, з якими користувач досягає поставлених цілей;
- **економічність** – витрачені ресурси відносно точності та змістовності, з якими користувач досяг поставлених цілей;
- **задоволеність** – ступінь задоволеності користувацьких потреб при використанні ПП у визначеному контексті.

Особливістю є те, що «задоволеність» має наступні підхарактеристики:

- **корисність (usefulness)** – задоволення від отриманих користувачем результатів;
- **довіра (trust)** – упевненість користувача в передбачуваності поведінки продукту;
- **приємність (pleasure)** – ступінь задоволення користувача, завдяки тому, що продукт відповідає його особистим та індивідуальним потребам;
- **комфорт (comfort)** – фізичний комфорт користувача під час використання продукту. [14, с. 32]

Показники якості та підхарактеристики можуть виражатися через зовнішні міри – значення, що демонструють, наскільки ПП забезпечує необхідні можливості у конкретних умовах експлуатації. Приклади таких мір наведено у стандарті ISO/IEC TR 9126-2 [19], який згодом планується замінити на ISO/IEC 25022.

Крім того, для кожної підхарактеристики визначаються внутрішні статичні властивості, вимірювані безпосередньо в процесі розробки ПЗ. Приклади таких мір подано в стандарті ISO/IEC 25023.

На сьогоднішній день існує перелік вітчизняних державних стандартів, які гармонізовані, серед них базовим вважають – ДСТУ 2850-94 [21]. Основні поняття про якість ПЗ та її складові характеристики також представлені у ДСТУ 2844-94 [21], який описує ЗВ як «сукупність властивостей ПЗ, що забезпечують користувачеві належні умови для ефективної роботи з ним.»

1.3 Вимірювання та оцінка зручності використання програмних продуктів

Процес вимірювання та подальшого оцінювання ЗВ є ключовим елементом забезпечення високої якості ПП. Як зазначалось раніше, ЗВ складається з низки взаємопов'язаних підхарактеристик, які формують основу для встановлення вимог до продукту та визначення рівня їх досягнення. Кожній підхарактеристиці відповідає певний набір властивостей ПП, для яких визначаються конкретні міри, значення таких мір отримують шляхом застосування спеціальних функцій вимірювання до відповідних елементів мір (рис. 1.1).

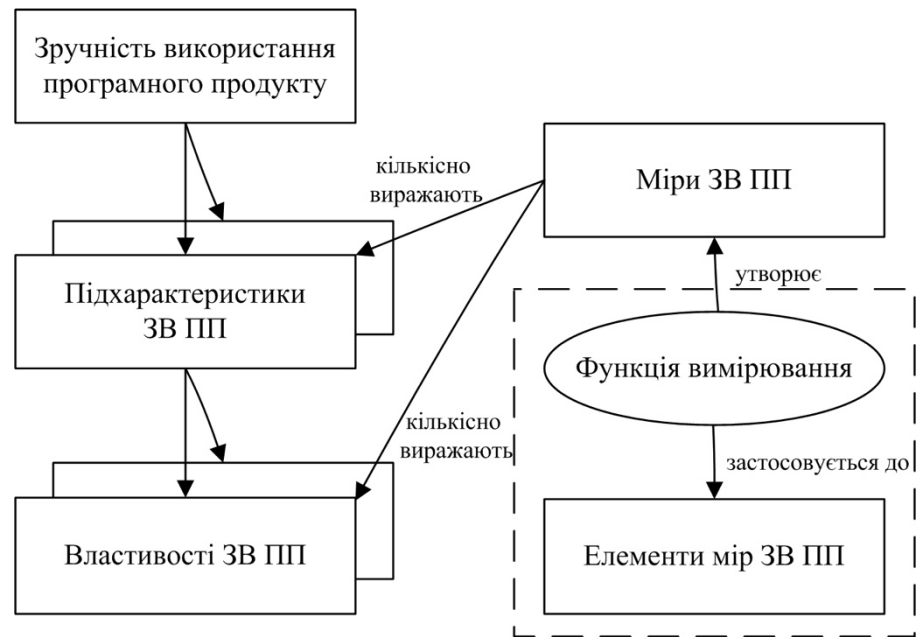


Рис. 1.1 Модель вимірювання ЗВ

Після отримання числових значень їх аналізують за допомогою різних методів оцінювання. У стандарті ISO/IEC 25040 під оцінкою розуміється впорядкований процес визначення рівня відповідності певного об'єкта встановленим критеріям. Виходячи з твердження наведеного вище стандарту можна сформулювати таке визначення:

Оцінка ЗВ – це процес визначення фактичних характеристик ПП щодо бажаного стану з погляду можливості досягнення користувачами певних цілей ефективно, результативно та із задоволеністю в конкретному контексті використання. Результатом такого процесу є числовий показник або набір показників.

Під *станом* ПП розуміємо множину кількісно виражених властивостей, які визначають поведінку ПП.

Можна сказати, що оцінювання ЗВ охоплює всі підходи й методики, які допомагають виявити шляхи підвищення зручності взаємодії з ПП, зрозуміти проблеми, що виникають у різних умовах використання, а також визначити обмеження, які можуть впливати на роботу користувачів. [16, с. 99]

Історичний розвиток методів оцінювання ЗВ охоплює приблизно шість десятиліть, що схематично подано на рис. 1.2.



Рис. 1.2 Розвиток підходів до оцінки ЗВ

Залежно від того, звідки походять дані для оцінювання, усі методи поділяють на три великі групи (рис. 1.3):

- оцінка, що ґрунтуються на відгуках користувачів,
- експертна оцінка,
- оцінка на основі моделей.

Усі вони застосовуються під час розробки або супроводу ПП та виконуються спеціалістами у сфері ЗВ.



Рис. 1.3 Класифікація методів оцінки ЗВ залежно від джерела вхідних даних

Оцінка на основі відгуків користувачів. Найстарішим джерелом оцінки ЗВ є відгуки користувачів. Вона здійснюється шляхом визначення типових

користувачів, завдань та створення процедур для висвітлення проблем, які можуть виникати під час використання певного програмного ПП при виконанні тих чи інших завдань. Протягом етапів проектування/тестування/розробки програмного забезпечення проводиться два види оцінки: формуюча та підсумкова. Формуюча оцінка використовується для аналізу отриманої від користувачів інформації і подальшого її використання в проектуванні. Результати підсумкової оцінки використовуються для визначення ефективності, раціональності та рівня задоволеності користувача програмним продуктом. Проведення обох видів оцінки є важливим для забезпечення людино-орієнтованої розробки. [7, с. 105-106]

Формуюча оцінка. Направлена на отримання перших відгуків користувачів щодо первинних концепцій ПП, прототипів інтерфейсу або ескізів екранних форм. Основним джерелом даних виступають вербальні (словесні) коментарі користувачів. Анкетування та бесіди дозволяють зрозуміти причини непорозумінь і складнощів у взаємодії з інтерфейсом.

Підсумкова оцінка. Має формалізованіший характер і спрямована на отримання числових значень підхарактеристик ЗВ. Зазвичай залучають 5-7 представників одного типу користувачів. Успішність оцінки залежить від якісної підготовки тестових сценаріїв та умов, наближених до реального контекста використання.

Переваги та недоліки оцінки ЗВ на основі відгуків користувачів.

Переваги:

- користувачі безпосередньо впливають на вдосконалення ПП;
- результати добре відображають реальні проблеми взаємодії;
- підсумкове оцінювання дає можливість керувати показниками ЗВ під час розробки.

Недоліки:

- значні часові та фінансові витрати;
- труднощі в пошуку представників цільової аудиторії;
- потреба у спеціальному обладнанні й професійних аналітиках.

Експерти як джерело оцінки ЗВ. Починаючи з другої половини 1990-х років у сфері оцінювання ЗВ активно поширюються підходи, що ґрунтуються на експертних оцінках. До найбільш застосовуваних на практиці належать аналіз стандартів і керівництв, когнітивне (мисленнєве) проходження та евристичне оцінювання.

Аналіз стандартів та керівництв.

Різні організації створюють власні настанови або використовують загальноприйняті стандартизовані рекомендації щодо забезпечення ЗВ у ПЗ. Зазвичай такі документи зосереджені на статичних особливостях GUI, тобто структурі та оформленні інтерфейсу, і рідко враховують реальний контекст використання ПП користувачем.

Когнітивне проходження.

Цей метод спрямований на визначення того, наскільки користувач здатен логічно вибудувати та виконати правильну послідовність дій для досягнення певної мети. Для його проведення необхідно мати опис інтерфейсу, сценарії дій, інформацію про типових користувачів та умови використання ПП. Далі кожен крок взаємодії аналізується з точки зору його зрозумілості та передбачуваності для користувача. [5, с. 271]

Евристичне оцінювання.

Цей підхід вважається одним із найбільш дієвих методів тестування ЗВ. Група експертів незалежно один від одного оцінює ПП на відповідність визначеним евристикам. З прикладами таких евристик можна ознайомитися в табл. 1.1. Для комплексних ПП рекомендується залучати приблизно 5 експертів, а для середніх і малих проєктів – не менше трьох оцінювачів, що забезпечує достатню точність та об'єктивність результатів.

Таблиця 1.1

Приклади евристик

№ з/п	Назва евристики
1.	Можливість дізнаватися статус системи (програмного продукту)
2.	Рівень відповідності системи (програмного продукту) вимогам реального світу
3.	Самостійність, тобто наскільки самостійно користувач може керувати системою (програмним продуктом)
4.	Узгодженість
5.	Здатність передбачати виникнення помилок
6.	Гнучкість та ефективність використання
7.	Естетичність та оформлення
8.	Можливість допомоги користувачам в розпізнаванні та діагностуванні помилок
9.	Наявність документації та довідкової інформації

Переваги та недоліки експертного оцінювання ЗВ.

Переваги:

- значно менші витрати, ніж у методах, заснованих на користувачах;
- можливість проводити оцінювання ще до створення прототипу;
- ефективність у виявленні значної частини потенційних проблем (до 50%).

Недоліки:

- залежність результатів від досвіду експертів;
- імовірність пропуску проблем, які проявляються тільки під час реальної роботи користувача.

Оцінка на основі моделей. Цей підхід ґрунтується на застосуванні спеціалізованих програмних засобів, що реалізують певні моделі ЗВ. Зазвичай такі інструменти оцінюють окремі аспекти ЗВ, а не всю її сукупність.

Найпоширеніші моделі й засоби:

- **GOMS (Goals, Operators, Methods, Selection rules) [20]** – модель, яка описує процес досягнення користувачем своїх цілей шляхом виконання певних операцій і правил вибору. Використовується для оцінювання процедурних аспектів ЗВ.

- **EPIC (Executive-Process/Interactive Control) [21]** – імітує когнітивні та моторні процеси користувача, відтворюючи сприйняття інформації та темп її обробки.
- **QUIM Editor (Quality in Use Integrated Measurement Editor) [18]** – реалізує модель QUIM, що містить 10 підхарактеристик і 128 мір ЗВ.
- **DRUM (Diagnostic Recorder for Usability Measurement) [19]** – інструмент для аналізу та представлення результатів, отриманих від користувачів.

Переваги та недоліки моделювальних методів

Переваги:

- низька вартість порівняно з емпіричними методами;
- можливість швидко оцінювати різні аспекти ЗВ без залучення користувачів.

Недоліки:

- моделі потребують перевірки на відповідність реальній поведінці людей;
- підготовка моделі починається з когнітивного аналізу завдання, що потребує додаткових ресурсів;
- точність результатів залежить від коректності моделі та початкових даних.

1.4 Управління зручністю використання програмних продуктів

Під час створення та супроводу ПП важливо не лише оцінювати досягнутий рівень ЗВ, але й здійснювати цілеспрямовані дії, спрямовані на формування зручного у використанні продукту. Це охоплюється поняттям управління ЗВ.

У сучасній теорії управління найширшого застосування набув процесний підхід, [9] який є базовим принципом міжнародних стандартів систем менеджменту якості. З позиції процесного підходу управління ЗВ можна визначити так:

Управління ЗВ – процеси із застосування необхідних управляючих дій з метою створення програмного продукту, який відповідає очікуванням користувачів, із зручністю використання не нижче заданого рівня. Під рівнем ЗВ будемо розуміти ступінь відповідності ЗВ вимогам. Розрізнятимемо поточний і заданий (розробником) рівні ЗВ.

Основними процесами управління є планування, забезпечення та контроль.

Планування ЗВ – процес, що охоплює вибір вимог, стандартів і методик, які застосовуються до створюваного ПП, а також визначення способів їх досягнення. На цьому етапі формуються критерії, показники та індикатори ЗВ.

Забезпечення ЗВ – передбачає виконання систематичних організаційних та технічних заходів, що гарантують реалізацію усіх визначених процесів, необхідних для досягнення вимог до ЗВ. Часто цей процес розглядають разом із управлінням змінами, оскільки будь-які зміни у вимогах або функціональності ПП впливають на характеристики ЗВ.

Контроль ЗВ – процес перевірки та документування результатів заходів із забезпечення ЗВ. Містить оцінку поточного рівня ЗВ на відповідність встановленим вимогам та визначення коригувальних дій. Метою контролю є виявлення причин незадовільних результатів і розробка рекомендацій щодо їх усунення.

На рис. 1.4 подано узагальнену модель процесів управління ЗВ, у якій показано взаємозв'язки між плануванням, забезпеченням і контролем як циклічними процесами, що забезпечують безперервне вдосконалення ПП.

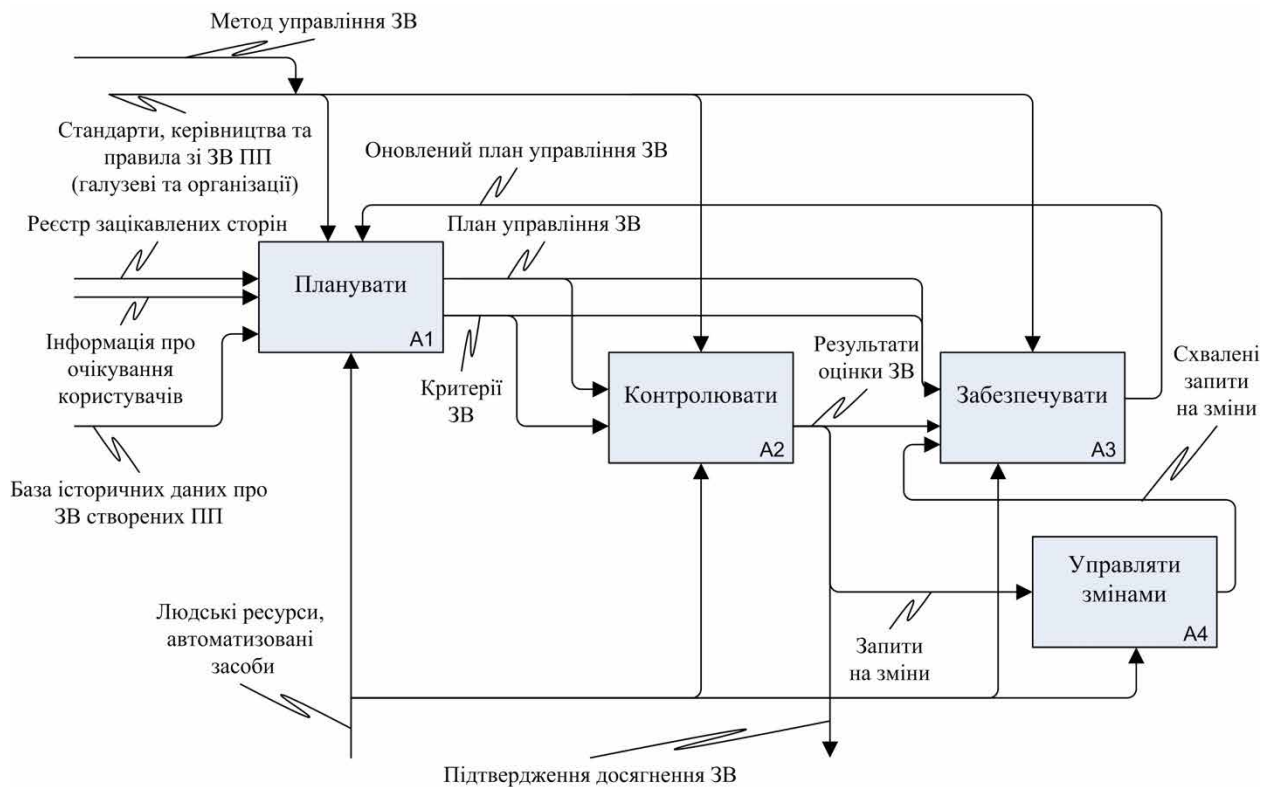


Рис. 1.4. Модель процесів управління ЗВ

Важливість належного управління ЗВ демонструє онтологія, зображена на рис. 1.5, побудована на основі ключових понять предметної області. Для формалізації використано схематичну мову **IDEF5**. У схемі застосовано:

- відношення **part-of** – використовується у композиційній частині онтології та позначає складові частини об'єкта;
- відношення **subkind-of** – використовується у класифікаційній частині та позначає різновиди об'єктів.

