

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет кораблебудування імені адмірала Макарова**

Навчально-науковий інститут комп'ютерних наук та управління проектами  
Кафедра програмного забезпечення автоматизованих систем  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»  
Завідувач кафедри

«\_\_\_» \_\_\_\_\_ 2024 р.

***КВАЛІФІКАЦІЙНА РОБОТА***  
**на здобуття ступеня вищої освіти «магістр»**

**на тему:** Нелінійна регресійна модель для оцінювання розміру систем керування  
вмістом на мові PHP та розробка програми для її реалізації.

Виконав: студент 6151м групи  
\_\_\_\_\_  
(підпис, ПІБ)  Савчук Г. Ю.

Керівник роботи:  
\_\_\_\_\_  
(посада, науковий ступень вчене звання) к.ф.-м.н., доцент  
 Латанська Л.О.  
(підпис, ПІБ)

Миколаїв – 2024 р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет кораблебудування імені адмірала Макарова**

Навчально-науковий інститут комп'ютерних наук та управління проектами  
 Кафедра програмного забезпечення автоматизованих систем  
 Спеціальність 121 «Інженерія програмного забезпечення»  
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

\_\_\_\_\_ проф. С.Б.Приходько  
 (підпис)

«14» жовтня 2024 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**  
**на здобуття ступеня вищої освіти «магістр»**

Студенту Савчуку Григорію Юрійовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: «Нелінійна регресійна модель для оцінювання розміру систем керування вмістом на мові PHP та розробка програми для її реалізації».

Керівник роботи Латанська Л. О.

Затверджені наказом ректора № 1070 УЧ уч від «11» \_\_\_\_\_ 10 \_\_\_\_\_ 2024 р.

2. Термін подання роботи: 09.12.2024 р. \_\_\_\_\_

3. Вихідні дані по роботі: \_\_\_\_\_

4. Перелік питань, що належать до розробки (найменування розділів) \_\_\_\_\_

Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації.); Огляд літератури та обґрунтування необхідності проведення досліджень за обраною темою; Викладення результатів власних досліджень з висвітленням того нового, що пропонується; Проект програмного забезпечення; Організаційно-економічний розділ; Розділи з охорони праці та охорони навколишнього середовища; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)

5. Перелік презентаційних матеріалів 1.Тема 2. Мета та завдання дослідження, 3. Об'єкт та предмет дослідження, 4. Методи дослідження, 5. Наукова новизна одержаних результатів, 6. Практичне значення одержаних результатів, 7. Особистий внесок здобувача 8. Апробація результатів досліджень та публікації, 9. Аналіз сучасних методів оцінювання розміру веб-застосунків, 10. Визначення необхідності проведення дослідження 11. Емпіричні дані метрик CMS на PHP, 12. Нормалізація даних та вилучення викидів, 13. Нелінійна регресійна модель, 14. Лінійна регресійна моделі, 15. Порівняння лінійної та нелінійної моделей, 16. Порівняння моделей для різних типів проектів, 17. Use-case діаграма застосунку для оцінювання CMS на PHP, 18. Діаграма діяльності застосунку для оцінювання CMS на PHP, 19. Діаграма класів застосунку для оцінювання CMS на PHP, 20. Діаграма послідовності для варіанту використання «Виконати оцінювання розміру CMS», 21. Інтерфейс користувача застосунку для оцінювання CMS на PHP, 22. Висновки

## 6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|-------------------------------------------|----------------|------------------|
|        |                                           | завдання видав | завдання прийняв |
|        |                                           |                |                  |
|        |                                           |                |                  |
|        |                                           |                |                  |
|        |                                           |                |                  |
|        |                                           |                |                  |

7. Дата видачі завдання 14.10.2024 р.**КАЛЕНДАРНИЙ ПЛАН**

| Номер | Назва етапів роботи                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Терміни виконання | Примітка                                                                |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------------------------------------------------------|
| 1.    | Підготовка розділу Вступ                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 16.10.2024        | НС*                                                                     |
| 2.    | Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою                                                                                                                                                                                                                                                                                                                                                                    | 18.10.2024        | НС*                                                                     |
| 3.    | Підготовка розділу (ів) з результатів власних досліджень                                                                                                                                                                                                                                                                                                                                                                                                                       | 21.10.2024        | НС*                                                                     |
| 4.    | Підготовка розділу з проекту програмного забезпечення                                                                                                                                                                                                                                                                                                                                                                                                                          | 27.11.2024        |                                                                         |
| 5.    | Підготовка організаційно-економічного розділу                                                                                                                                                                                                                                                                                                                                                                                                                                  | 29.11.2024        |                                                                         |
| 6.    | Підготовка розділу з охорони праці                                                                                                                                                                                                                                                                                                                                                                                                                                             | 02.12.2024        |                                                                         |
| 7.    | Підготовка розділу з охорони навколишнього середовища                                                                                                                                                                                                                                                                                                                                                                                                                          | 04.12.2024        |                                                                         |
| 8.    | Підготовка розділу Висновки                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 05.12.2024        |                                                                         |
| 9.    | Оформлення списку використаних джерел та додатків                                                                                                                                                                                                                                                                                                                                                                                                                              | 06.12.2024        |                                                                         |
| 10.   | Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.) | 09.12.2024        | не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку) |
| 11.   | Підготовка презентації та доповіді                                                                                                                                                                                                                                                                                                                                                                                                                                             | 13.12.2024        |                                                                         |
| 12.   | Попередній захист роботи на засіданні кафедри ПЗАС                                                                                                                                                                                                                                                                                                                                                                                                                             | 16.12.2024        |                                                                         |
| 13.   | Подання на кафедру ПЗАС електронних версії наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді                                                                                                                                                                                                                                            | 29.12.2024        |                                                                         |

\* - за результатами наукового стажування (НС), яке було з 02.09.2024 по 13.10.2024 р.

Студент

(підпис)

Савчук Г. Ю.

(ПБ)

Керівник роботи

(підпис)

Латанська Л. О.

(ПБ)

## РЕФЕРАТ

Савчук Григорій Юрійович

«Нелінійна регресійна модель для оцінювання розміру систем керування вмістом на мові PHP та розробка програми для її реалізації»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2024 р.