Онтологічна схема відображає взаємозв'язки між поняттями ЗВ, її характеристиками, показниками, методами оцінювання та процесами управління. [17]

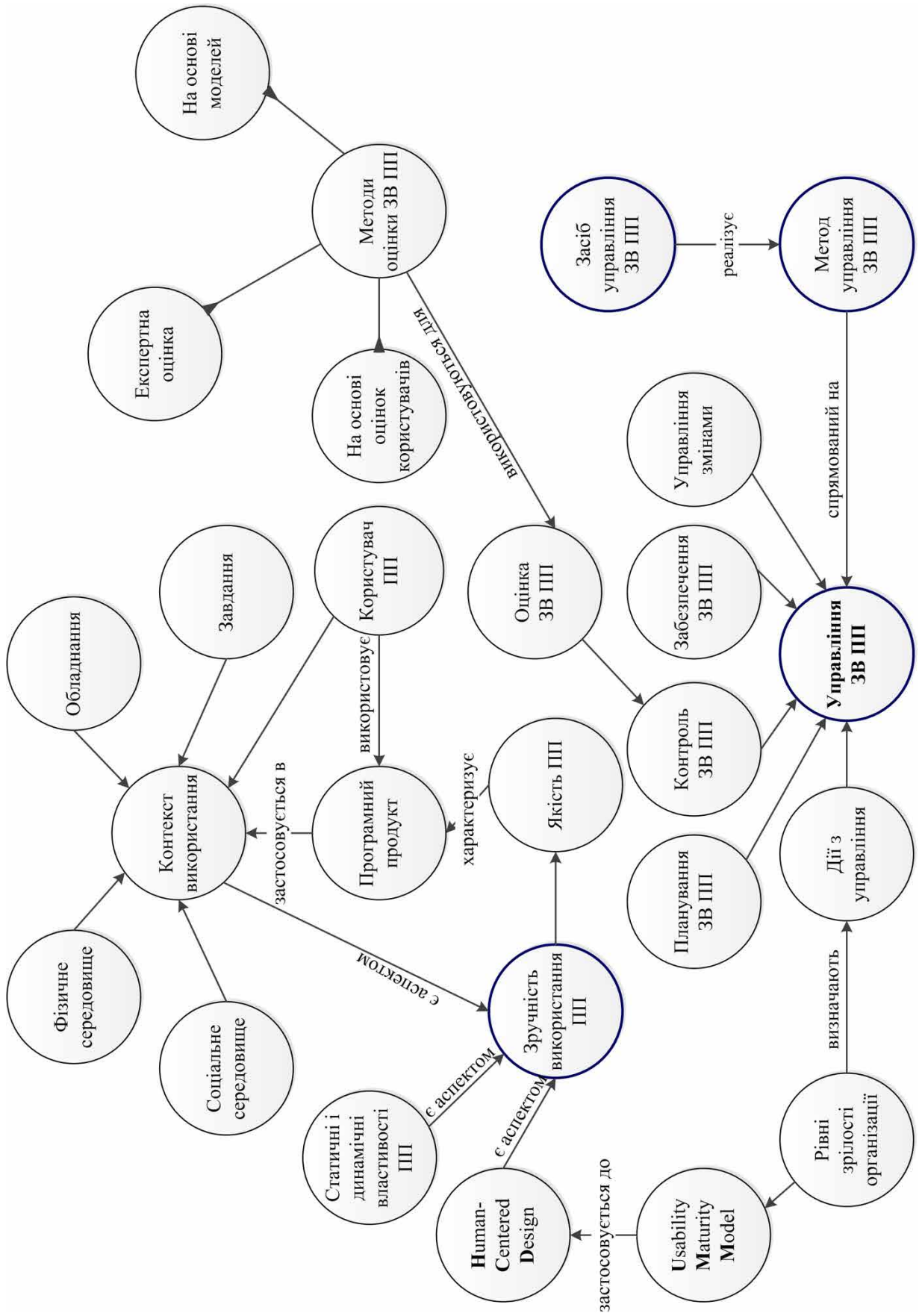


Рис. 1.5 Схема онтології зручності використання програмних продуктів

Висновки до розділу

1. У межах дослідження ЗВ визначено як властивість ПП, що характеризує його здатність забезпечувати користувачам можливість виконувати потрібні завдання із достатньою результативністю, раціональними витратами ресурсів та прийнятним рівнем задоволеності у конкретному контексті використання. Поняття «користувач» розглядається як суб'єкт (фізична або юридична особа), що взаємодіє з ПС для досягнення визначених цілей.

2. Аналіз показує, що ЗВ слід оцінювати не лише через внутрішні та зовнішні характеристики ПП, а й через організацію процесів його створення. Особливе значення має застосування принципів людино-орієнтованого підходу (HCD), який дозволяє врахувати потреби користувачів, обмеження розробників та вимоги власників ПЗ, формуючи збалансовану основу для отримання якісного кінцевого продукту.

3. Наявні методи забезпечення ЗВ здебільшого зосереджені на контролі вже створених рішень, а саме на діагностиці, аналізі та вимірюванні характеристик ПП. Водночас інструментів, які б системно підтримували виявлення причинних недоліків та пропонували цілеспрямоване вдосконалення ПП у процесі розробки, практично не існує. Це підкреслює значущість впровадження підходів, що дозволяють здійснювати управління ЗВ, а не лише її фіксацію.

4. Управління ЗВ є ключовим елементом моделей зрілості організацій, що впроваджують HCD. Для переходу до високих рівнів зрілості необхідно не лише оцінювати поточний стан ЗВ, але й аналізувати чинники, що знижують її якість, формувати коригувальні дії та забезпечувати їх впровадження. Досягти цього можливо лише за умови ітераційності оцінювання та документування змін, а також активного використання зворотного зв'язку від реальних користувачів. Такий підхід створює передумови для формування методу та засобу управління ЗВ, які забезпечують постійний розвиток ПП та підвищення його відповідності потребам користувачів.

РОЗДІЛ II

МЕТОД УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ

2.1 Метод управління зручністю використання програмних продуктів

У межах даної роботи запропоновано метод управління ЗВ ПП, який базується на принципі повторюваної (ітераційної) оцінки стану ЗВ під час розроблення ПП та визначенні оптимальної стратегії досягнення цільового рівня ЗВ, встановленого розробниками на початку проєкту. У випадку використання ітераційних методологій розробки оцінювання ЗВ здійснюється після кожного завершенного циклу – ітерації, результатом якої є нова версія ПП. Для каскадних або інших неітераційних підходів повторюваність означає періодичне виконання оцінювання в рамках процесів управління.

У цьому методі ЗВ представлено як структуру, що складається з атрибутів, показників та відповідних метрик.

Атрибут – це невід’ємна властивість сутності, яка характеризується кількісно чи якісно людиною або за допомогою автоматизованих засобів.

Показник – кількісне відображення стану атрибутів з урахуванням конкретних умов створення й використання ПП.

Метрика – це комбінація конкретного методу вимірювання (способу отримання значень) показника та шкали вимірювання.

У дослідженні ЗВ розглядається як комплекс властивостей, що підлягають деталізації до рівня, на якому стає можливим формування точних числових оцінок. [6, с. 94]

Відповідно до процесного підходу, поданого у підрозділі 1.3, управління ЗВ включає такі етапи:

1. Формування експертної ієрархії ЗВ. Цей етап реалізується за принципом руху «від загального до конкретного» (top–down) і містить такі рівні:

а) *атрибути ЗВ* – визначаються на підставі вимог до ЗВ, галузевих стандартів, накопичених даних про раніше створені ПП та очікувань користувачів. Для атрибутів задаються пріоритети, встановлюються зв'язки з вимогами, а також визначається допустимий інтервал їхніх значень (за участю менеджерів або замовника);

б) показники ЗВ – атрибути деталізуються до рівня показників, що можуть бути обчислені;

в) метрики ЗВ – кожен показник розкладається на конкретні метрики, які дають можливість отримувати числові оцінки користувачів щодо роботи ПП. Метрики використовуються для подальшого обчислення значень показників і атрибутів.

2. Розрахунок значень показників ЗВ. Обчислення виконується на основі значень метрик, отриманих за результатами оцінок користувачів.

3. Побудова математичної моделі оцінювання ЗВ. Модель дає змогу агрегувати часткові критерії ЗВ (результати оцінок користувачів та експертні ранжування) в єдину інтегральну кількісну характеристику.

Якщо отримане значення ЗВ відповідає або перевищує заданий рівень, формується підсумковий звіт.

Якщо ж зафіксоване відхилення – виконується перехід до наступного етапу.

4. Побудова математичної моделі забезпечення необхідного рівня ЗВ. Модель оцінки доповнюється функцією трудомісткості змін показників, що дозволяє знайти найоптимальніший варіант покращення ЗВ.

5. Формування оптимального плану забезпечення ЗВ. У результаті визначається перелік показників, що потребують покращення, та обчислюється величина необхідних змін. Для аналізу взаємного впливу показників пропонується визначення типу та характеру зв'язків між їхніми парами.

6. Реалізація плану змін та перевірка досягнення встановленого рівня на наступній ітерації. За потреби моделі уточнюються, а процес управління повторюється. [1, с. 344]

На рис. 2.1 подано діаграму діяльності, яка відображає послідовність виконання етапів методу управління ЗВ.

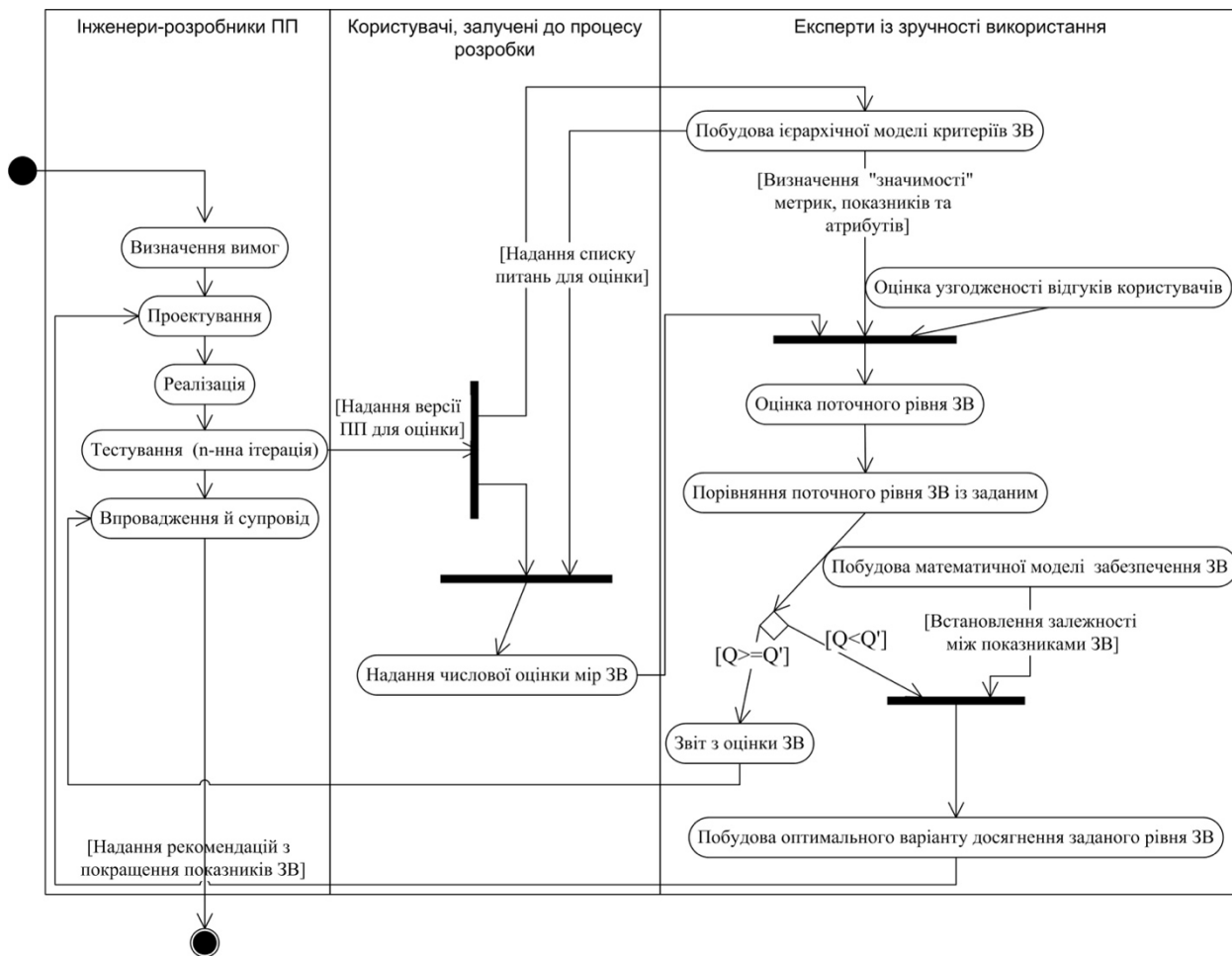


Рис. 2.1 Схема виконання етапів методу управління ЗВ

Принциповою особливістю даного методу є те, що в ньому передбачається не лише оцінка, але й управління ЗВ в процесі створення ПП.

Останнє досягається за рахунок автоматизованої побудови варіанту забезпечення заданого рівня ЗВ на наступній ітерації оптимальним чином на основі математичних моделей оцінки та забезпечення ЗВ, орієнтованих на використання відгуків користувачів. Під *варіантом забезпечення ЗВ* мається на увазі одна з можливих реалізацій зміни показників ЗВ, при якій отримуємо заданий розробником рівень ЗВ. Класифікація створеного методу за ознаками, представлена на рис. 2.2.

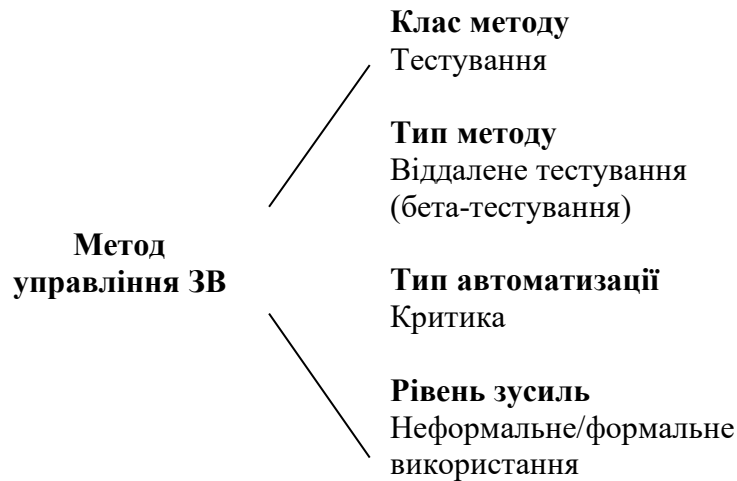


Рис. 2.2 Класифікація методу управління ЗВ

Значним аспектом є визначення етапів життєвого циклу (ЖЦ) ПП, на яких мають бути реалізовані рекомендації з удосконалення ЗВ, а також ступеня їхнього впливу на робочі продукти. Властивості ЗВ тісно пов'язані з архітектурою ПП і можуть бути уточнені у межах концепції патернів ЗВ (usability patterns). **Патерн ЗВ** – це архітектурний підхід чи механізм, що забезпечує певні властивості ЗВ. Один патерн може впливати на кілька показників, і навпаки – окремий показник ЗВ може бути покращений різними патернами. Вибір ефективного набору патернів залежить від специфіки ПП та визначається фахівцями під час розробки. На рис. 2.3 наведено приклад взаємозв'язків між атрибутами, показниками та патернами ЗВ. [2, с. 126–127]

Враховуючи вище сказане, впливає, що для підвищення рівня ЗВ зміни доцільно починати з етапу проектування, адже саме там формується базова архітектура ПП. Подальше уточнення архітектури може спричинити необхідність перегляду вимог, що потребує внесення змін на ранніх фазах ЖЦ ПП і робить процес удосконалення ЗВ більш ресурсозатратним.

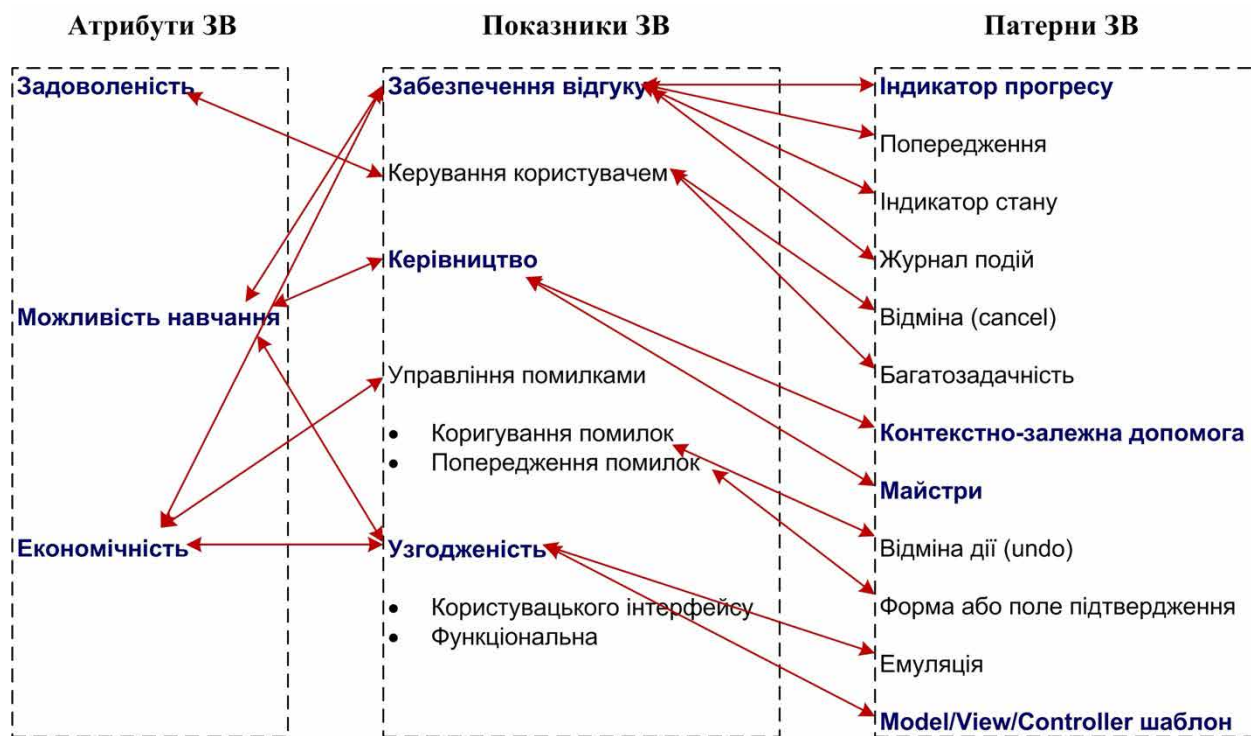


Рис. 2.3 Зв'язок між ЗВ та архітектурою ПП

2.2 Кореляційно-регресійний аналіз показників зручності використання

У межах даного дослідження обрано методологію, яка дає можливість визначити характер та інтенсивність взаємозв'язку між окремими показниками зручності використання (ЗВ). Для цього застосовано алгоритм побудови емпіричних залежностей, що базується на поєднанні кореляційного та регресійного аналізів.