**Обсяг роботи:** 89 стор., 16 табл., 8 рис., 34 використаних джерел, 5 додатків

**Актуальність теми:** необхідність більш точного оцінювання розміру CMS на мові PHP через відсутність побудованих моделей для цього типу проєктів.

**Мета дослідження.** Метою роботи є підвищення достовірності оцінювання розміру систем керування вмістом на мові PHP за допомогою нелінійної регресійної моделі.

**Об'єкт дослідження:** процес оцінювання розміру систем керування вмістом на мові PHP.

**Предмет дослідження:** нелінійні регресійні моделі для оцінювання розміру систем керування вмістом на мові PHP.

**Методи дослідження:** методи ймовірнісного аналізу, математичної статистики, регресійного аналізу, оптимізації, об'єктно-орієнтованого програмування, чисельні методи.

**Наукова новизна роботи** полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінки розміру PHP-систем керування вмістом. Це досягнуто через використання одновимірного нормалізуючого перетворення десяткового логарифму та врахування викидів регресії. Порівняно з існуючими моделями для фреймворків Symfony, CakePHP і Yii, запропонована модель демонструє вищий множинний коефіцієнт детермінації, нижчу середню відносну похибку та більший відсоток прогнозованих результатів.

**Практичне значення одержаних результатів.** Результатом кваліфікаційної роботи стала розробка програми, яка дозволяє будувати нелінійні регресійні моделі

та оцінювати розмір РНР-орієнтованих систем управління вмістом. Інструмент надає можливість проводити оцінювання розміру таких застосунків ще на початкових етапах розробки.

**Апробація результатів роботи.** Основні положення й результати кваліфікаційної роботи оприлюднені на V Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи – 2024» (м. Миколаїв, 30-31 жовтня 2024 року).

**Публікації:** Основні положення й результати кваліфікаційної роботи опубліковано у 1 науковій праці – тезах конференції.

**Ключові слова:** ЛОС оцінювання, кількості рядків коду, нелінійні регресійні моделі, нормалізуючі перетворення, CMS, РНР.

## ABSTRACT

Savchuk Hryhorii Yuriiovych

«A nonlinear regression model for estimating the size of PHP-based content management systems and development the software for its implementation»

Qualification work for obtaining a master's degree in specialty 121 – "Software Engineering". Admiral Makarov National University of Shipbuilding. Mykolaiv, 2024.

**Volume:** 89 p., 16 tables, 8 figures, 34 references, 5 appendices.

**Topic Relevance:** is the issue of more precise estimation of the size of PHP-based content management systems due to the lack of built regression models for this framework.

**Research goal.** The goal is to increase the confidence of estimation the size of PHP-based content management systems using a nonlinear regression model.

**Object of research:** the process of estimating the size of PHP-based content management systems.

**Subject of research:** the nonlinear regression models for estimating the size of PHP-based content management systems.

**Methods of research:** probabilistic analysis, mathematical statistics, regression analysis, optimization, object-oriented programming, and numerical methods.

**Scientific contribution:** the three-factor nonlinear regression model for estimating the size of PHP-based content management systems based on a one-dimensional normalizing transformation of the decimal logarithm and regression outliers, was improved. This three-factor non-linear regression model, compared to the existing non-linear regression models for PHP frameworks CakePHP, Yii and Symfony, has a higher value of the multiple coefficient of determination  $R^2$ , a lower value of the average value of the relative error  $MMRE$  and a higher value of the percentage of predicted results  $PRED(0.25)$ .

**Practical value of obtained results.** The program that implements build nonlinear regression model for estimating the size of PHP-based content management systems was

developed. The program allows to estimate the size of those web application in early project stages.

**Approbation of the thesis results.** The 5rd All-Ukrainian scientific-practical Internet conference "Information technologies: models, algorithms, systems (ITMAS 2024)" (Mykolaiv, October 30-31, 2024).

**Publications.** The main provisions and results of the qualification work are published in 1 scientific work – theses of the conference.

**Keywords:** LOC estimation, code number lines, non-linear regression models, normalizing transformations, CMS, PHP.

## ЗМІСТ

|                                                                                                                                                                    |           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І<br/>ТЕРМІНІВ .....</b>                                                                                | <b>10</b> |
| <b>ВСТУП .....</b>                                                                                                                                                 | <b>11</b> |
| <b>1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ І<br/>ВИЗНАЧЕННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ.....</b>                                      | <b>14</b> |
| 1.1 Аналіз сучасних методів оцінювання розміру веб-застосунків .....                                                                                               | 14        |
| 1.2 Обґрунтування необхідності проведення досліджень.....                                                                                                          | 19        |
| 1.3 Висновки до розділу 1 .....                                                                                                                                    | 20        |
| <b>2 СТВОРЕННЯ НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ<br/>СИСТЕМ КЕРУВАННЯ ВМІСТОМ НА МОВІ PHP .....</b>                                             | <b>21</b> |
| 2.1 Перевірка даних для побудови нелінійної регресійної моделі для оцінювання розміру<br>CMS на PHP на наявність мультиколінеарності та викидів.....               | 21        |
| 2.2 Побудова нелінійної регресійної моделі для оцінювання розміру CMS на PHP .....                                                                                 | 27        |
| 2.3 Висновки до розділу 2 .....                                                                                                                                    | 34        |
| <b>3 РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ<br/>РОЗМІРУ СИСТЕМ КЕРУВАННЯ ВМІСТО НА МОВІ PHP .....</b>                                            | <b>35</b> |
| 3.1 Формулювання завдання на створення ПЗ для оцінювання розміру CMS на PHP.....                                                                                   | 35        |
| 3.2 Розробка ескізного проекту ПЗ для оцінювання розміру CMS на PHP.....                                                                                           | 35        |
| 3.2.1 Розробка моделі сценаріїв використання ПЗ для оцінювання розміру CMS на PHP..<br>.....                                                                       | 36        |
| 3.2.2 Розробка сценарію виконання ПЗ для оцінювання розміру CMS на PHP за<br>допомогою діаграми діяльності .....                                                   | 37        |
| 3.3 Розробка технічного проекту ПЗ для оцінювання розміру CMS на PHP .....                                                                                         | 39        |
| 3.3.1 Статична модель ПЗ для оцінювання розміру CMS на PHP у вигляді діаграми<br>класів .....                                                                      | 39        |
| 3.3.2 Специфікації класів ПЗ для оцінювання розміру CMS на PHP.....                                                                                                | 40        |
| 3.3.3 Розробка динамічного зображення ПЗ для оцінювання розміру CMS на PHP у<br>вигляді діаграм послідовностей .....                                               | 41        |
| 3.4 Розробка робочого проекту ПЗ для оцінювання розміру CMS на PHP .....                                                                                           | 42        |
| 3.4.1 Кодування ПЗ для оцінювання розміру CMS на PHP.....                                                                                                          | 42        |
| 3.4.2 Тестування ПЗ для оцінювання розміру CMS на PHP.....                                                                                                         | 43        |
| 3.4.3 Проведення випробувань ПЗ для оцінювання розміру CMS на PHP.....                                                                                             | 45        |
| 3.5 Висновки до розділу 3 .....                                                                                                                                    | 47        |
| <b>4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І<br/>ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ СИСТЕМ<br/>КЕРУВАННЯ ВМІСТОМ НА МОВІ PHP.....</b> | <b>48</b> |
| 4.1 Розрахунок витрат на створення й експлуатацію програмного забезпечення .....                                                                                   | 48        |
| 4.2 Економічна ефективність розробки та впровадження програмного забезпечення .....                                                                                | 50        |

|                                                                                             |           |
|---------------------------------------------------------------------------------------------|-----------|
| 4.3 Висновки до розділу 4 .....                                                             | 51        |
| <b>5 ОХОРОНА ПРАЦІ.....</b>                                                                 | <b>52</b> |
| 5.1 Аналіз шкідливих та небезпечних факторів в офісі фірми .....                            | 53        |
| 5.2 Заходи щодо зменшення впливу шкідливих факторів роботи за персональним комп'ютером..... | 55        |
| <b>6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....</b>                                              | <b>57</b> |
| 6.1 Забруднення навколишнього середовища .....                                              | 58        |
| 6.2 Розробка заходів щодо зменшення забруднення .....                                       | 59        |
| <b>ВИСНОВКИ .....</b>                                                                       | <b>62</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>                                                     | <b>63</b> |
| <b>ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....</b>                                                   | <b>68</b> |
| <b>ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ .....</b>                                   | <b>72</b> |
| <b>ДОДАТОК В – ІНСТРУКЦІЯ КОРИСТУВАЧА.....</b>                                              | <b>75</b> |
| <b>ДОДАТОК Г – ТЕКСТ ПРОГРАМИ .....</b>                                                     | <b>79</b> |
| <b>ДОДАТОК Д – ОПИС ПРОГРАМИ.....</b>                                                       | <b>88</b> |

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

ПК – Персональний комп'ютер

ПЗ – Програмне забезпечення

COCOMO – COnstructive COst Model

NOC – Number Of Classes

DIT – Depth of Inheritance Tree

KLOC – Thousand lines of code

LB – Lower Bound

UP – Upper Bound

LOC – Lines Of Code

MMRE – Mean Magnitude of Relative Error

MRE – Magnitude of Relative Error

PHP – Hypertext Preprocessor

PRED – Percentage of prediction

SMD – the Square Mahalanobis Distance

VIF – Variance Influence Factor

## ВСТУП

**Актуальність теми.** Розмір програмного забезпечення відіграє критичну роль у процесі його розробки, адже він тісно пов'язаний із витратами, зусиллями та ресурсами, необхідними для успішної реалізації проекту [1]. Відомості про розмір коду дозволяють використовувати моделі COCOMO, COCOMO II, ISBSG для прогнозування витрат і параметрів, пов'язаних із проектом [2-4]. Під поняттям "розмір" найчастіше мають на увазі кількість рядків вихідного коду.

Попри це, проектні менеджери та розробники стикаються з проблемою відсутності універсального підходу до оцінювання розміру ПЗ [5, 6]. Сучасні методи включають аналіз кількості коду, функціональних точок, функціональних і технічних характеристик [7]. Проте, через нестачу даних, ресурсів чи експертних знань, більшість із цих підходів виявляються обмеженими у використанні.

Системи керування вмістом (CMS) дозволяють користувачам без технічних навичок створювати та редагувати веб-контент. Вони складаються з модуля управління вмістом (CMA), який забезпечує зручний інтерфейс, і модуля доставки (CDA), що відповідає за публікацію даних. CMS допомагають зменшити витрати на підтримку сайтів і забезпечують одночасну роботу кількох користувачів через веб-інтерфейс. PHP, що лежить в основі багатьох популярних CMS, пропонує розробникам значні можливості завдяки відкритому коду, інтеграції з базами даних (MySQL, PostgreSQL) і підтримці масштабованості через фреймворки Laravel та Symfony. Це робить PHP чудовим вибором для створення динамічних CMS із можливістю їх подальшого розширення.

LOC-методом з використанням регресійних моделей для оцінювання розміру ПЗ, зокрема для мов PHP, Java та C++, C#, Kotlin, Visual Basic, часто базуються на метриках діаграм класів, що формуються на етапі проектування [8-12]. Хоча для PHP [13-16] і її фреймворків (Laravel, Symfony, CakePHP, CodeIgniter) [17-20] існують нелінійні регресійні моделі [13], їх достовірність у практичному

застосуванні залишається недостатньою для широкого використання, особливо в специфічних проектах, як-от CMS [19].

Для нормалізації даних при побудови моделі для оцінювання розміру систем керування вмістом на мові PHP обрано перетворення  $\log_{10}$ , яке давно зарекомендувало себе в оцінюванні програмного забезпечення (COCOMO, ISBSG). Це перетворення допомагає зменшити розмаїття даних, стабілізувати модель і покращити достовірність прогнозів.

**Мета та завдання дослідження.** Мета роботи полягає в підвищенні достовірності оцінювання розміру систем керування вмістом, створених на мові PHP, за допомогою нелінійної регресійної моделі.

Для того, аби досягти поставленої мети треба:

- Дослідити наявні моделі для оцінювання розміру програмних застосунків, розроблених на PHP
- Провести збір даних, необхідних для створення нелінійної регресійної моделі оцінювання розміру систем управління вмістом.
- Удосконалити нелінійну регресійну модель із застосуванням нормалізуючого перетворення.
- Створити програмний продукт, що реалізує розрахунки за допомогою розробленої моделі.