Під час оцінювання метрик ЗВ користувачами кожен показник подається у вигляді числового ряду. Розглядаючи пару таких рядів, один із показників позначається як залежна змінна (**у_k**), а інший – як незалежна (**х_k**). У загальному випадку значення обох змінних можуть містити не лише закономірні (детерміновані) компоненти, а й випадкові складові різної природи. Ці випадкові варіації формуються як через статистичні особливості досліджуваних процесів, так і під впливом зовнішніх чинників, пов'язаних з умовами вимірювань або особливостями отримання даних.

Незалежна змінна x_k традиційно розглядається як детермінована, а випадковість «переходить» головним чином на залежну змінну y_k . Також передбачається, що випадкові компоненти залежного показника підпорядковуються певному ймовірнісному закону, найчастіше – нормальному розподілу. У цьому контексті сам факт залежності значень однієї величини від іншої (показника ЗВ) інтерпретується як регресія.

Пошук потенційної залежності між показниками здійснювався у два етапи:

1. **Кореляційний аналіз** – встановлення сили та статистичної значимості зв'язку між показниками.
2. **Регресійний аналіз** – визначення можливості опису цього зв'язку у вигляді математичного рівняння.

Кореляційний аналіз. Для кількісної оцінки існування зв'язку між досліджуваними сукупностями випадкових величин використовується спеціальний статистичний показник – коефіцієнт кореляції r . Коефіцієнт r – це безрозмірна величина, вона може змінюватися від 0 до ± 1 . Чим ближче значення коефіцієнта до одиниці (неважливо, з яким знаком), тим з більшою впевненістю можна стверджувати, що між двома розглянутими показниками зручності використання існує лінійний зв'язок. Якщо виявиться, що $r = 1$ (або -1), то має місце класичний випадок чисто функціональної залежності (тобто реалізується ідеальний взаємозв'язок). Існують різні аналітичні прийоми визначення коефіцієнта кореляції.

У межах дослідження розглядалися два типи коефіцієнтів:

- **лінійний коефіцієнт кореляції Пірсона**, який дозволяє визначити тісноту лінійного зв'язку за умови нормальності розподілу даних;
- **ранговий коефіцієнт кореляції**, що застосовується у випадках необхідності аналізу порядкових оцінок (зокрема, при узгодженні експертного ранжування).

Коефіцієнт Пірсона обчислюється за формулою:

$$r_{XY} = \frac{\text{cov}_{XY}}{\sigma_X \sigma_Y} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}},$$

де x_i – числові значення, які приймає показник X ,

y_i – числові значення, які приймає показник Y ,

\bar{x} – математичне сподівання по X ,

\bar{y} – математичне сподівання по Y .

Знаючи коефіцієнт кореляції, можна дати якісно-кількісну оцінку тісноти зв'язку. У дослідженні використовується шкала Чеддока, представлена в таблиці 2.1.

Оцінка значимості коефіцієнта парної кореляції проводиться шляхом порівняння його абсолютної величини з табличним (або критичним).

Таблиця 2.1

Якісна оцінка тісноти зв'язку за Чеддоком

Величина коефіцієнта парної кореляції	Характеристика сили зв'язку
До 0,3	Практично відсутній
0,3 – 0,5	Слабкий
0,5 – 0,7	Помітний
0,7 – 0,9	Сильний
0,9 – 0,99	Дуже сильний

показником $r_{\text{крит}}$, значення якого беруться зі спеціальної таблиці. Якщо $|r_{\text{розрах}}| \geq r_{\text{крит}}$, то із заданою ймовірністю (зазвичай 95 %) можна стверджувати, що між розглядуваними показниками існує значимий лінійний зв'язок. У зворотному випадку робиться висновок про відсутність такого зв'язку.

Коефіцієнт рангової кореляції застосовується в дослідженні при перевірці узгодженості ранжувань експертів, про що йтиметься далі. Для визначення сили зв'язку між показниками ЗВ застосовується лінійний коефіцієнт кореляції.

Алгоритм визначення сили та значимості зв'язку між обчисленими, на основі користувацьких оцінок метрик, значеннями показників зручності використання представлено на рис. 2.4.

Регресійний аналіз. Якщо на попередньому етапі було встановлено статистично значимий і достатньо сильний зв'язок між двома показниками, наступним кроком є визначення форми цієї залежності. Для формального опису регресійної моделі потрібен умовний закон розподілу Y за фіксованих значень

Х. На практиці отримати таку інформацію, як правило, не вдається, оскільки дослідник має лише вибірку пар значень (x_i, y_i) обмеженого об'єму n .

У даному випадку мова може йти лише про оцінку (апроксимацію) за вибіркою функції регресії. Такою оцінкою є вибіркова лінія (крива) регресії:

$$\hat{y} = \hat{\phi}(x, b_0, b_1, \dots, b_p),$$

де \hat{y} – умовна групова середня показника Y при фіксованому значенні показника $X = x$, b_0, b_1, \dots, b_p – параметри кривої.

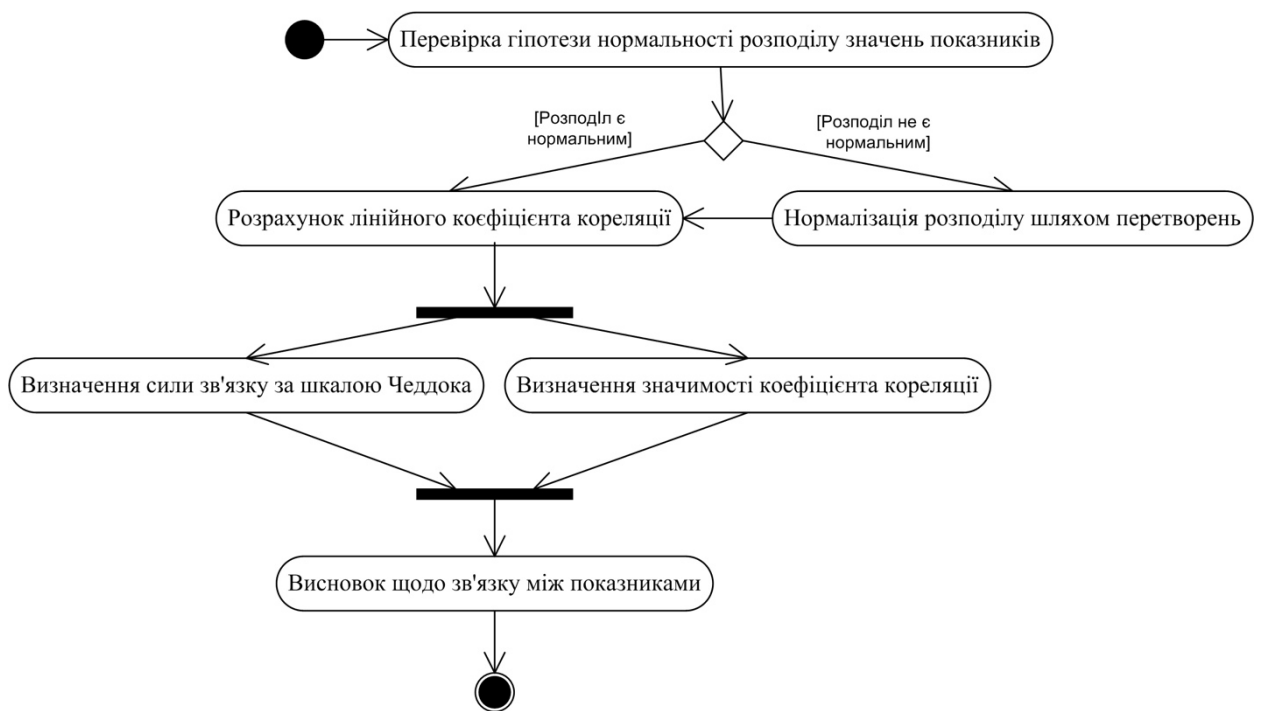


Рис. 2.4 Алгоритм визначення зв'язку між показниками ЗВ

Апроксимуючі рівняння (емпіричні формули) доцільно використовувати, оскільки:

1. Точний аналітичний вираз залежності між досліджуваними показниками ЗВ може залишатись невідомим і тому за необхідністю доводиться обмежуватись наближеними формулами емпіричного характеру.

2. Точна функціональна залежність виражається формулою настільки складною, що її безпосереднє застосування при обчисленнях є проблематичним.

Залежність між однією й тією ж самою парою показників можна виразити різними за видом рівняннями регресії. Найточніша апроксимація

характеризується найбільшим співпадінням значень, обчислених за формулою, з дослідними даними.

Рівняння регресії перевіряється на значимість (адекватність моделі) за допомогою F-критерія (критерія Фішера):

$$F_{emp} = \frac{S_X^2}{S_Y^2},$$

де S_X^2 і S_Y^2 – відповідно дисперсії двох вибірок, членами яких є значення залежних показників.

Перед застосуванням критерію Фішера необхідно перевірити виконання наступних умов:

1. Значення показників мають бути виміряні у шкалі інтервалів або відношень.
2. Розподіл значень залежної змінної повинен бути близьким до нормального.
3. У чисельник дробу підставляється більша дисперсія, у знаменник – менша.

Отже, якщо гіпотеза нормальності розподілу не підтверджується, то необхідно виконати відповідні математичні перетворення вхідних даних. [12, с. 154-155]

2.3 Кількісний та якісний аналіз узгодженості експертів

Експертне оцінювання є одним із найефективніших інструментів роботи з даними, що важко піддаються формалізації. Воно дає можливість відібрати найбільш обґрунтовані й узгоджені судження фахівців та використати їх у подальшому для підготовки рішень щодо характеристик ПП, параметрів ЗВ чи інших властивостей ПС. Перш ніж формувати колективне рішення стосовно вагомості показників, необхідно дослідити результати опитування експертів та визначити ступінь узгодженості їхніх оцінок.

У більшості випадків експерти надають свої оцінки у порядковій шкалі – найчастіше у вигляді ранжувань або класифікацій об’єктів. Згідно з результатами багатьох досліджень, люди легше дають відповіді на запитання порівняльного характеру, ніж на точні кількісні запити [22], що підтверджує доцільність використання рангових оцінок.

Специфічними є підходи до перевірки узгодженості, що використовуються при оцінці об’єктів методом ранжування. У цьому випадку результатом роботи експерта є ранжування, що представляє собою послідовність рангів (для експерта j): $x_{1j}, x_{2j}, \dots, x_{nj}$.

Узгодженість між ранжуваннями двох експертів можна визначити за допомогою коефіцієнта рангової кореляції Спірмена:

$$r = 1 - \frac{6 \sum_{i=1}^n (x_{ij} - x_{ik})^2}{n(n^2 - 1)} = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)},$$

де x_{ij} – ранг, наданий i -му об’єкту j -им експертом;

x_{ik} – ранг, наданий i -му об’єкту k -им експертом;

d_i – різниця між рангами, наданими i -му об’єкту;

n – число парних членів ряду, або об’єм вибірки.

Величина r може змінюватися від -1 до $+1$. При повному співпадінні оцінок коефіцієнт дорівнює 1 , при найбільшому розходженні – -1 .

При необхідності визначення узгодженості ранжувань більше, ніж двох експертів, розраховується коефіцієнт конкордації – загальний коефіцієнт рангової кореляції для групи, що складається з m експертів. Він інтерпретується як нормована сума відхилень сумарного рангу i -го об’єкту від середнього сумарного рангу. Коефіцієнт конкордації обчислюється через статистику Фрідмана. При відсутності зв’язаних (однакових) рангів:

$$W = \frac{S}{m(n-1)},$$

де $S = \frac{12}{m \cdot n(n+1)} \sum_{i=1}^n \left(\left(\sum_{j=1}^m x_{ij} \right)^2 - 3m(n+1) \right)$ – статистика Фрідмана (або X^2 – хі-квадрат).

Коефіцієнт конкордації набуває значень від 0 до 1: що ближче значення до 0 – то менш узгоджені оцінки. Для інтерпретації використовується шкала Чеддока (табл. 2.1), а значущість оцінюється шляхом – порівняння S із табличним значенням S_T .

У практиці аналізу експертних оцінок найчастіше застосовують коефіцієнт конкордації Кендела. Однак цей підхід має суттєві обмеження: він не визначає, які саме експерти є джерелом розбіжностей; не дозволяє виявити об'єкти, що викликають найбільші суперечності; та передбачає перевірку нульової гіпотези про рівномірний розподіл ранжувань. Водночас навіть її прийняття чи відхилення не гарантує однозначної інтерпретації, адже у групі можуть існувати кілька «кластерів думок». Крім того, точність коефіцієнта залежить від однорідності оцінюваних об'єктів: наявність скупчень може спричинити похибки.

З огляду на ці недоліки, у дипломній роботі пропонується комбінувати традиційні статистичні методи з підходами інтелектуального аналізу даних (Data Mining), насамперед – алгоритмами кластеризації. Такий підхід відповідає концепції стійкого аналізу, де узагальнені висновки робляться лише на основі тих результатів, що підтверджуються різними підходами, а суперечливі висновки вказують на можливий суб'єктивізм.

Використання **Data Mining** для аналізу структури ранжувань.

Data Mining – це процес виявлення інтерпретованих та практично корисних знань у необроблених даних, що зберігаються в БД, СД або інших джерелах. Залежно від цілей аналізу методи поділяються на описові та прогнозуючі. Для завдання оцінки узгодженості ранжувань доцільно застосовувати описові методи, зокрема кластерний аналіз:

- алгоритм k -середніх,
- алгоритм k -медіан,

- ієрархічні методи,
- карти Кохонена,
- методи візуалізації даних.

Ієрархічні методи здебільшого працюють із категорійними ознаками, а тому не підходять для обробки порядкових рангових даних. Серед неієрархічних підходів найпоширенішим є алгоритм k -середніх, проте його недоліки (чутливість до викидів, низька ефективність на великих БД, та труднощі при перетинанні кластерів) роблять його непридатним для цього завдання. Тому він не розглядається як базовий метод аналізу ранжувань [2, с. 129].

У рамках дипломної роботи узгодженість оцінок пропоную визначати за допомогою двох взаємодоповнювальних інструментів:

1. коефіцієнта конкордації Кендела – для кількісного аналізу;
2. кластеризації ранжувань – для якісного аналізу та виявлення груп експертів зі схожими думками.

Алгоритм реалізований у створеній ПС та представлений на рис. 2.5.

Працюючи над дипломною роботою, пропоную використовувати розрахунок коефіцієнта конкордації Кендела разом із кластеризацією ранжувань експертів для визначення їх узгодженості на кількісному та якісному рівнях. Відповідний алгоритм представлено на рис. 2.5, а також реалізовано в розробленій програмній системі.

Зауважимо, що метод кластеризації може також застосовуватися для визначення узгодженості оцінок користувачів, які вони надають метрикам ЗВ. Замість коефіцієнта конкордації в цьому випадку доцільно використовувати статистичні характеристики – міри розсіювання, – оскільки застосовується шкала інтервалів:

- Варіаційний розмах $R = x_{\max} - x_{\min}$, де x_{\max} – максимальна оцінка метрики, x_{\min} – мінімальна оцінка метрики.

- Середнє квадратичне відхилення $\sigma = \sqrt{\frac{\sum_{j=1}^m (x_j - \bar{x})^2}{m-1}}$, де x_j – оцінка, дана j -им користувачем, m – кількість користувачів, \bar{x} – середнє арифметичне оцінок користувачів.
- Коефіцієнт варіації $V = \frac{\sigma}{\bar{x}} \cdot 100\%$.