**Об'єкт дослідження:** процес оцінювання розміру систем керування вмістом на мові PHP.

**Предмет дослідження:** нелінійні регресійні моделі для оцінювання розміру систем керування вмістом на мові PHP.

**Методи дослідження:** методи ймовірнісного аналізу, математичної статистики, регресійного аналізу, оптимізації, об'єктно-орієнтованого програмування, чисельні методи.

**Наукова новизна роботи** полягає в удосконаленні трьохфакторної нелінійної регресійної моделі для оцінки розміру PHP-систем керування вмістом. Це досягнуто через використання одновимірного нормалізуючого перетворення десяткового логарифму та врахування викидів регресії. Порівняно з існуючими

моделями для фреймворків Symfony, CakePHP і Yii, запропонована модель демонструє вищий множинний коефіцієнт детермінації, нижчу середню відносну похибку та більший відсоток прогнозованих результатів.

**Практичне значення одержаних результатів.** Результатом кваліфікаційної роботи стала розробка програми, яка дозволяє будувати нелінійні регресійні моделі та оцінювати розмір PHP-орієнтованих систем управління вмістом. Інструмент надає можливість проводити оцінювання розміру веб-застосунків ще на початкових етапах розробки.

**Особистий внесок здобувача.** Ця кваліфікаційна робота є самостійною науковою працею, усі результати отримано автором особисто. У співавторській публікації [19] автору належить розробка нелінійної регресійної моделі для оцінки розміру систем керування вмістом на PHP, заснованої на нормалізації через десятковий логарифм.

**Апробація результатів роботи.** Основні положення й результати кваліфікаційної роботи опубліковані на V Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи – 2024» (м. Миколаїв, 30-31 жовтня 2024 року).

**Публікації:** Основні положення й результати кваліфікаційної роботи опубліковано у 1 науковій праці – тезах конференції.

# **1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ОЦІНЮВАННЯ РОЗМІРУ ВЕБ-ЗАСТОСУНКІВ І ВИЗНАЧЕННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ.**

## **1.1 Аналіз сучасних методів оцінювання розміру веб-застосунків**

Розмір програмного забезпечення відіграє критичну роль у процесі його розробки, адже він тісно пов'язаний із витратами, зусиллями та ресурсами, необхідними для успішної реалізації проекту [1]. Відомості про розмір коду дозволяють використовувати моделі COCOMO, COCOMO II, ISBSG для прогнозування витрат і параметрів, пов'язаних із проектом [2-4]. Під поняттям «розмір» найчастіше мають на увазі кількість рядків вихідного коду.

Попри це, проектні менеджери та розробники стикаються з проблемою відсутності універсального підходу до оцінювання розміру ПЗ. Сучасні методи включають аналіз кількості коду, функціональних точок, функціональних і технічних характеристик [5-7]. Проте, через нестачу даних, ресурсів чи експертних знань, більшість із цих підходів виявляються обмеженими у використанні.

Система керування вмістом (Content management system, CMS) – це ПЗ, завдяки якому кінцеві користувачі можуть створювати вміст веб-сайту та керувати ним. CMS розроблені, щоб полегшити керування вмістом для нетехнічних користувачів. Однією з ключових особливостей хорошої системи керування вмістом є те, що для створення чи зміни вмісту не потрібно кодувати. На CMS покладено обробку та перетворення у код нових елементів, тому користувачі можуть зосередитися на тому, що побачать відвідувачі веб-сайту, а не на тому, що відбувається за лаштунками.

Система управління вмістом складається з двох основних елементів. По-перше, це програма для керування вмістом (Content management application, CMA). CMA — це частина програми, яка дозволяє користувачам додавати вміст і керувати ним. Другим елементом є програма доставки вмісту (Content delivery application,

CDA). Це серверна програма, яка форматує вміст і робить його доступним для відвідувачів сайту.

Використання CMS дозволяє значно скоротити витрати на підтримку веб-сайту, оскільки відсутня необхідність у програмуванні. Проте, за потреби складного налаштування або проектування, можна звернутися до послуг веб-розробника. Ще однією перевагою CMS є підтримка співпраці: кілька користувачів можуть одночасно створювати та публікувати контент. Завдяки інтерфейсу, що працює через браузер, система доступна з будь-якого місця.

Історично склалось, що PHP є основою багатьох популярних систем управління вмістом, які використовуються сьогодні. Як потужна серверна мова сценаріїв, PHP забезпечує ключову функціональність, необхідну для створення динамічних та інтерактивних веб-сайтів, і є наріжним каменем у світі CMS. Незалежно від того, чи створюється невеликий персональний блог або великий сайт корпоративного рівня, PHP часто лежить в основі CMS, забезпечуючи основні операції, налаштування та гнучкість. Серед найпопулярніших CMS мовою PHP можна виділити WordPress, Drupal, Joomla, Magento. Зокрема, 43,7% усіх сайтів використовують WordPress [22], тоді як на мові PHP розроблено 75,4% усіх сайтів [23]. В цілому можна сказати, що більша частина популярних CMS розроблені саме мовою PHP.

Основні причини того, чому мова PHP домінує як мова розробки CMS:

- Відкрите програмне забезпечення – PHP є мовою з відкритим кодом, що означає, що вона безкоштовна у використанні, а велика й активна спільнота розробників постійно сприяє її розвитку. Це призвело до створення тисяч плагінів, тем і розширень для таких платформ CMS, як WordPress, Joomla, Drupal, Magento.

- Гнучкість і налаштування – PHP забезпечує гнучкість у створенні та налаштуванні платформ CMS. Завдяки доступу до базового PHP-коду розробники можуть створювати спеціальні функції, змінювати теми або навіть будувати абсолютно нові модулі, щоб задовольнити конкретні потреби бізнесу.

- Інтеграція з базами даних – PHP легко інтегрується з різними базами даних, такими як MySQL, PostgreSQL і SQLite, для зберігання та отримання контенту. Це

є критично важливим аспектом розробки CMS, оскільки більшість веб-сайтів значною мірою покладаються на взаємодію з базами даних для управління статтями, сторінками та мультимедійним контентом.

– Масштабованість – Сучасні PHP-фреймворки, такі як Laravel і Symfony, часто використовуються разом із традиційними платформами CMS для підвищення масштабованості та продуктивності. Поєднуючи PHP із цими фреймворками, розробники можуть створювати системи управління контентом, здатні справлятися з великим навантаженням, великими базами даних і складними архітектурами.