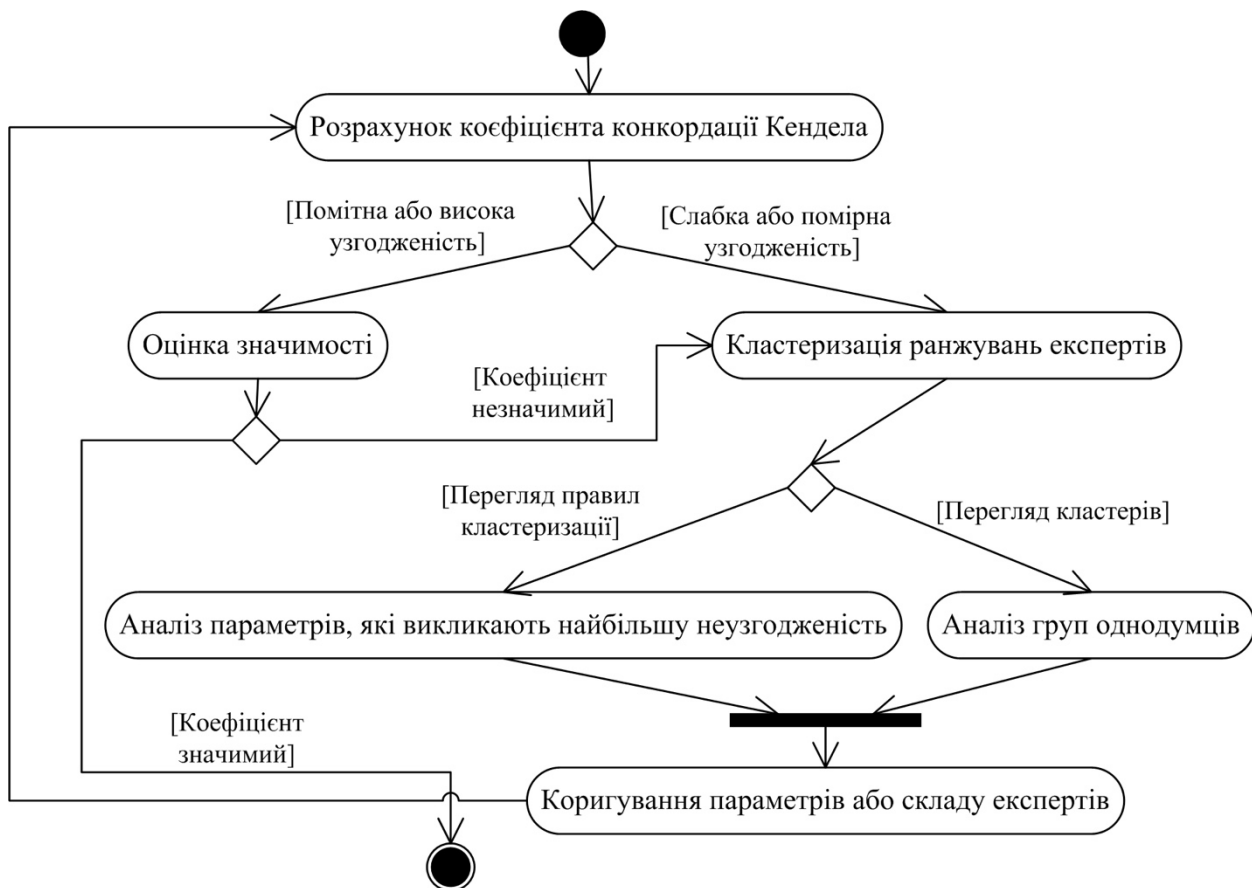


Рис. 2.5 Схема методики аналізу узгодженості ранжувань експертів

2.4 Визначення ваги метрик, показників та атрибутів ЗВ

Визначення ваги метрик, показників та атрибутів ЗВ із подальшим обчисленням вагових коефіцієнтів є однією з ключових задач ефективного використання математичної моделі забезпечення ЗВ. Найбільш поширеними підходами, які застосовують експерти для оцінки значимості критеріїв, є безпосередня оцінка, ранжування та повне парне порівняння. У межах

розробленого методу управління ЗВ використовується ранжування, що забезпечує впорядкування критеріїв за спаданням їх відносної важливості.

Передбачається впорядкування критеріїв відповідно зі зменшенням їх відносної важливості. При цьому кожному складовому критерію нижчого рівня ієрархії приписується ранг в рамках критерію вищого рівня, до якого він входить, – натуральне число, що характеризує його порядковий номер. При неможливості розрізнити за важливістю два або декілька критеріїв використовується один і той самий ранг, значення якого є середнє суми місць, поділених цими критеріями. Результатом роботи експерта є ранжування, що представляє собою послідовність рангів (для експерта j): $x_{1j}, x_{2j}, \dots, x_{nj}$.

Якщо думки експертів виявляються узгодженими, наступним кроком є побудова загального підсумкового ранжування. Оскільки оцінки експертів представлені в порядковій шкалі, для них неприйнятне усереднення методом арифметичних середніх.

Для побудови загальної думки експертів введемо скінченномірний дискретний простір рангів та метрику в ньому. Ранжування j -го експерта представляє собою точку R_j в просторі рангів. Метрика $d(R_i, R_j)$ – це відстань між ранжуваннями експертів i та j . Медіана Кемені дозволяє визначити групову думку як точку в просторі рангів, сума відстаней від якої до всіх інших точок є мінімальною:

$$R_M = \arg \min_R \sum_{j=1}^m k_j d(R_j, R),$$

де R_M – медіана Кемені або підсумкове ранжування, k_j – коефіцієнт компетентності експерта j , R – поточне ранжування, за яким проводиться мінімізація.

Метрику відстані визначимо наступним чином:

$$d(R_i, R_j) = \sum_{k=1}^n |x_{ki} - x_{kj}|,$$

де k – кількість критеріїв в ранжуванні.

На основі підсумкового ранжування можна визначити ваговий коефіцієнт кожного критерію ЗВ у програмному продукті, використовуючи шкалу

$$\text{Фішберна: } p_i = \frac{2(n - i + 1)}{n(n + 1)}$$

де p_i – коефіцієнт значимості i -го критерію; i – ранг поточного критерія в підсумковому ранжуванні; n – кількість критеріїв.

Висновки до розділу

1. Розроблено метод управління ЗВ ПП, який передбачає не лише оцінку, але й активне управління ЗВ протягом процесу створення та супроводу ПП. Метод забезпечує автоматизоване формування оптимальних варіантів підвищення ЗВ на наступних ітераціях за допомогою математичних моделей оцінки та управління, орієнтованих на відгуки користувачів та експертні оцінки.

2. Запропоновано математичну модель забезпечення ЗВ, яка базується на багатокритеріальній системі оцінки показників, метрик та атрибутів ЗВ. Задача забезпечення ЗВ формалізована як двокритеріальна оптимізаційна задача методом головного критерію із цільовою функцією трудомісткості. Використання такої моделі дозволяє систематизовано планувати дії щодо досягнення заданого рівня ЗВ під час розробки та супроводу ПП.

3. Показано ефективність застосування **кореляційно-регресійного аналізу** для встановлення взаємозв'язків між показниками ЗВ. Виявлені залежності дозволяють передбачати вплив змін одного показника на інші, що є основою для реалізації управління змінами ЗВ.

4. Розроблено підхід до оцінки узгодженості експертних оцінок, який поєднує **коефіцієнт конкордації Кендела** та методи кластеризації ранжувань експертів. Це дозволяє визначати як кількісну, так і якісну узгодженість думок експертів щодо значимості метрик та показників ЗВ.

5. Визначено вагові коефіцієнти для метрик, показників та атрибутів ЗВ на основі підсумкового ранжування експертів із застосуванням **медіани Кемені та шкали Фішберна**. Отримані ваги забезпечують кількісне представлення

відносної важливості кожного критерію та можуть бути інтегровані у систему підтримки прийняття рішень щодо управління ЗВ.

РОЗДІЛ ІІІ

ПРОГРАМНА СИСТЕМА УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ

3.1 Функціональні вимоги до програмної системи та вибір архітектурного стилю

ПС повинна забезпечувати процес управління ЗВ під час розроблення ПП шляхом реалізації визначених функціональних вимог. Модель цих вимог представлено за допомогою діаграми станів (рис. 3.1). Наявні інструменти, короткий огляд яких наведено в підрозділі 1.3, не дають можливості реалізувати запропонований метод управління ЗВ (тип автоматизації – «Критика»). У зв'язку з цим виникає потреба створення спеціалізованої програмної системи, функціональні можливості якої представлено нижче.

Функціональні вимоги які забезпечують завантаження результатів опитування у БД:

- автентифікація експерта;
- вибір шаблону для імпорту даних;
- автоматичне заповнення шаблону інформацією з бази даних;
- передавання сформованого шаблону анкети користувачу чи експерту.

Вимоги первинного статистичного аналізу отримання даних:

- Імпорт оцінок, рангів метрик та показників та атрибутів ЗВ із БД;
- Обчислення основних статистичних показників: дисперсії,

математичного сподівання, медіани, коефіцієнтів ексцесу та асиметрії;

- Побудова двовимірної гистограми розподілу значень оцінок і рангів;
- Додавання результатів опитування до БД;
- Повідомлення користувача або експерта про успішне внесення даних.

[3, с. 189-190].

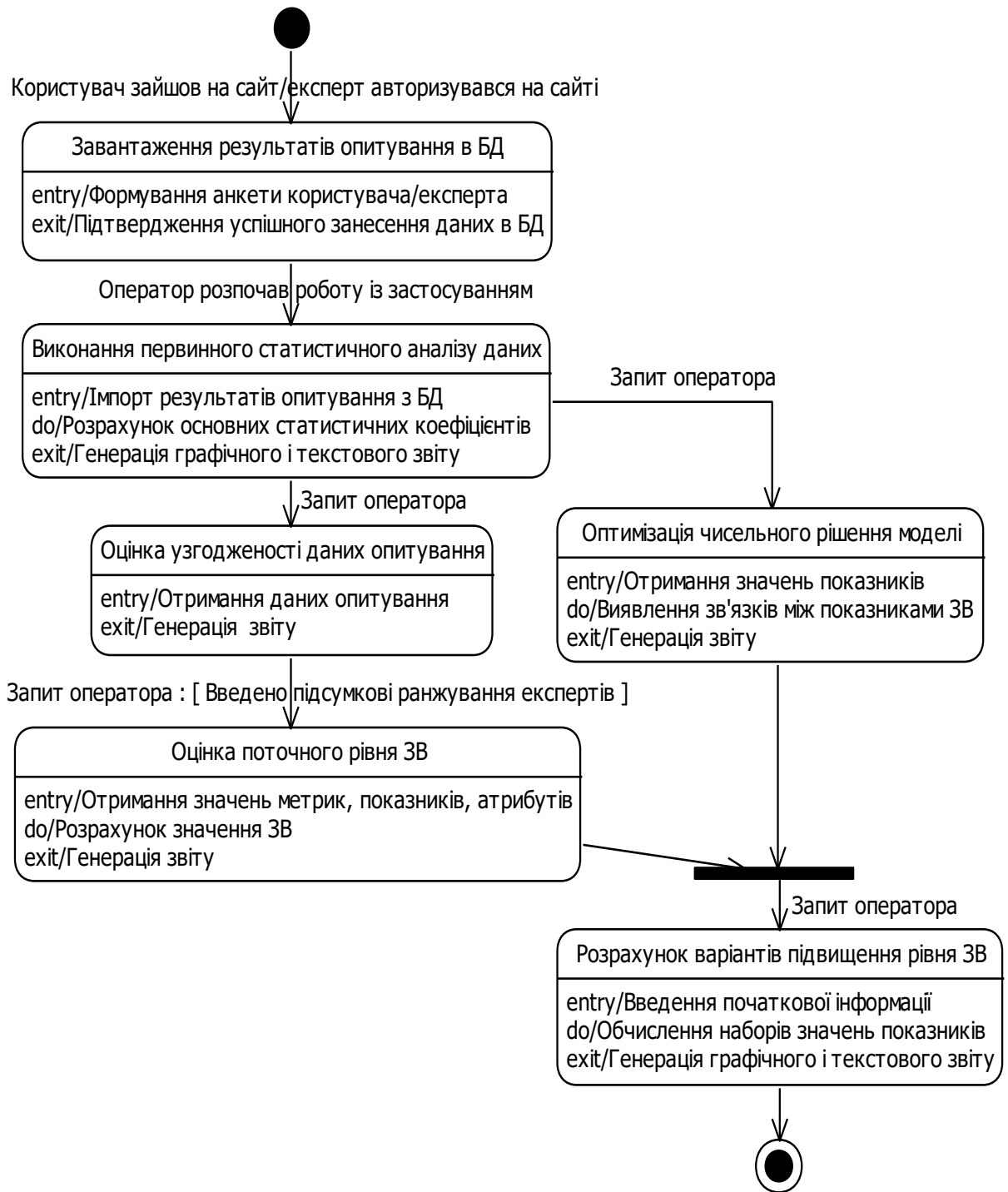


Рис. 3.1 Модель функціональних вимог до програмної системи

Функції системи забезпечуються вимогами оцінки узгодженості:

- обчислення коефіцієнта конкордації по окремоті: група користувачів, експерти;
- виконання кластерного аналізу відповідно до даних опитування експертів.

Вимоги оптимізації чисельного рішення:

- Перерахунок статистичних показників (середнє, медіана, дисперсія, ексцес, асиметрія) для показників ЗВ;
- Коригування закону розподілу відповідно до вибраного математичного перетворення;
- Обчислення коефіцієнтів кореляції для всіх можливих пар показників;
- Вибір регресійної моделі для певних пар показників;
- Оцінка коефіцієнта детермінації та формулювання рівняння регресії;
- Побудова графіка лінії регресії.

Вимоги визначення поточного рівня ЗВ:

- Введення підсумкових значень рангів для метрик і показників ЗВ;
- Обчислення вагових коефіцієнтів для метрик та показників;
- Оцінка поточного рівня ЗВ на основі введених даних.

Вимога розрахунку варіантів підвищення рівня ЗВ:

- Введення формули залежності між показниками;
- Визначення бажаного рівня ЗВ;
- Прописування кроку для варіацій значень показників;
- Розрахунок комбінацій показників, що відповідають вимогам;
- Побудова тривимірної діаграми з точками, що відображають зміни в показниках;
- При виборі точки на графіку виведення інформації про показники: їх початкові значення та необхідні зміни.

Для реалізації зазначених функціональних вимог запропоновано використати **гібридну архітектуру ПЗ** (рис. 3.2), яка комбінує елементи клієнт – серверної архітектури, орієнтованої на дані.

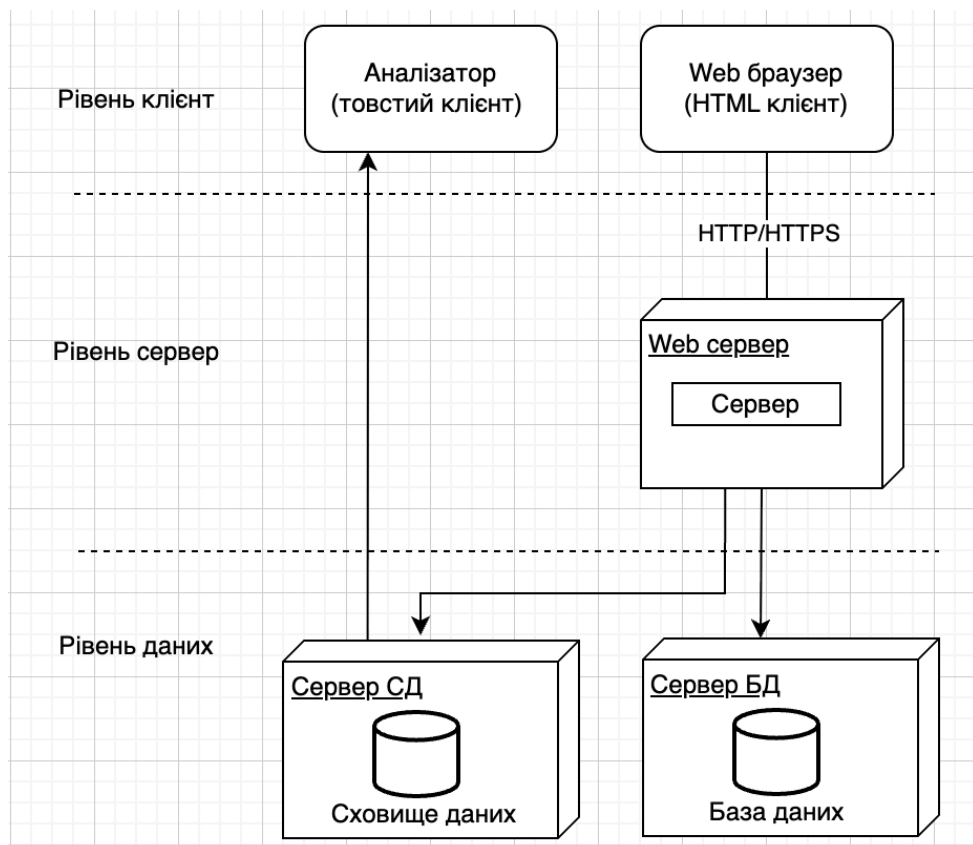


Рис. 3.2 Архітектура програмної системи

Таке архітектурне рішення зумовлене потребою в централізованому зборі інформації її довготривалому зберіганні та організації сховища даних. Передбачається використання моделі «товстого клієнта», оскільки оброблення інформації має інтерактивний характер і потребує значних обчислювальних ресурсів [15, с. 44].

3.2 Основні характеристики та варіанти використання системи

Згідно з методом який був запропонований у даному дослідженні, розроблена ПС забезпечує збір інформації про ЗВ ПП на основі оцінок користувачів та висновків експертів, з подальшою її обробкою. Метою є виявлення ключових напрямків для покращення характеристик зі ЗВ досліджуваного ПП. Програмна система управління ЗВ ефективно вирішує – це завдання через автоматизацію збору даних за допомогою опитувань

користувачів, а також створення даних для експертного аналізу, заснованого на математичних моделях оцінки та забезпечення ЗВ.

Система складається з чотирьох основних частин:

1. Web-сервер, який забезпечує збір даних.
2. База даних (БД), що зберігає зібрану інформацію та налаштування системи.
3. Сховище даних (СД), яке зберігає агреговану та підготовлену для аналізу інформацію з операційної БД.
4. Аналізатор, що здійснює управління системою, аналізує дані та формує результати.

База даних. Дані на сервері зберігаються в БД, що керується MariaDB. Вибір цієї бази даних зумовлений характеристиками розв'язуваної задачі управління ЗВ: порівняно невеликий обсяг збережених даних і обмежена кількість зв'язків між таблицями дозволяють ефективно використовувати цей економічний та ресурсозберігаючий інструмент.