Розмір програмного забезпечення за LOC-методом оцінюється за допомогою численних регресійних моделей, як лінійних, так і нелінійних, які були створені для мов програмування, таких як C++, C#, Java, PHP, Kotlin, Visual Basic [8-16]. Основою для побудови цих моделей слугували метрики діаграми класів, оскільки саме на стадії проектування можна визначити концептуальну організацію класів. Для PHP, а також її популярних фреймворків Symfony, CakePHP, Yii, CodeIgniter і Laravel, розроблено кілька нелінійних моделей [17-20]. Однак, як було зазначено в роботі [19], використання існуючих моделей для оцінювання розміру ПЗ показало низьку ефективність, незважаючи на попередні припущення про їх потенційну достовірність. Вищезазначене працює й по відношенню до різних типів проектів, тобто, й для систем керування вмістом.

Для нормалізації даних у запропонованій моделі було обрано перетворення  $\log_{10}$ . Цей підхід уже довів свою ефективність у різних галузях оцінювання програмного забезпечення, включаючи відомі моделі COCOMO, ISBSG і COCOMO II. Перетворення  $\log_{10}$  часто використовується для зменшення розмаїття даних, переведення показників у шкалу, яка краще підходить для регресійного аналізу, та забезпечення стабільності моделі.

У випадку з PHP  $\log_{10}$  застосовувалося в кількох дослідженнях, спрямованих на побудову нелінійних регресійних моделей. Його використання обумовлене тим, що воно не лише дозволяє нормалізувати вихідні дані, але й забезпечує зручність при порівнянні моделей між собою. Це перетворення також

зменшує чутливість моделі до екстремальних значень, що є важливим для покращення достовірності прогнозів.

Лінійна регресійна модель для даних, що є нормальними, має вигляд [24]

$$Y = \hat{Y} + \varepsilon = b_0 + b_1X_1 + b_2X_2 + \dots + b_kX_k + \varepsilon, \quad (1.1)$$

де  $\hat{Y}$  – передбачення, отримане за лінійною регресією,  $b_i$  ( $i = \overline{0, k}$ ) – параметри лінійного рівняння регресії,  $X_i$  ( $i = \overline{1, k}$ ) – значення незалежної змінної,  $\varepsilon$  – гаусівська випадкова величина,  $k$  – кількість предикторів у моделі.

Довірчий інтервал лінійної регресії для даних, що є нормальними, має вигляд [24]

$$\hat{Y} \pm t_{\alpha/2, v} S_Y \left\{ \frac{1}{N} + (\mathbf{x}^+)^T [(\mathbf{X}^+)^T \mathbf{X}^+]^{-1} (\mathbf{x}^+) \right\}^{1/2}, \quad (1.2)$$

а інтервал передбачення [24]

$$\hat{Y} \pm t_{\alpha/2, v} S_Y \left\{ 1 + \frac{1}{N} + (\mathbf{x}^+)^T [(\mathbf{X}^+)^T \mathbf{X}^+]^{-1} (\mathbf{x}^+) \right\}^{1/2}, \quad (1.3)$$

де  $S_Y^2 = \frac{1}{v} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$ ;  $v = N - k - 1$ ;  $(\mathbf{X}^+)^T \mathbf{X}^+ - k \times k$  матриця

$$(\mathbf{X}^+)^T \mathbf{X}^+ = \begin{pmatrix} S_{X_1 X_1} & S_{X_1 X_2} & \dots & S_{X_1 X_k} \\ S_{X_1 X_2} & S_{X_2 X_2} & \dots & S_{X_2 X_k} \\ \dots & \dots & \dots & \dots \\ S_{X_1 X_k} & S_{X_2 X_k} & \dots & S_{X_k X_k} \end{pmatrix}, \quad (1.4)$$

де  $S_{X_q X_r} = \sum_{i=1}^N [X_{qi} - \bar{X}_q][X_{ri} - \bar{X}_r]$ ,  $q, r = \overline{1, k}$ .

Коли початкові дані не відповідають гаусівському розподілу, виникає потреба у додаткових діях. Для побудови нелінійної регресійної моделі можна застосувати один з трьох підходів: метод підбору, лінеаризуючі перетворення або

нормалізуючі перетворення. Останній підхід зазвичай є найменш витратним у часі та дозволяє досягти високої якості моделі.

Алгоритм методів, що базуються на нормалізуючих перетвореннях, включає кілька етапів. Спершу початкові дані нормалізуються за обраним перетворенням. На другому етапі видаляються викиди, а очищені дані використовуються для побудови лінійної регресії. Залишки регресії побудованої моделі тестуються за допомогою критерію Пірсона, і в разі їх невідповідності нормальному розподілу найбільш проблемний рядок даних (найбільший по модулю) виключається, після чого процес повторюється. Завершивши цей етап, використовують зворотне перетворення для отримання остаточної нелінійної моделі. Заключним етапом є побудова інтервалів передбачення і видалення даних, що виходять за їх межі, тобто викидів [25, 26].

Нормалізацію даних, що не є нормальними, виконуємо за таким перетворенням

$$T = \psi(P), \quad (1.5)$$

тут  $P = \{Y, X_1, X_2, \dots, X_k\}^T$  – вектор ненормалізованих даних,  $T = \{Z_Y, Z_1, Z_2, \dots, Z_k\}^T$  – вектор нормалізованих даних,  $\psi = \{\psi_Y, \psi_1, \psi_2, \dots, \psi_k\}^T$  – застосоване нормалізуюче перетворення.

Зворотнє перетворення до (1.5) можна представити як

$$P = \psi^{-1}(T). \quad (1.6)$$

Фінальний вигляд нелінійної регресійної моделі, яка побудована з застосуванням нормалізуючого перетворення [25]

$$Y = \psi_Y^{-1}(\hat{Z}_Y + \varepsilon) = \psi_Y^{-1}(b_0 + b_1 Z_1 + b_2 Z_2 + \dots + b_k Z_k + \varepsilon), \quad (1.7)$$

де  $\psi_Y$  – перша компонента вектору  $\psi$  перетворення (1.5).