БД містить таблиці чотирьох типів (рис. 3.3):

1. Таблиці зв'язків: `attribute_indicator_map`, `indicator_metric_map`, `metric_question_map`. Ці таблиці містять зв'язки між атрибутами і показниками, показниками і метриками, метриками і оцінками користувачів відповідно до ієрархії критеріїв ЗВ.
2. Таблиці ідентифікаторів: `attributes_dict`, `indicator_dict`, `metric_dict`, `question_dict`, `experts_dict`, `users_dict` для зберігання інформації про атрибути, показники, метрики, оцінки ЗВ, експертів та користувачів відповідно.
3. Таблиці оцінок та рангів: Ці таблиці, містять числові оцінки та рангові значення, виставлені критеріям ЗВ досліджуваного ПП, наприклад, таблиці `attributes_answers` (для користувачів) та `attributes_answers_exp` (для експертів).
4. Таблиця налаштувань: Таблиця `settings` містить загальні налаштування програмного комплексу для оцінки та управління ЗВ, такі як максимальна

кількість користувачів та експертів, які можуть брати участь в опитуванні, а також паролі для доступу експертів [10].

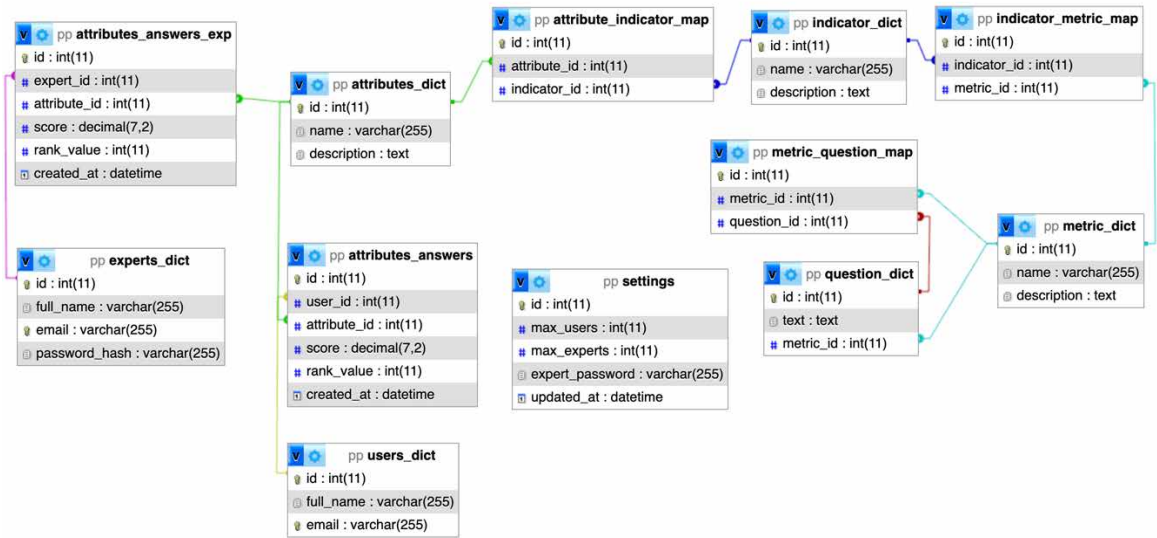


Рис. 3.3. Схема бази даних

Сховище даних. Згідно з розробленою БД, було створено СД, яке забезпечує централізоване зберігання та обробку інформації, а також проведення аналітичного аналізу отриманих оцінок і показників. Структура СД дозволяє виконувати регресійний аналіз, порівнювати результати за різними атрибутами та прогнозувати зміни характеристик ЗВ програмного продукту. Схему сховища даних можна побачити на рис. 3.4.

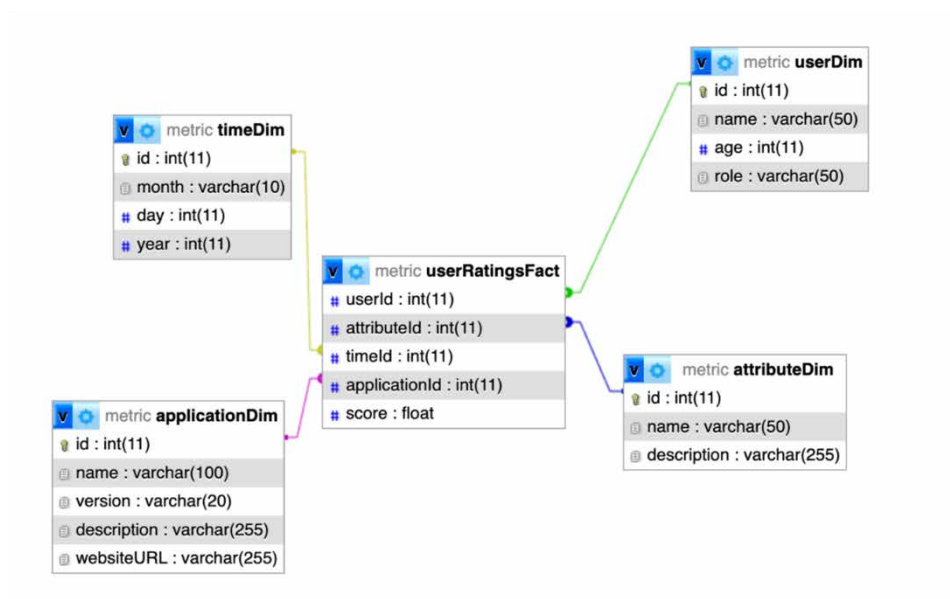


Рис. 3.4. Схема сховища даних

Аналізатор реалізовано як інтерактивну графічну систему, яка дозволяє налаштовувати ПЗ, здійснювати вибірку даних із СД, проводити їхній аналіз у напівавтоматичному режимі та показувати підсумкові результати. Під час розробки були використані мови програмування **PHP** і **Python**.

PHP був обраний через свою зручність у створенні веб-інтерфейсів і налаштуванні інтеграції з БД. Це дозволяє швидко реалізувати інтерфейс для користувача та ефективно впроваджувати інтерактивні елементи, що зручні для роботи в графічному середовищі. Веб-система, побудована на PHP, сприяє оперативному налаштуванню параметрів програми та забезпечує інтуїтивно зрозуміле управління усіма функціями системи.

Python, у свою чергу, обраний для аналітичної обробки даних, оскільки має потужні бібліотеки, які дозволяють здійснювати складні обчислення та статистичний аналіз. Такі бібліотеки, як *pandas*, *numpy* і *scikit-learn*, надають великий набір інструментів для машинного навчання, обробки великих обсягів даних та реалізації моделей прогнозування, що необхідно для точної аналітики й автоматизації обробки результатів.

Інтерфейсна частина аналізатора відділена від моделі даних і містить елементи інтегрованого контролера. З метою зручності доступу до класу *ModelController*, який фактично містить модель даних та забезпечує доступ до неї для всіх компонентів програмної системи, а також для гарантії єдності моделі даних використовується породжуючий шаблон проектування *Singleton* (рис. 3.5). Статична функція *Get* завжди повертає один і той самий екземпляр *ModelController*.

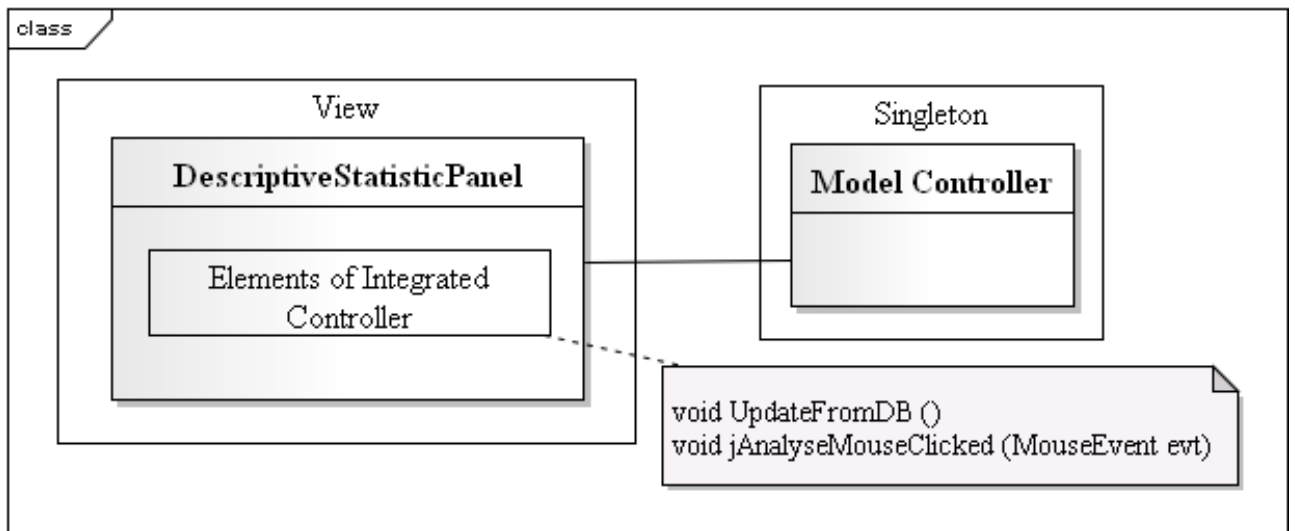


Рис. 3.5 Використання шаблону проектування Singleton

Варіанти використання програмної системи управління ЗВ. Можливі варіанти застосування програмної системи управління ЗВ ілюструються за допомогою діаграми прецедентів, представлено на рис. 3.6. Детальніше зміст кожного з прецедентів можна подати наступним чином [13, с. 59].

Варіант використання 1:

Отримати оцінку рівня ЗВ

Рівень: ціль інженера

Основна дійова особа: інженер із ЗВ

Тригер: інженер запустив аналізатор.

Головний успішний сценарій:

1. Інженер **перевіряє узгодженість даних.**
2. Інженер відкриває вкладку оцінки поточного рівня ЗВ та вводить ранги метрик/показників/атрибутів згідно загального підсумкового ранжування експертів.
3. Система розраховує значення поточного рівня ЗВ.
4. Інженер може продовжити аналіз або вийти з додатку.

Розширення:

- 1а. Дані виявляються неузгодженими.

1. Інженер сповіщає про необхідність коригування вхідних даних опитування.

Варіант використання 2:

Заповнити анкету

Рівень: ціль користувача

Основна дійова особа: користувач

Тригер: користувач запустив web-браузер.

Головний успішний сценарій:

1. Користувач відкриває сторінку загрузки даних.
2. Система обирає шаблон форми загрузки, заповнює його даними з бази та пересилає користувачу.
3. Користувач заповнює шаблон анкети та підтверджує внесені дані.
4. Система вносить отримані дані в базу.
5. Система сповіщає користувача про успішне занесення даних.

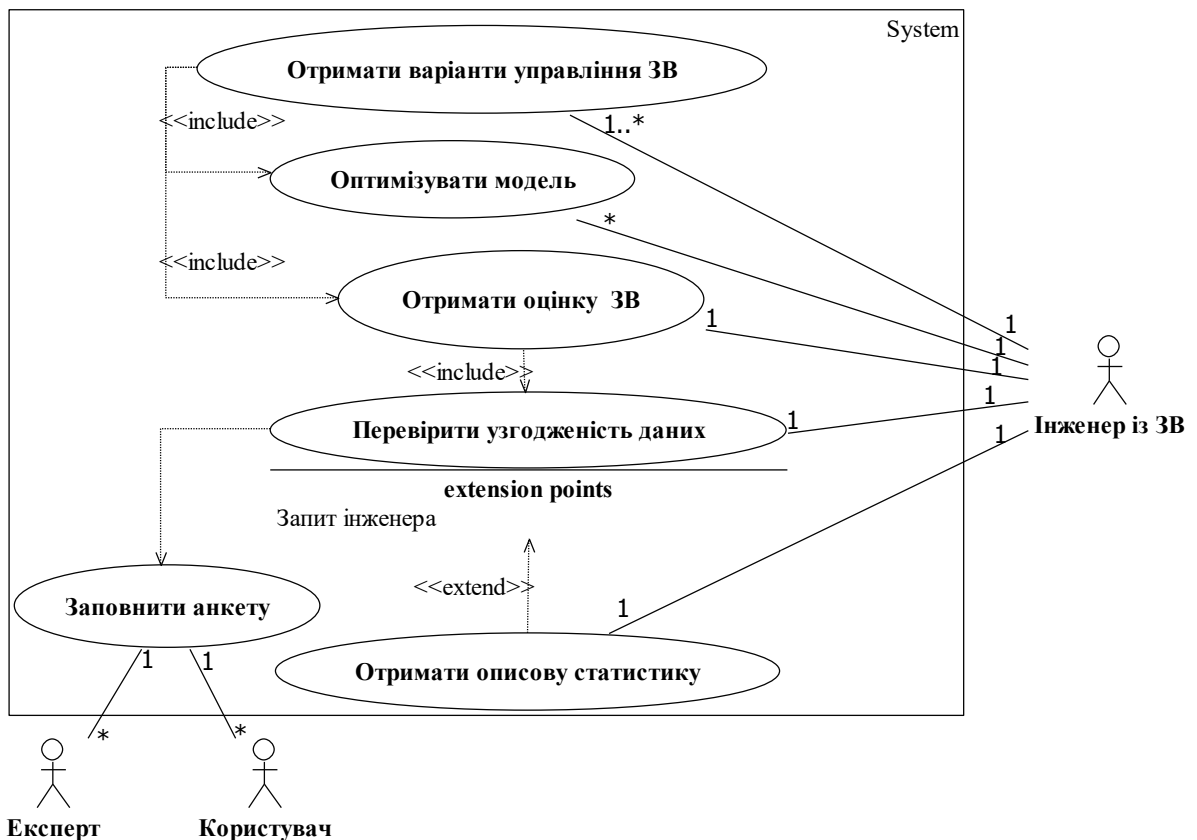


Рис. 3.6 Варіанти використання програмної системи

Розширення:

1а. Користувач є експертом предметної області.

1. Експерт вводить логін та пароль.
2. Система здійснює авторизацію експерта.

5а. Система не підтверджує успішне занесення даних.

1. Користувач може повторити заповнення та відправку даних або закінчити сеанс.

Варіант використання 3:

Перевірити узгодженість даних

Рівень: ціль інженера

Основна дійова особа: інженер із ЗВ

Тригер: інженер запустив аналізатор.

Головний успішний сценарій:

1. Інженер відкриває вкладку отримання даних та запускає процес аналізу.
2. Система імпортує дані опитування з БД.
3. Інженер запускає процес кластерного аналізу та розрахунку коефіцієнтів конкордації .
4. Система виводить результати розрахунків окремо для експертів та користувачів.
5. Інженер може продовжити аналіз або вийти з додатку.

Розширення:

1а. Інженер бажає отримати описову статистику для первинного аналізу даних.

1. Інженер обирає метрики/показники/атрибути зі списку.
2. Система будує гістограму розподілу оцінок/рангів для кожної метрики/показника/атрибута, а також виводить значення основних статистичних коефіцієнтів.

Варіант використання 4:

Оптимізувати модель

Рівень: ціль інженера

Основна дійова особа: інженер із ЗВ

Тригер: інженер запустив аналізатор.

Головний успішний сценарій:

1. Інженер відкриває вкладку кореляційного аналізу та перевіряє нормальність розподілу значень показників.
2. Інженер запускає процес кореляційного аналізу.
3. Система розраховує коефіцієнти кореляції та виводить результати на екран.
4. Інженер визначає пари пов'язаних показників та переходить на вкладку регресійного аналізу.
5. Інженер обирає форму залежності (крок та початкові значення коефіцієнтів, якщо потрібно) та запускає процес аналізу.
6. Система виводить графічні та аналітичні результати розрахунків.
7. Інженер може продовжити аналіз або вийти з додатку.

Розширення:

1а. Розподіл значень показників не є нормальним.

1. Інженер обирає вид та значення коефіцієнту математичного перетворення ряду значень показника.
2. Система будує оновлену гістограму розподілу значень показників, а також виводить значення основних статистичних коефіцієнтів.
3. Інженер оцінює нормальність отриманого розподілу та зберігає його в разі необхідності.
4. Система запам'ятовує нові значення показника.

Таким чином, розглянуті основні варіанти використання програмної системи управління ЗВ. Для повного опису необхідно детальніше розглянути підсистеми аналізатора.

3.3 Підсистеми аналізатора

Збір даних та підсистема імпорту і початкової обробки результатів опитування. У першому розділі було доведено важливість оцінки відповідності ПП заданому рівню ЗВ на основі відгуків користувачів. Для цього необхідно було організувати процес збору, зберігання та аналізу їхніх оцінок.

У рамках даної дипломної роботи інформація про ЗВ збирається за допомогою опитувань – одного з найпоширеніших методів збору первинних даних. Опитування складається з ряду закритих запитань, на які користувачі відповідають кількісно, застосовуючи порядкову та інтервальну шкали.

Особливості програмної реалізації веб-сервісу були детально описані в підрозділі 3.1. На рис. 3.7 представлена схема послідовності, яка ілюструє процес взаємодії користувача з програмною системою під час збору даних.

Шаблон анкети, який заповнюється автоматично з БД та відправляється користувачу, включає набір атрибутів, показників, метрик і питань для оцінки цих метрик відповідно до ієрархічної моделі ЗВ. Перелік критеріїв ЗВ, що використовуються при практичній реалізації методу управління ЗВ, наводиться в розділі 4. Після завершення опитування інженер з ЗВ може розпочати первинну обробку отриманих даних за допомогою підсистеми завантаження та первинного аналізу.

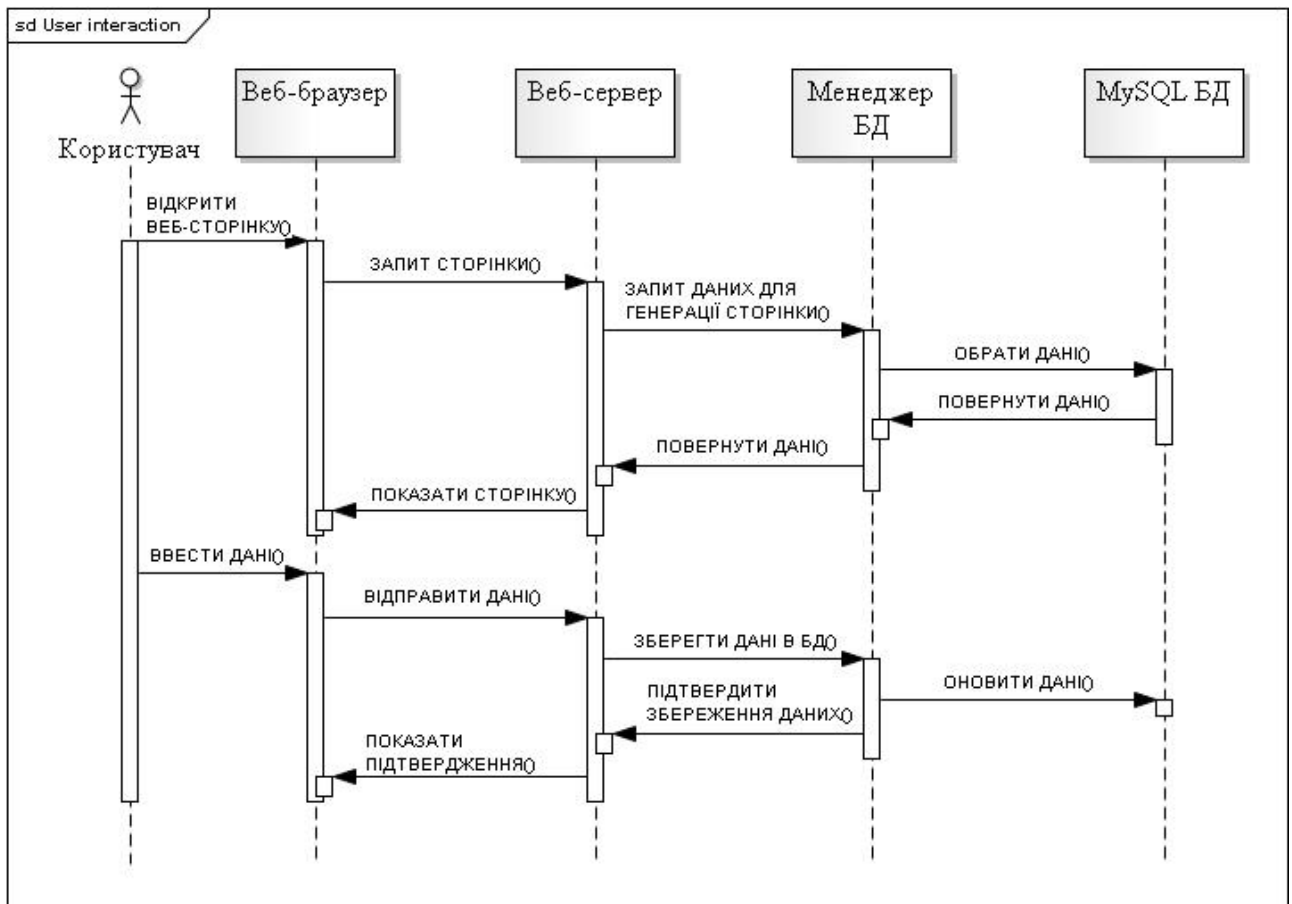


Рис. 3.7 Етап збору даних

Меню «Аналіз статистичних показників» призначена для перегляду розподілу оцінок та рангів, а також для обчислення основних статистичних показників, таких як математичне сподівання, медіана, дисперсія, коефіцієнти ексцесу та асиметрії. Для цього інженер зі ЗВ має натискати кнопку «Аналіз».

Підсистема через інтегровані компоненти контролера, що взаємодіють через інтерфейс, отримує дані про ранги метрик, показників, атрибутів і оцінки критеріїв ЗВ з моделі (клас ModelController). Далі проводить розрахунки статистичних коефіцієнтів, а також будує двовимірну гістограму, що відображає розподіл значень оцінок та рангів (рис. 3.8). Результати аналізу виводяться у числовому та графічному форматі.

Міри центральної тенденції відображають середні значення, навколо якого згруповані оцінки користувачів. Дисперсія ж є показником розсіювання, який характеризує, наскільки сильно варіюють значення навколо центрального значення.

Переглянувши ці статистичні дані, інженер з ЗВ може зробити попередній висновок щодо якості зібраних відгуків і у разі виявлення не відповідностей чи неналежної якості даних, може рекомендувати повторне проведення опитування та корекцію результатів. [11, с. 301]

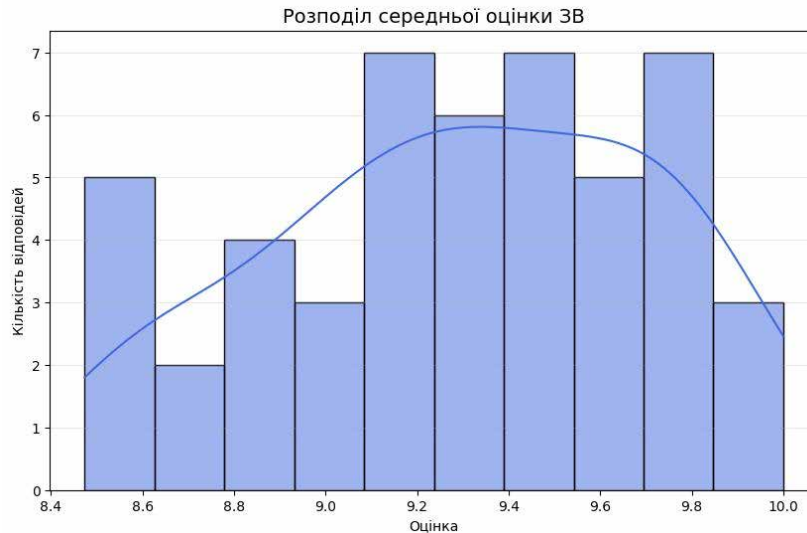


Рис. 3.8 Гістограма розподілу значень

Підсистема аналізу взаємозв'язків між показниками ЗВ.

Як було зазначено у другому розділі, важливим аспектом при вирішенні задачі оптимізації ЗВ є врахування взаємозалежних змінних, зокрема показників. Підсистема побудови зв'язків у математичній моделі управління ЗВ дозволяє вирішувати завдання виявлення залежностей між показниками та побудови лінії регресії, а також надання емпіричних залежностей. Як і в попередніх випадках, ініціація відповідних функцій системи здійснюється інженером зі ЗВ через користувацький інтерфейс. Інженер має змогу виконувати наступні операції:

- Аналіз розподілу значень показників ЗВ.
- Застосування математичних перетворень для приведення розподілу до нормальної форми, якщо це необхідно.
- Оцінка коефіцієнтів кореляції за результатами розрахунку.
- Визначення залежності показників та вибір типу регресій, з можливістю введення початкових значень коефіцієнтів і кроку.

- Оцінка емпіричної залежності та коригування форми регресії за необхідності для покращення точності наближення. [8]

Підсистема за допомогою інтегрованих компонентів контролера, що містять інтерфейс, отримує оцінки користувачів і здійснює відповідні розрахунки (рис. 3.9). Результати розрахунків виводяться у вигляді звітів, що містять коефіцієнти кореляції та статистичні дані, які оцінюють точність проназування, а також графічно відображується лінія регресії. Після того, як інженер з ЗВ застосовує необхідні перетворення для приведення розподілу до нормального виду, підсистема автоматично зберігає зміни у БД.

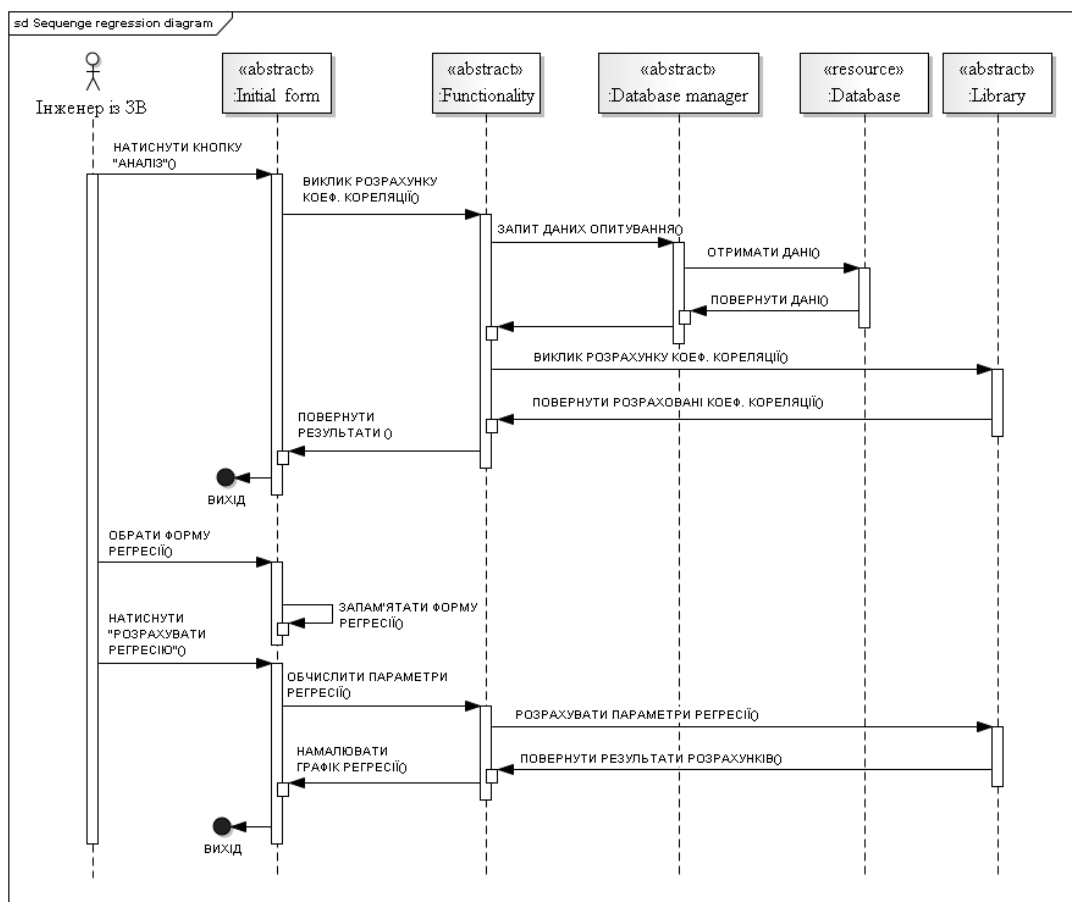


Рис. 3.9 Побудова зв'язків між показниками ЗВ

Підсистема розрахунку варіантів підвищення ЗВ.

Розроблена підсистема виконує своє основне функціональне призначення. Відповідальність за коректність і ефективність роботи підсистеми покладається на всі інші компоненти системи. Алгоритми та послідовність кроків для

прийняття рішень, на яких ґрунтується підсистема, детально описані у другому розділі.

У рамках підсистеми реалізовано розрахунок варіантів змін k показників (де $k = 5$), які дозволяють досягти заданого рівня ЗВ згідно з розробленим методом. Окрім цього, передбачено функціонал введення формули залежності між показниками, що враховується при виборі оптимального варіанту для забезпечення необхідного рівня ЗВ.

- ПЗ надає наступні можливості для роботи з підсистемою:
- Введення формули, яка визначає залежність одного показника від іншого.
- Задання кроку перебору та встановлення бажаного значення ЗВ.
- Перегляд варіантів змін показників для досягнення оптимального рівня ЗВ. [11, с. 302]

Висновки по розділу

У результаті виконаного дослідження та розробки ПС управління ЗВ було досягнуто кілька важливих результатів. Зокрема, була розроблена система, яка автоматизує процес збору, обробки та аналізу даних, отриманих від користувачів та експертів. Врахування цих даних дозволяє більш точно оцінювати поточний рівень ЗВ програмного продукту та прогнозувати варіанти його підвищення.

Основною особливістю запропонованої системи є використання математичних моделей та статистичних методів для аналізу залежностей між показниками, що дозволяє з високою точністю визначити оптимальні варіанти змін для досягнення заданого рівня ЗВ. Водночас, система забезпечує зручний інтерфейс для інженерів зі ЗВ, що дозволяє проводити необхідні налаштування та коригування на різних етапах процесу.

Крім того, система містить потужні функціональні можливості для аналізу і візуалізації даних, включаючи побудову гістограм, графіків та розрахунок статистичних коефіцієнтів, що допомагає інженерам отримувати чіткі і зрозумілі

результати. Це дозволяє швидко приймати обґрунтовані рішення щодо оптимізації характеристик ЗВ і підвищення якості програмних продуктів.

Загалом, реалізація запропонованої ПС є важливим кроком у напрямку автоматизації та удосконалення процесу управління ЗВ, що дає змогу суттєво покращити ефективність роботи з ПП та забезпечити їх високу якість на всіх етапах життєвого циклу.

РОЗДІЛ IV

ЗАСТОСУВАННЯ ПРОГРАМНОЇ СИСТЕМИ УПРАВЛІННЯ ЗРУЧНІСТЮ ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ

4.1 Вибір критеріїв зручності використання та опис досліджуваного програмного продукту

Одним із ключових етапів управління ЗВ є створення моделі, яка дозволяє оцінити відповідність ПП заданому рівню ЗВ. Для цього була обрана ієрархічна модель, заснована на інтегрованій оцінці зважених показників ЗВ. У процесі формування поточних значень атрибутів, показників і метрик проводиться опитування користувачів, які надають числові оцінки характеристикам ЗВ, а також ранжування експертів для визначення ваги критеріїв. Це дозволяє створити точну картину ефективності та ЗВ ПП.

Особливу увагу потрібно приділити формуванню переліку критеріїв ЗВ, який має бути якомога наповненим і повинен дозволяти експертам і користувачам дати об'єктивну оцінку продукту. Цей список критеріїв може бути адаптований під кожен конкретний ПП і повинен уточнюватись експертами.

Використовуючи основу моделі QUIM декомпонуюємо атрибути ЗВ на окремі показники:

1. **Часова характеристика (*Time Behavior*)** – здатність ПП ефективно витратити час на виконання заданих функцій.
2. **Привабливість (*Attractiveness*)** – візуальне приваблення користувача через графічний дизайн і використання кольору, що впливає на інтерес до продукту.
3. **Приємність (*Likeability*)** – чуттєве сприйняття і загальне враження користувача від роботи з програмою.
4. **Гнучкість (*Flexibility*)** – здатність інтерфейсу адаптуватися під індивідуальні вподобання користувача.
5. **Керування користувачем (*User Guidance*)** – наявність зворотного зв'язку на помилки, що підвищує ефективність роботи з ПП.

6. Мінімальна дія (Minimal Action) – здатність програми зменшувати кількість кроків, необхідних для досягнення результату.

7. Мінімальне навантаження пам'яті (Minimal Memory Load) – зменшення необхідної кількості інформації, яку користувач має запам'ятовувати для виконання завдання.

8. Узгодженість (Consistency) – узгодженість функціональності та дизайну інтерфейсу на рівні як внутрішніх елементів, так і у відповідності до зовнішніх стандартів.

9. Інформативність (Self-Descriptiveness) – здатність програми чітко передавати своє призначення та забезпечувати підтримку користувача в процесі роботи.

10. Правильність (Accuracy) – здатність програми надавати точні результати та виконувати правильні операції.

11. Відгук (Feedback) – швидкість та ефективність відгуку програми на дії користувача.

12. Стійкість до помилок (Fault-Tolerance) – здатність програми працювати при виникненні помилок або збоїв.

13. Зручність читання (Readability) – легкість сприйняття і читання контенту, особливо в веб-додатках.

14. Контрольованість (Controllability) – ступінь відчуття користувачем контролю над програмою.

15. Навігація (Navigability) – зручність переміщення користувача між різними елементами програми, особливо для веб-сайтів.

16. Демонстрації – наявність наочних демонстрацій функціональності програми.

17. Простота (Simplicity) – усунення зайвих елементів інтерфейсу без втрати важливої інформації.

18. Знайомість (Familiarity) – наявність знайомих користувачеві елементів інтерфейсу, що полегшує навчання та використання програми.

19. Керівництво з використання – змістовність і ефективність інструкцій з використання програмного продукту.

20. Довідникова служба – зміст та ефективність довідкової інформації.

Взаємодія між атрибутами та показниками ЗВ зображена у таблиці 4.1. Для кожного з показників розраховуються метрики, які оцінюються на основі простих підрахунків або за допомогою спеціальних формул. У загальному списку нараховується 124 метрики, що охоплюють різноманітні аспекти взаємодії користувача з ПП. Для вимірювання часових характеристик, наприклад, використовуються: час витрачений на помилки, час вибору та час виконання завдання. Оцінка точності здійснюється через частоту помилок – чим менша частота, тим кращий рівень точності. [5, с. 283]

Таблиця 4.1

Взаємодія між атрибутами та показниками ЗВ

Показники	Атрибути					
	Операбельність	Захист від помилок користувача	Естетика користувацького інтерфейсу	Можливість навчання	Доступність	Розпізнавання сумісності
Часова характеристика	+					
Привабливість			+			
Приємність			+			
Гнучкість			+		+	
Мінімальна дія			+	+	+	
Мінімальне навантаження пам'яті			+	+	+	
Керування користувачем		+	+	+	+	
Узгодженість	+		+	+	+	
Інформативність				+	+	
Відгук	+					
Правильність		+				
Стійкість до помилок	+	+	+			
Зручність читання					+	
Контрольованість	+		+		+	
Навігація					+	
Простота				+	+	

Знайомість				+		
Керівництво з використання				+		+
Демонстрації				+		+
Довідникова служба				+		+

Метрики для оцінки показників сумісності включають ефективність довідкової служби, яка вимірюється через два основні відношення: *ефективність* (відсоток функціональних елементів, вірно використаних після перегляду довідки) та *зрозумілість* (відсоток функціональних елементів, правильно інтерпретованих після використання довідки).