Довірчий інтервал нелінійної регресії має вигляд [24]

$$\psi_Y^{-1} \left( \hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ \frac{1}{N} + (\mathbf{z}_X^+)^T [(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right). \quad (1.8)$$

Інтервал передбачення нелінійної регресії має вигляд [24]

$$\psi_Y^{-1} \left( \hat{Z}_Y \pm t_{\frac{\alpha}{2}, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (\mathbf{z}_X^+)^T [(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right). \quad (1.9)$$

## 1.2 Обґрунтування необхідності проведення досліджень

Застосування лінійного регресійного аналізу для оцінювання розміру ПЗ обмежується низкою умов, наприклад, вимогою нормальності розподілу залишків. У більшості випадків ці умови не виконуються через складність і різноманітність реальних даних, що робить лінійні моделі менш ефективними.

Нелінійні регресійні моделі стають необхідністю, оскільки вони здатні враховувати складні взаємозв'язки між змінними та адаптуватися до даних із негаусівським розподілом. Ці моделі забезпечують більшу гнучкість і достовірність порівняно з лінійними підходами, особливо коли мова йде про задачі прогнозування.

Більше того, сучасні методи, які базуються на багатовимірних нормалізуючих перетвореннях, дозволяють значно підвищити ефективність побудови нелінійних регресійних моделей, знижуючи вплив викидів і покращуючи стабільність результатів. Усе це робить представлений підхід незамінним інструментом у задачах оцінювання розміру програмного забезпечення [25, 26].

Найбільш застосованою мовою веб-розробки є PHP [23], для якої відомо чимало нелінійних регресійних моделей, побудованих за метриками діаграми класів [13-20]. У дослідженні [19] показано, що побудова нелінійної регресійної моделі для конкретного фреймворку є виправданою, адже готові нелінійні регресійні моделі не будуть враховувати специфіку обраного фреймворку. Подібне

властиве й для різних типів проектів. Це ставить питання щодо необхідності проведення дослідження за даною темою.

### 1.3 Висновки до розділу 1

У розділі 1 проведено аналіз важливості розміру програмного забезпечення у процесі його розробки та розглянуто сучасні підходи до його оцінювання. Особливу увагу приділено ролі мов програмування, зокрема PHP, у розробці CMS, а також методам регресійного аналізу, що використовуються для оцінювання LOC. Було висвітлено переваги й обмеження лінійних та нелінійних регресійних моделей, а також обґрунтовано доцільність застосування нормалізуючих перетворень для покращення достовірності прогнозування. Отримані результати підкреслюють необхідність розробки моделей, адаптованих до специфіки конкретних фреймворків та типів проектів, що створює основу для подальших досліджень у цій галузі.

## 2 СТВОРЕННЯ НЕЛІНІЙНОЇ РЕГРЕСІЙНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ СИСТЕМ КЕРУВАННЯ ВМІСТОМ НА МОВІ PHP

2.1 Перевірка даних для побудови нелінійної регресійної моделі для оцінювання розміру CMS на PHP на наявність мультиколінеарності та викидів

У межах дослідження було обрано 45 проектів систем управління вмістом на основі PHP для побудови нелінійної регресійної моделі. Пошук проектів здійснювався через GitHub (<https://github.com/>), а метрики діаграми класів отримано за допомогою інструмента PhpMetrics. Аналізовані метрики включають: KLOC – кількість рядків коду в тисячах ( $Y$ ), кількість класів ( $X_1$ ), середню кількість методів на клас ( $X_2$ ), а також глибину дерева успадкування DIT ( $X_3$ ). Ці показники формують основу для побудови моделі. Метрики наведено у таблиці 2.1

Таблиця 2.1 – Метрики разом з розрахунком відстані Махаланобіса ( $SMD_x$ )

| №  | Project                                                                                                       | KLOC<br>( $Y$ ) | Classes<br>( $X_1$ ) | Methods<br>by Class<br>( $X_2$ ) | DIT<br>( $X_3$ ) | $SMD_x$ |
|----|---------------------------------------------------------------------------------------------------------------|-----------------|----------------------|----------------------------------|------------------|---------|
| 1  | 2                                                                                                             | 3               | 4                    | 5                                | 6                | 7       |
| 1  | <a href="https://github.com/LavaLite/cms">https://github.com/LavaLite/cms</a>                                 | 1,192           | 47                   | 1,55                             | 1,97             | 8,147   |
| 2  | <a href="https://github.com/PhileCMS/Phile">https://github.com/PhileCMS/Phile</a>                             | 3,476           | 51                   | 3,55                             | 1,32             | 1,410   |
| 3  | <a href="https://github.com/s-cart/s-cart">https://github.com/s-cart/s-cart</a>                               | 3,598           | 65                   | 3,45                             | 1,85             | 3,229   |
| 4  | <a href="https://github.com/flextype/flextype">https://github.com/flextype/flextype</a>                       | 4,547           | 84                   | 3,08                             | 1,1              | 3,865   |
| 5  | <a href="https://github.com/nasirkhan/laravel-starter">https://github.com/nasirkhan/laravel-starter</a>       | 5,54            | 110                  | 2,87                             | 1,51             | 1,626   |
| 6  | <a href="https://github.com/arifszn/ezfolio">https://github.com/arifszn/ezfolio</a>                           | 5,558           | 76                   | 2,62                             | 1,44             | 2,206   |
| 7  | <a href="https://github.com/noumo/easyii">https://github.com/noumo/easyii</a>                                 | 6,719           | 146                  | 4,05                             | 2,05             | 6,192   |
| 8  | <a href="https://github.com/htmlburger/wpemerge">https://github.com/htmlburger/wpemerge</a>                   | 6,797           | 83                   | 4,51                             | 1,12             | 2,633   |
| 9  | <a href="https://github.com/Azuriom/Azuriom">https://github.com/Azuriom/Azuriom</a>                           | 11,298          | 261                  | 3,67                             | 1,4              | 0,704   |
| 10 | <a href="https://github.com/code16/sharp">https://github.com/code16/sharp</a>                                 | 11,382          | 381                  | 3,28                             | 1,5              | 0,875   |
| 11 | <a href="https://github.com/magicblack/maccms10">https://github.com/magicblack/maccms10</a>                   | 11,556          | 204                  | 3,16                             | 1,37             | 1,273   |
| 12 | <a href="https://github.com/uasoft-indonesia/badaso">https://github.com/uasoft-indonesia/badaso</a>           | 11,556          | 204                  | 3,16                             | 1,37             | 1,273   |
| 13 | <a href="https://github.com/moonshine-software/moonshine">https://github.com/moonshine-software/moonshine</a> | 14,733          | 337                  | 4,15                             | 1,18             | 1,764   |
| 14 | <a href="https://github.com/liufee/cms">https://github.com/liufee/cms</a>                                     | 14,756          | 164                  | 4,46                             | 1,85             | 2,994   |