Метод управління ЗВ, описаний у другому розділі, включає якісну оцінку атрибутів і показників експертами, тоді як метрики виступають джерелом кількісних даних для оцінки цих характеристик.

Досліджуваний програмний продукт. У ході роботи над дипломним проектом було запропоновано метод управління ЗВ, який базується на ітераційному оцінюванні поточного рівня ЗВ в процесі розробки та супроводу ПП, а також на формуванні оптимального варіанту забезпечення бажаного рівня ЗВ, що визначається розробниками на початковому етапі. Для перевірки запропонованого методу та створення ПП було обрано веб-браузер **Chromium** з відкритим вихідним кодом, на основі якого розробляється популярний браузер Google Chrome. **Chromium** доступний для операційних систем Microsoft Windows (починаючи з XP SP2), Mac OS X (починаючи з 10.4.5), а також для різних дистрибутивів Linux, зокрема: Linux, Fedora, Gentoo Linux, openSUSE, Ubuntu, Debian та їх похідні.

Характеристики стабільної версії, що досліджується, наведено в таблиці 4.2.

Таблиця 4.2

Характеристики досліджуваного ПП

Виробник	Google
Назва	Chromium
Версія	140
Дата випуску	02.09.2025
Підтримувані платформи	Windows, Mac, Linux, ChromeOS

Двигун для відображення веб-сторінок	WebKit
Двигун JavaScript	V8

4.2 Емпірична оцінка та варіанти забезпечення ЗВ

З метою збору емпіричних даних було проведено опитування серед користувачів та експертів. У ролі користувачів виступала група з 49 осіб, серед яких працівники веб-студій *SamsonOs* та *Vervoe*. Їм було запропоновано оцінити міри ЗВ, що визначаються відповідними метриками, обраними для аналізу досліджуваного ПП. Функціонал для введення користувацьких ранжувань показників, атрибутів і метрик ЗВ реалізовано в програмній системі управління ЗВ, однак його використання не є обов'язковим, оскільки для визначення ваги критеріїв зазвичай застосовуються ранги експертів. Кількість опитаних користувачів дозволяє узагальнити отримані результати з довірчою ймовірністю 87% та довірчим інтервалом (похибкою) у 11%, за умови нормального розподілу.

Результати первинної обробки даних опитування показали, що в більшості випадків розподіли оцінок не містять викидів і є придатними для подальшого аналізу. (рис. 4.1).

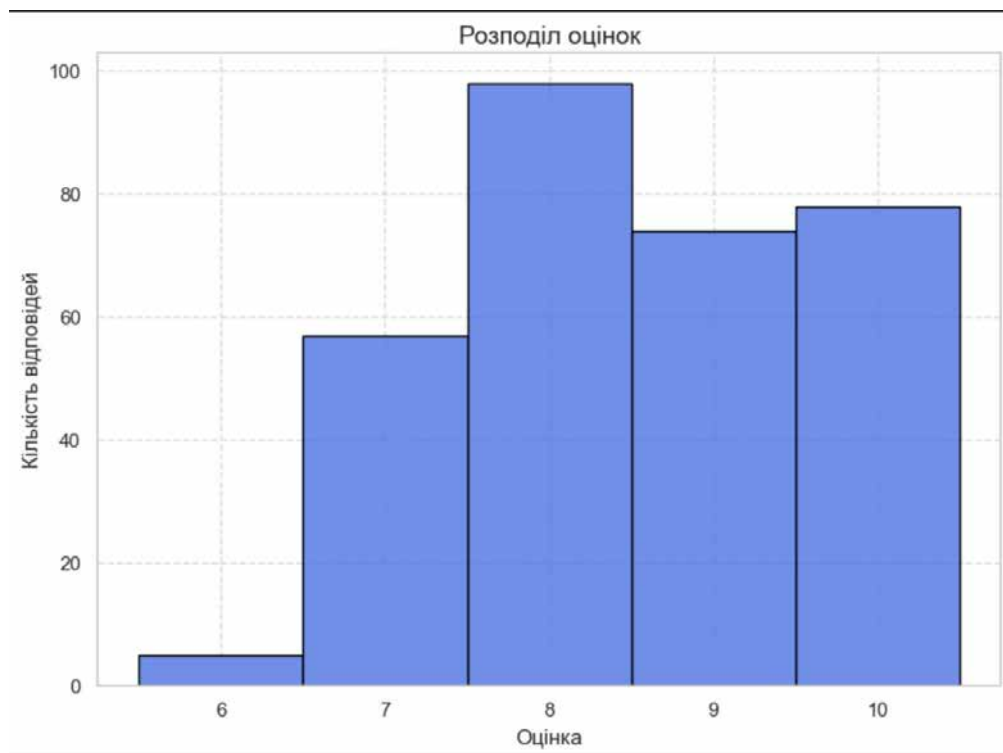


Рис. 4.1 Розподіл оцінок для метрик

Нижче наводиться коротка характеристика деяких метрик ЗВ для веб-браузера **Chromium**. Серед метрик, запропонованих для оцінки ЗВ, більшість розподілів мають ненормальний характер, при цьому значення медіани $Me = 8$ та дисперсією $0,37 < \sigma^2 < 2,8$, а отже, стандартним відхиленням $0,61 < \sigma < 1,67$. Таким чином похибка репрезентативності (середнього значення) наступна:

$$0,085 < m = \frac{\sigma}{\sqrt{n}} < 0,232, \text{ де } n=49 \text{ – кількість елементів вибірки.}$$

Відсоткові співвідношення оцінок, виставлених за іншими окремими метриками, представлені у вигляді кругових діаграм на рис. 4.2.

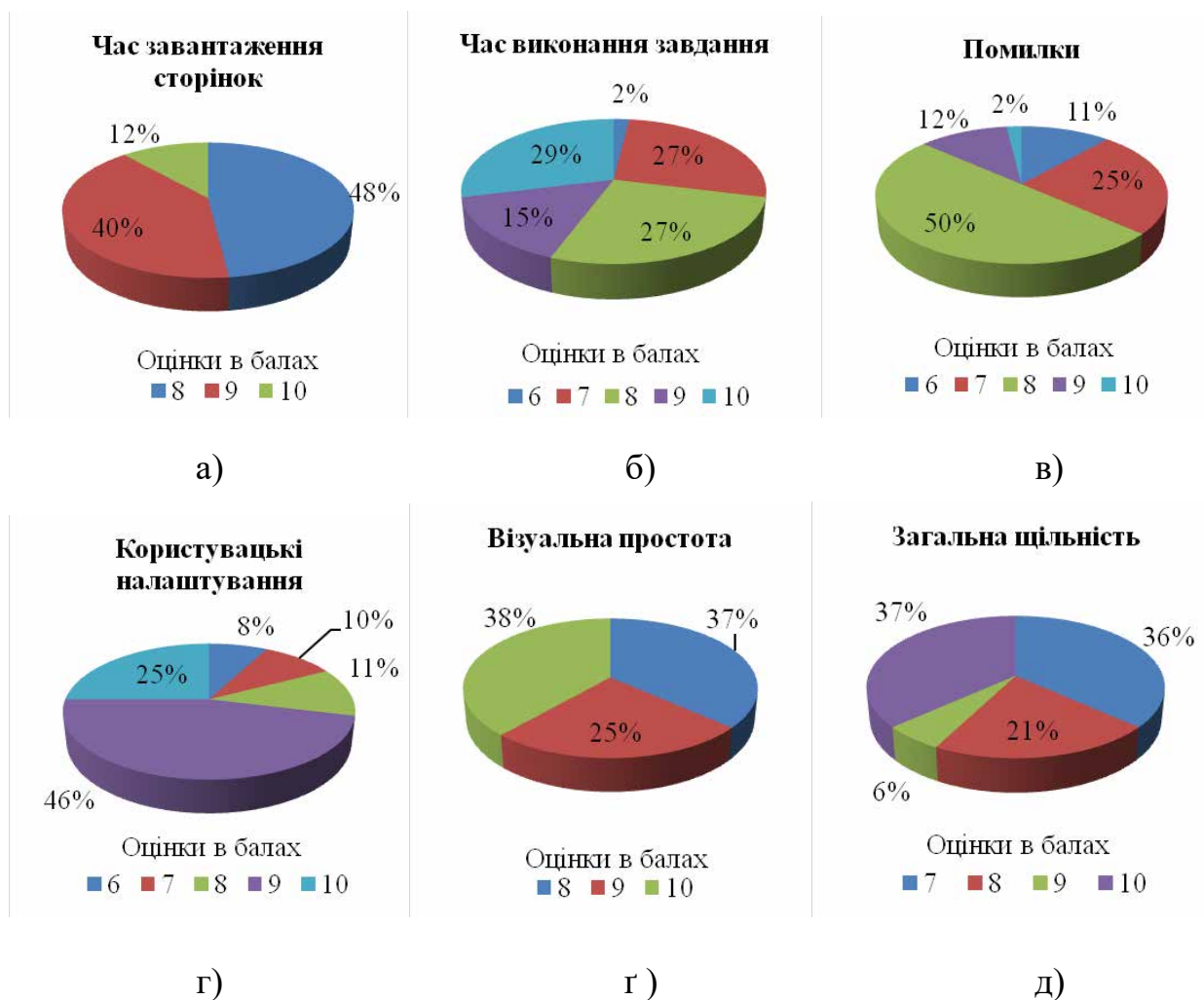


Рис. 4.2 Оцінка web-браузеру Chromium користувачами

Час завантаження сторінок був оцінений більшістю опитуваних вище 8 балів (рис. 4.2), $Me = 9$. Отримані результати узгоджуються з дослідженнями компанії Gomez, згідно з якими веб-сайти найшвидше завантажуються у браузері

Chrome (порівняно з Firefox, Safari та Internet Explorer). Середня швидкість завантаження сторінки на Chrome склала 2,8 секунди. Час виконання завдання (рис. 4.2б) та частота виникнення помилок (рис. 4.2в) були оцінені опитуваними трохи нижче. Для виявлення помилок, Google, проводить політику добровільного бета-тестування **Chromium**, а з 2011 року виплачує грошові винагороди за виявлені баги.

Щодо користувацьких налаштувань (рис. 4.2г), 46% опитуваних оцінили на 9 із 10 балів кількість функцій, які можна налаштувати відповідно до своїх уподобань. За результатами дослідження, інтерфейс **Chromium** був визнаний візуально простим (рис. 4.2г), а 36% користувачів оцінили загальну щільність на (рис. 4.2д), на 7 балів, тоді як 37% – на 10 балів.

Загалом, більшість метрик, за якими оцінювалася ЗВ веб-браузера **Chromium**, отримали високі бали.

За допомогою нашої системи ми побудували гістограму розподілу оцінок за метриками зручності використання рис. 4.3. На гістограмі ми можемо побачити що у нас є нетипові оцінки, так звані викиди.

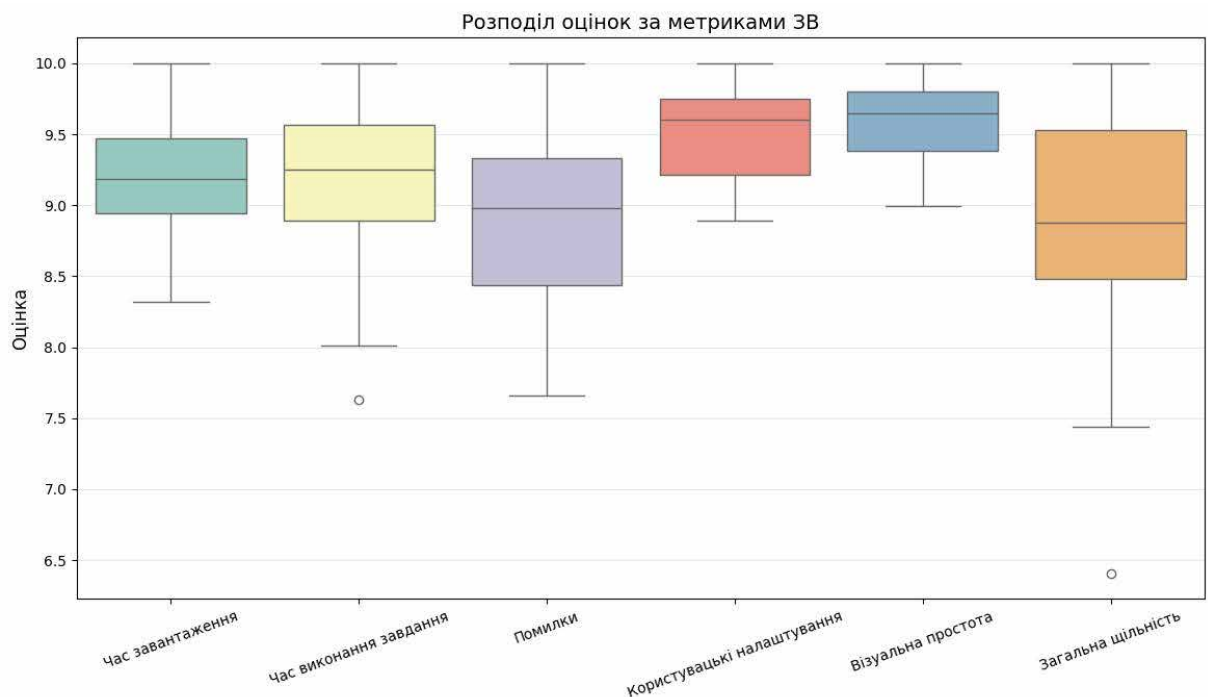


Рис. 4.3 Розподіл оцінок за метриками зручності використання

Наступним етапом дослідження було отримання експертних ранжувальних для подальшого розрахунку ваги метрик, показників та атрибутів ЗВ. Експертами виступили 7 фахівців у відповідній предметній області. Коефіцієнт конкордації для виставлених рангів атрибутів склав $W = 0,89$, що є значимим результатом і свідчить про високу ступінь узгодженості за шкалою Чеддока. Ваги критеріїв ЗВ, отримані на основі підсумкового ранжування, наведені в таблиці 4.3.

Таблиця 4.3

Вага атрибутів та показників зручності використання Chromium

Атрибут	Ранг	Вага	Показник	Ранг	Вага
Операбельність	3	0,19048	Часова характеристика	2	0,26667
			Узгодженість	1	0,33333
			Відгук	3	0,2
			Стійкість до помилок	4	0,13333
			Контрольованість	5	0,06667
Захист від помилок користувача	4	0,14286	Керування користувачем	3	0,16667
			Правильність	1	0,5
			Стійкість до помилок	2	0,33333
Естетика користувачького інтерфейсу	1	0,28571	Привабливість	4	0,13333
			Приємність	7	0,06667
			Гнучкість	6	0,08889
			Мінімальна дія	5	0,11111
			Мінімальне навантаження пам'яті	3	0,15556
			Узгодженість	2	0,17778
			Керування користувачем	1	0,2
			Стійкість до помилок	9	0,02222
Контрольованість	8	0,04444			
Можливість			Мінімальна дія	9	0,03636
Атрибут	Ранг	Вага	Показник	Ранг	Вага
навчання	2	0,2381	Мінімальне навантаження пам'яті	1	0,18182
			Керування користувачем	8	0,05455
			Узгодженість	6	0,09091
			Інформативність	3	0,14545
			Простота	4	0,12727
			Знайомість	5	0,10909
			Демонстрації	10	0,01818
			Довідникова служба	7	0,07273
Керівництво з використання	2	0,16364			
Доступність			Гнучкість	9	0,03636

	5	0,09524	Мінімальна дія	8	0,05455
			Мінімальне навантаження пам'яті	7	0,07273
			Керування користувачем	1	0,18182
			Узгодженість	10	0,01818
			Інформативність	4	0,12727
			Зручність читання	5	0,10909
			Контрольованість	3	0,14545
			Навігація	6	0,09091
			Простота	2	0,16364
Розпізнавання сумісності	6	0,04762	Керівництво з використання	1	0,5
			Демонстрації	3	0,16667
			Довідникова служба	2	0,33333

Згідно з оцінками користувачів та вагами критеріїв ЗВ, можна за формулою (2.1) розрахувати поточний рівень ЗВ для веб-браузера **Chromium**. Відповідно до отриманих даних, тіснота зв'язку між показниками визначалася за шкалою Чеддока: для аналізу бралися лише ті пари показників, що мають сильний зв'язок (табл. 4.4). Визначені зв'язки дозволяють оцінити, як зміна одного з показників впливає на інший. Результати дослідження показують, що всі залежності є прямо пропорційними, тобто при збільшенні (або зменшенні) значення незалежного показника, значення залежного також змінюється відповідно.

Таблиця 4.4

Емпіричні залежності між показниками ЗВ web-браузеру Chromium

Показник 1, (x)	Показник 2, (y)	Коефіцієнт рангової кореляції, r	Коефіцієнт детермінації, R^2	Оцінка за шкалою Чеддока
Часова характеристика, A[0]	Мінімальна дія, A[2]	0,71	0,5	Помірний зв'язок
Навігація, A[6]	Узгодженість, A[9]	0,8	0,64	Тісний зв'язок
Керування користувачем, A[14]	Інформативність, A[15]	0,84	0,71	Тісний зв'язок

Було спрогнозовано залежність «часу завантаження сторінки» від двох показників: «час виконання завдання» та «користувацькі зміни» (рис. 4.4). Площина регресії відображає прогнозовану середню залежність «часу завантаження» від цих двох незалежних показників. Фактичні точки на графіку представляють реальні оцінки користувачів. Як показує результат, площина регресії зростає у напрямку збільшення «часу виконання» та «користувацьких змін», що свідчить про те, що підвищення часу виконання завдання та більша активність користувача щодо налаштувань призводять до збільшення «часу завантаження сторінки». Отримана залежність дозволяє прогнозувати, як зміни в одних метриках впливають на інші, що є важливим для оптимізації зручності використання програмного продукту.