Продовження табл. 2.1

| 1  | 2                                                                                                                       | 3       | 4    | 5    | 6    | 7             |
|----|-------------------------------------------------------------------------------------------------------------------------|---------|------|------|------|---------------|
| 15 | <a href="https://github.com/luyadev/luya">https://github.com/luyadev/luya</a>                                           | 17,349  | 160  | 3,41 | 1,72 | 2,003         |
| 16 | <a href="https://github.com/AsgardCms/Platform">https://github.com/AsgardCms/Platform</a>                               | 19,497  | 504  | 2,92 | 1,17 | 2,334         |
| 17 | <a href="https://github.com/NamelessMC/Nameless">https://github.com/NamelessMC/Nameless</a>                             | 20,931  | 310  | 3,5  | 1,21 | 1,692         |
| 18 | <a href="https://github.com/croogo/croogo">https://github.com/croogo/croogo</a>                                         | 24,66   | 367  | 3,36 | 1,64 | 1,293         |
| 19 | <a href="https://github.com/area17/twill">https://github.com/area17/twill</a>                                           | 24,9    | 396  | 4,42 | 1,3  | 0,767         |
| 20 | <a href="https://github.com/Contentify/Contentify">https://github.com/Contentify/Contentify</a>                         | 32,281  | 465  | 4,38 | 1,88 | 3,666         |
| 21 | <a href="https://github.com/skeeks-cms/cms">https://github.com/skeeks-cms/cms</a>                                       | 40,427  | 601  | 3,74 | 1,84 | 3,197         |
| 22 | <a href="https://github.com/redaxo/redaxo">https://github.com/redaxo/redaxo</a>                                         | 43,668  | 305  | 6,97 | 1,32 | 3,005         |
| 23 | <a href="https://github.com/Piwigo/Piwigo">https://github.com/Piwigo/Piwigo</a>                                         | 44,775  | 308  | 5,41 | 1,24 | 1,036         |
| 24 | <a href="https://github.com/pH7Software/pH7-Social-Dating-CMS">https://github.com/pH7Software/pH7-Social-Dating-CMS</a> | 46,029  | 651  | 4,01 | 1,19 | 1,137         |
| 25 | <a href="https://github.com/wintercms/winter">https://github.com/wintercms/winter</a>                                   | 49,238  | 345  | 6,92 | 1,53 | 2,830         |
| 26 | <a href="https://github.com/filamentphp/filament">https://github.com/filamentphp/filament</a>                           | 50,009  | 886  | 5,72 | 1,09 | 4,242         |
| 27 | <a href="https://github.com/bolt/bolt">https://github.com/bolt/bolt</a>                                                 | 59,851  | 576  | 4,98 | 1,26 | 0,503         |
| 28 | <a href="https://github.com/backdrop/backdrop">https://github.com/backdrop/backdrop</a>                                 | 60,045  | 476  | 6,2  | 1,71 | 2,674         |
| 29 | <a href="https://github.com/Kunstmaan/KunstmaanBundlesCMS">https://github.com/Kunstmaan/KunstmaanBundlesCMS</a>         | 60,694  | 856  | 3,69 | 1,19 | 1,130         |
| 30 | <a href="https://github.com/forkcms/forkcms">https://github.com/forkcms/forkcms</a>                                     | 65,159  | 768  | 4,53 | 1,32 | 0,232         |
| 31 | <a href="https://github.com/getgrav/grav">https://github.com/getgrav/grav</a>                                           | 67,812  | 409  | 8,78 | 1,47 | 7,533         |
| 32 | <a href="https://github.com/statamic/cms">https://github.com/statamic/cms</a>                                           | 89,874  | 1519 | 5,09 | 1,24 | 3,502         |
| 33 | <a href="https://github.com/contao/contao">https://github.com/contao/contao</a>                                         | 98,994  | 986  | 4,76 | 1,15 | 0,859         |
| 34 | <a href="https://github.com/daylightstudio/FUEL-CMS">https://github.com/daylightstudio/FUEL-CMS</a>                     | 113,294 | 507  | 8,43 | 1,38 | 4,071         |
| 35 | <a href="https://github.com/xibosignage/xibo-cms">https://github.com/xibosignage/xibo-cms</a>                           | 117,984 | 776  | 4,67 | 1,06 | 2,782         |
| 36 | <a href="https://github.com/pimcore/pimcore">https://github.com/pimcore/pimcore</a>                                     | 120,181 | 1291 | 5,46 | 1,16 | 1,173         |
| 37 | <a href="https://github.com/craftcms/cms">https://github.com/craftcms/cms</a>                                           | 169,691 | 1198 | 6,24 | 1,83 | 5,393         |
| 38 | <a href="https://github.com/ClassicPress/ClassicPress">https://github.com/ClassicPress/ClassicPress</a>                 | 178,023 | 455  | 9,81 | 1,38 | 7,768         |
| 39 | <a href="https://github.com/processwire/processwire">https://github.com/processwire/processwire</a>                     | 184,02  | 521  | 9,75 | 1,32 | 7,599         |
| 40 | <a href="https://github.com/ExpressionEngine/ExpressionEngine">https://github.com/ExpressionEngine/ExpressionEngine</a> | 249,572 | 1538 | 6,23 | 1,16 | 5,442         |
| 41 | <a href="https://github.com/joomla/joomla-cms">https://github.com/joomla/joomla-cms</a>                                 | 268,991 | 2001 | 3,66 | 1,19 | <b>9,895</b>  |
| 42 | <a href="https://github.com/concretecms/concretecms">https://github.com/concretecms/concretecms</a>                     | 286,531 | 4058 | 3,91 | 1,2  | <b>14,262</b> |
| 43 | <a href="https://github.com/Qloapps/QloApps">https://github.com/Qloapps/QloApps</a>                                     | 317,919 | 1361 | 8,19 | 1,26 | <b>14,040</b> |
| 44 | <a href="https://github.com/TYPO3/typo3">https://github.com/TYPO3/typo3</a>                                             | 326,03  | 3030 | 4,92 | 1,14 | 7,182         |
| 45 | <a href="https://github.com/PrestaShop/PrestaShop">https://github.com/PrestaShop/PrestaShop</a>                         | 350,116 | 4741 | 3,23 | 1,1  | <b>18,567</b> |