Таким чином, 3D-графік наочно демонструє вплив двох показників на третій, а присутність фактичних точок дозволяє оцінити точність зробленого прогнозу.

Вплив часу завантаження та часу виконання на загальну щільність

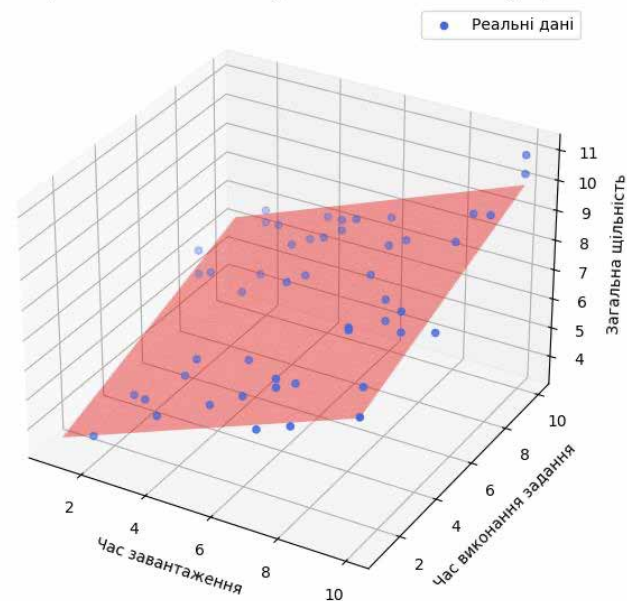


Рис. 4.4 Прогнозування на основі двох показників

У рамках перевірки роботи ПС перед нами не ставилась задача покращення показників ЗВ досліджуваного ПП та перерахунки оцінки ЗВ на наступних ітераціях. Це пояснюється тим, що моделі оцінки ЗВ базуються на формулі адитивної згортки. Таким чином, зміна показників згідно з обраним варіантом забезпечення ЗВ, а також підтвердження величини цієї зміни оцінками

користувачів на наступній ітерації, забезпечують досягнення заданого рівня ЗВ. Важливо зазначити, що в наступні стабільні версії web-браузера **Chromium** були внесені зміни, спрямовані на покращення згаданих показників:

а) Мінімальне навантаження пам'яті. Покращення було досягнуто за рахунок зміни зовнішнього вигляду вкладок, що пов'язано з метрикою візуальної когерентності. Користувач отримав можливість переміщатися між найбільш використовуваними основними меню такі як, найчастіше відвідуванні сторінки та встановленні додатки.

б) Гнучкість. Зміни стосуються налаштувань інтерфейсу – це дозволило створювати кілька різних користувацьких профілів з власним набором налаштувань.

Висновки до розділу

У процесі практичної перевірки розробленої ПС та перевірки дієвості запропонованого методу управління ЗВ було зібрано та проаналізовано результати опитувань користувачів і експертів на прикладі web-браузера **Chromium**. Зокрема, було виконано такі кроки.

1. Сформовано перелік критеріїв ЗВ, доступних для оцінювання користувачами, та визначено ті показники, щодо яких експерти можуть встановити ранги. Після цього проведено відповідні опитування.

2. Надано узагальнену характеристику частини отриманих метрик ЗВ для браузера Chromium із використанням базових методів описової статистики.

3. Визначено рівень поточної ЗВ досліджуваного ПП на основі зібраних даних.

4. Проаналізовано взаємозв'язки між окремими показниками ЗВ та встановлено форму цих залежностей, що дозволило оцінити, як зміна одного параметра може позначитися на значеннях інших.

5. Розглянуто можливі варіанти забезпечення необхідного рівня ЗВ для браузера Chromium за умов , а також зафіксованої кількості таких показників, що підлягають коригуванню, – $k = 5$.

ВИСНОВКИ

У дипломній роботі проведено комплексне дослідження теоретичних, методичних та прикладних аспектів управління ЗВ ПП та реалізовано ПС, призначену для підтримки процесів оцінювання та покращення ЗВ на основі даних користувачів й експертів.

У першому розділі визначено поняття ЗВ як інтегральної властивості ПП, що визначає ефективність, результативність та задоволеність користувача під час взаємодії з ПЗ. Проаналізовано сучасні підходи до оцінки ЗВ, їх обмеження та практичні труднощі застосування. Сформовано ієрархічну структуру критеріїв, показників та атрибутів ЗВ, яка стала основою для подальших аналітичних рішень та побудови вимог до ПС.

У другому розділі описано методичну базу оцінювання ЗВ, що спирається на готові математичні моделі, які дозволяють виконувати комплексну оцінку за кількома критеріями. Було застосовано методи описової статистики, ранжування критеріїв, визначення узгодженості експертних оцінок за медіаною Кемені, аналіз кореляцій та побудову регресійних залежностей між показниками. Використання готових математичних моделей забезпечило коректність обчислень та відтворюваність результатів без потреби у розробці додаткових формальних інструментів.

У третьому розділі сформовано функціональні та архітектурні вимоги до ПС, обґрунтовано вибір компонентів та засобів реалізації. Запропонована архітектура забезпечує збір, зберігання та багаторівневий аналіз даних. Створено структуру БД на основі MariaDB, реалізовано модулі імпорту, статистичної обробки, аналізу узгодженості, виявлення кореляцій та побудови регресійних моделей. Використання PHP та Python надало можливість поєднати веб-інтерфейс із потужним інструментарієм аналітичних обчислень.

У четвертому розділі здійснено практичне застосування розробленої ПС на прикладі web-браузера Chromium. Зібрано дані опитування користувачів та експертів, виконано обчислення інтегрального рівня ЗВ, виявлено залежності

між показниками, визначено критерії, що найбільше впливають на підсумкову оцінку. На основі отриманих результатів сформовано варіанти покращення ЗВ з урахуванням ресурсних обмежень і пріоритетності змін. Практичне застосування підтвердило коректність використаних моделей, стабільність роботи ПС та її прикладну корисність.

У результаті виконання роботи:

- систематизовано теоретичні положення щодо ЗВ та чинних методів її аналізу;
- використано готові математичні моделі комплексного оцінювання показників ЗВ;
- розроблено ПС, здатну автоматизувати збір, обробку та інтерпретацію даних;
- підтверджено можливість практичного застосування ПС для реальних ПП;
- продемонстровано потенціал використання аналітичних залежностей для обґрунтованого вибору напрямів покращення ЗВ.

Створена ПС може бути інтегрована в існуючі ІС управління якістю, використана на різних етапах життєвого циклу ПЗ та адаптована під інші ПП. Подальші дослідження доцільно спрямувати на розширення набору критеріїв ЗВ, використання сучасних методів прогнозного аналізу та вдосконалення рекомендаційних механізмів для прийняття інженерних рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глибовець М. М. Основи комп'ютерних алгоритмів / М. М. Глибовець. – К. : Вид. дім «КМ Академія», 2003. – 452 с.
2. Караванова Т. П. Інформатика: методи побудови алгоритмів та їх аналіз: обчислювальні алгоритми. – К. : Генеза, 2009. – 336 с.
3. Кузьменко Т. О. Якість програмного забезпечення та стандартизація. – К. : Кондор, 2020. – 276 с.
4. Мельник Л. О. Аналіз нефункціональних вимог до програмного забезпечення // Вісник НТУ «ХПІ». – 2019. – № 45. – С. 35–41.
5. Поліщук В. П. Інженерія програмного забезпечення. – К. : КНЕУ, 2018. – 512 с.
6. Собко Г. В. Управління вимогами в програмних проєктах. – Одеса : ОНАХТ, 2021. – 198 с.
7. Тимошенко І. В. Використання моделей якості ISO/IEC у процесі оцінювання програмного забезпечення // Вісник ХНУРЕ. Серія: Інформатика. – 2020. – № 2. – С. 102–109.
8. Федоров О. В. Комп'ютерна графіка: курс лекцій. – Донецьк, 2009.
9. Гобов Д. Визначення архітектури вимог до ІТ-рішення як бізнес-сервісу // Інформаційні технології та системи управління. – 2024.
10. Грицюк Ю. Аналіз вимог до програмного забезпечення. – К. : Наш Формат, 2022.
11. Дьяченко Ю. С., Маслов В. В. Програмна інженерія: методи та інструменти розробки. – Львів : ЛНУ, 2017. – 420 с.
12. Перевозчикова О. Л. Інформаційні системи і структури даних: навч. посіб. / О. Л. Перевозчикова. – К. : Вид. дім «Києво-Могилянська академія», 2007. – 304 с.
13. Коваленко Т. В. Методи оцінки якості програмних систем за нефункціональними характеристиками // Системні дослідження та інформаційні технології. – 2018. – № 1. – С. 55–62.

14. Сіренко О. М., Гриценко Н. П. Визначення метрик нефункціональних вимог при тестуванні програмних систем // Комп'ютерні науки та інформаційні технології. – 2021. – № 4. – С. 29–35.

15. Онищенко Д. І., Руденко І. В. Формалізація нефункціональних вимог у контексті Agile-розробки // Східноєвропейський журнал передових технологій. – 2016. – № 6(2). – С. 42–47.

16. Піх І., Сеньківська Н., Сеньківський Н. Ю., Білик О. З., Андріїв Р. Р., Андріїв Т. Р. Модель факторів якості тестування програмного забезпечення // Науковий вісник УАД. – 2024. – № 1(51). – С. 90–108.

17. ДСТУ ISO/IEC 25010:2016. Системи та програмна інженерія. Моделі якості системи та програмного забезпечення.

18. ДСТУ ISO/IEC 12207:2018. Системи та програмна інженерія. Процеси життєвого циклу програмного забезпечення.

19. ДСТУ ISO/IEC 9126:2014. Якість програмного забезпечення. Характеристики та метрики якості.

20. ДСТУ ISO/IEC 29148:2018. Інженерія систем та програмного забезпечення. Процеси управління вимогами.

21. ДСТУ 4163:2020. Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлення документів.

22. Функціональні та нефункціональні вимоги до програмного забезпечення [Електронний ресурс]. – Wezom Group, 2025. – Режим доступу: <https://wezom.com.ua/ua/blog/funktsionalni-ta-nefunktsionalni-vimogi-do-programnogo-zabezpechennya>.

ДОДАТКИ

```

<?php
// --- Список пунктів меню (аналог
TreeView) ---
$menu = [
    "Розпізнавання сумісності",
    "Здатність вчитися",
    "Естетичність інтерфейсу",
    "Доступність",
    "Захист від помилок",
    "Операбельність"
];
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Оцінка НФВ</title>
    <style>
        body { font-family: Arial; }
        .menu { width: 260px; float:left;
margin-right:30px; }
        .item { padding: 8px; background:#eef;
margin-bottom:5px; cursor:pointer; }
        .item:hover { background:#cdd; }
    </style>
</head>
<body>

<h2>Програмне забезпечення оцінки не
функціональних вимог</h2>

<div class="menu">
    <?php foreach ($menu as $m): ?>
        <div class="item"
onclick="location.href='metric.php?name=<
?urlencode($m)?>'><?=$m?></div>
    <?php endforeach; ?>
</div>

</body>
</html>

<?php
/**
* Мова: PHP 8+
*/

/** -----
-----
* КЛАС МОДЕЛІ МЕТРИК
* -----
-----
*/

class Metric
{
    public function __construct(
        public string $name,
        public string $description,
        public float $weight
    ){}
}

/** -----
-----
* КЛАС ДЛЯ ЗБЕРІГАННЯ
РЕЗУЛЬТАТІВ ОПИТУВАННЯ
* -----
-----
*/

class SurveyResult
{
    public function __construct(
        public int $userId,
        public array $scores // ['metricName'
=> score]
    ){}
}

/** -----
-----
* РЕПОЗИТОРІЙ МЕТРИК
* -----
-----
*/

class MetricRepository
{
    /** @var Metric[] */

```

```

private array $metrics = [];

public function add(Metric $metric): void
{
    $this->metrics[$metric->name] =
    $metric;
}

/** @return Metric[] */
public function all(): array
{
    return $this->metrics;
}
}

```

```

/** -----
-----

```

* РЕПОЗИТОРІЙ РЕЗУЛЬТАТІВ
ОПИТУВАННЯ

```

* -----
-----

```

```

*/
class SurveyRepository
{
    /** @var SurveyResult[] */
    private array $results = [];

    public function add(SurveyResult
    $result): void
    {
        $this->results[] = $result;
    }

    /** @return SurveyResult[] */
    public function all(): array
    {
        return $this->results;
    }
}

```

```

/** -----
-----

```

* СЕРВІС РОЗРАХУНКУ
ЗАГАЛЬНОГО ПОКАЗНИКА ЯКОСТІ

```

* -----
-----

```

```

*/
class EvaluationService
{
    public function __construct(
        private MetricRepository $metricRepo,
        private SurveyRepository $surveyRepo
    ){}
}

```

```

/**
 * Обчислення середнього значення по
 кожній метриці.
 */

```

```

public function calculateAverages(): array
{
    $metrics = $this->metricRepo->all();
    $results = $this->surveyRepo->all();

    $sum = [];
    $count = count($results);

    foreach ($metrics as $metric) {
        $sum[$metric->name] = 0;
    }

    foreach ($results as $res) {
        foreach ($res->scores as
        $metricName => $value) {
            if (isset($sum[$metricName])) {
                $sum[$metricName] += $value;
            }
        }
    }

    $avg = [];
    foreach ($sum as $metricName =>
    $value) {
        $avg[$metricName] = $count > 0 ?
        round($value / $count, 2) : 0;
    }

    return $avg;
}
}

```

```

/**
 * Обчислення інтегрального показника
 якості на основі ваг метрик.
 */
public function calculateQualityIndex():
float
{
    $metrics = $this->metricRepo->all();
    $avg = $this->calculateAverages();

    $sum = 0;
    $weightSum = 0;

    foreach ($metrics as $metric) {
        $sum += ($avg[$metric->name] ??
0) * $metric->weight;
        $weightSum += $metric->weight;
    }

    return $weightSum > 0 ? round($sum /
$weightSum, 3) : 0.0;
}
}

/** -----
-----
 * ПРИКЛАД ЗАСТОСУВАННЯ
 * -----
-----
 */
$metricRepo = new MetricRepository();
$metricRepo->add(new Metric('Usability',
'Зручність використання інтерфейсу',
0.4));
$metricRepo->add(new
Metric('Performance', 'Швидкодія
системи', 0.35));
$metricRepo->add(new
Metric('Accessibility', 'Доступність
функцій', 0.25));

$surveyRepo = new SurveyRepository();
$surveyRepo->add(new SurveyResult(1,
['Usability' => 8, 'Performance' => 9,
'Accessibility' => 7]));

```

```

$surveyRepo->add(new SurveyResult(2,
['Usability' => 9, 'Performance' => 8,
'Accessibility' => 8]));
$surveyRepo->add(new SurveyResult(3,
['Usability' => 7, 'Performance' => 7,
'Accessibility' => 6]));

$service = new
EvaluationService($metricRepo,
$surveyRepo);

$averages = $service->calculateAverages();
$qualityIndex = $service-
>calculateQualityIndex();

header('Content-Type: application/json');
echo json_encode([
    'average_scores' => $averages,
    'quality_index' => $qualityIndex
], JSON_PRETTY_PRINT);

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import
LinearRegression

# Встановлюємо зерно для
відтворюваності результатів
np.random.seed(42)

# Кількість користувачів
n = 50

data = csv.export('metric.csv');

# Обчислення середнього значення для
кожного користувача
data["Середня оцінка"] =
data.mean(axis=1)

# --- Побудова 3D-регресії ---
X = data[["Легкість навчання",
"Швидкість виконання"]]

```

```

y = data["Загальне задоволення"]

model = LinearRegression()
model.fit(X, y)

# Створюємо сітку для побудови
поверхні
x_surf, y_surf = np.meshgrid(np.linspace(2,
5, 20), np.linspace(2, 5, 20))
z_pred =
model.predict(np.array([x_surf.ravel(),
y_surf.ravel()]).T).reshape(x_surf.shape)

# --- Побудова 3D-графіка ---
fig = plt.figure(figsize=(6, 7))
ax = fig.add_subplot(111, projection='3d')

# Точки (реальні дані)
ax.scatter(data["Легкість навчання"],
data["Швидкість виконання"],
data["Загальне задоволення"],
color='blue', alpha=0.7, s=50,
label="Реальні дані")

# Поверхня (прогноз)
ax.plot_surface(x_surf, y_surf, z_pred,
color='red', alpha=0.4, label="Поверхня
регресії")

# Підписи осей
ax.set_xlabel("Легкість навчання (1–10)")
ax.set_ylabel("Швидкість виконання (1–
10)")
ax.set_zlabel("Загальне задоволення (1–
10)")
ax.set_title("Вплив легкості навчання та
швидкості виконання на загальне
задоволення", fontsize=12)

plt.tight_layout()
plt.show()

# Виведення коефіцієнтів моделі
print("Коефіцієнти регресії:")
print(f"  $\beta_1$  (легкість навчання):
{model.coef_[0]:.3f}")
print(f"  $\beta_2$  (логічність інтерфейсу):
{model.coef_[1]:.3f}")
print(f"  $\alpha$  (вільний член):
{model.intercept_:.3f}")

```