Відстань Махаланобіса (SMD) для  $i$ -тої точки багатовимірних даних можна обчислити за формулою [24]:

$$d_i^2 = (X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X}), \quad (2.1)$$

де  $X_i$  –  $i$ -та точка багатовимірного вектора  $X$ ;  $\bar{X}$  – вектор середніх значень;  $S_N$  – вибіркова коваріаційна матриця,  $N$  – розмір вибірки. Вектор має чотири компоненти. Вибіркова коваріаційна матриця для (2.1) обчислюється як [24]

$$S_N = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X})^T. \quad (2.2)$$

Аналіз показав, що значення SMD для рядків 41–43 та 45 перевищують критичне значення квантиля розподілу  $\chi^2$  із рівнем значущості 0,005 (9,488). Це свідчить про те, що дані метрик не відповідають нормальному розподілу.

Щоб підтвердити цей результат, застосовується тест на багатовимірний ексцес  $\hat{\beta}_2$  [27], який обчислюється за формулою:

$$\hat{\beta}_2 = \frac{1}{N} \sum_{i=1}^N \{(X_i - \bar{X})^T S_N^{-1} (X_i - \bar{X})\}^2. \quad (2.3)$$

Для метрик систем управління вмістом на РНР отримано значення  $\hat{\beta}_2 = 31,432$ . Це значення порівнюється з критичним

$$\beta_2 = m(m + 2),$$

де  $m = 4$  (для трьох факторної моделі)

Критичне значення  $\beta_2 = 24$ , що менше за значення зі спостережень за (2.2). Це ще раз підтверджує, що дані метрик не мають гаусівського розподілу.

Важливим кроком перед побудовою нелінійної регресійної моделі є перевірка майбутніх предикторів на мультиколінеарність.

Мультиколінеарність у множинному регресійному аналізі є проблемою, яка не дозволяє точно оцінити вплив кожного предиктора на залежну змінну. Це явище характеризується сильним лінійним зв'язком між факторами моделі.

Для виявлення мультиколінеарності використовують показники Variance Inflation Factors (VIFs), які розраховуються як елементи головної діагоналі оберненої коваріаційної матриці розміром  $k \times k$  для моделі з  $k$ -предикторами. Значення VIFs у діапазоні від 1 до 5 свідчать про відсутність мультиколінеарності, тоді як показники, що перевищують 10, сигналізують про її наявність [28].

Аналіз для предикторів  $X_1, X_2, X_3$  (таблиця 2.1) показав значення VIFs: 1,164, 1,014 і 1,180. Оскільки всі ці значення не перевищують поріг у 5, модель успішно пройшла перевірку на мультиколінеарність.

Результати обчислення SMD для ненормалізованих даних показують, що їх розподіл не відповідає нормальному. У випадку ненормальності вихідних даних викиди видаляються за допомогою методу, який включає нормалізуючі перетворення та розрахунок SMD [29, 30]. Для нормалізованих даних  $i$ -та точка SMD обчислюється за формулою [24]

$$d_i^2 = (Z_i - \bar{Z})^T S_N^{-1} (Z_i - \bar{Z}), \quad (2.4)$$

де  $Z_i$  –  $i$ -та точка багатовимірного нормалізованого вектора  $Z$ ;  $\bar{Z}$  – вектор вибірових середніх;  $S_N$  – вибіркова коваріаційна матриця для нормалізованих даних, вектор є чотирьох-компонентним. Вибіркова коваріаційна матриця для (2.4) розраховується як [24]

$$S_N = \frac{1}{N} \sum_{i=1}^N (Z_i - \bar{Z}) (Z_i - \bar{Z})^T. \quad (2.5)$$

Нормалізований вектор  $Z$  отримується за допомогою нормалізуючого перетворення  $\log_{10}$ , визначеного як  $\psi = \{\lg Y, \lg X_1, \lg X_2, \lg X_3\}^T$ .

Для багатовимірного нормального розподілу значення SMD узгоджуються з розподілом  $\chi^2$ . У великій вибірці ( $N - m > 25$ ) значення  $d_i^2$  близькі до випадкової величини  $\chi_{m,\alpha}^2$ , де  $\chi_{m,\alpha}^2$  — квантіль розподілу  $\chi^2$  для рівня значущості  $\alpha$  [31].

Для 45 рядків усі викиди були видалені за одну ітерацію на рівні значущості 0,005. Значення SMD для кожної ітерації показано в таблиці 2.2. Критичне значення квантіля  $\chi^2$  для цього рівня значущості становить 9,488.

Таблиця 2.2 – Пошук та вилучення викидів

| №  | $SMD_1$       | $SMD_2$ | №  | $SMD_1$ | $SMD_2$ |
|----|---------------|---------|----|---------|---------|
| 1  | <b>11,067</b> | -       | 24 | 0,856   | 0,817   |
| 2  | 5,175         | 5,318   | 25 | 2,267   | 2,229   |
| 3  | 3,990         | 4,812   | 26 | 7,850   | 8,208   |
| 4  | 6,655         | 6,634   | 27 | 0,279   | 0,257   |
| 5  | 2,109         | 2,861   | 28 | 2,831   | 2,754   |
| 6  | 6,383         | 7,514   | 29 | 1,364   | 1,338   |
| 7  | 7,167         | 7,535   | 30 | 0,332   | 0,305   |
| 8  | 6,687         | 6,518   | 31 | 4,680   | 4,890   |
| 9  | 2,212         | 2,231   | 32 | 4,503   | 4,535   |
| 10 | 5,245         | 5,287   | 33 | 1,106   | 1,091   |
| 11 | 1,065         | 1,402   | 34 | 3,401   | 3,566   |
| 12 | 1,065         | 1,402   | 35 | 4,392   | 4,284   |
| 13 | 4,705         | 4,601   | 36 | 1,740   | 1,864   |
| 14 | 2,950         | 3,311   | 37 | 6,297   | 6,159   |
| 15 | 5,585         | 6,567   | 38 | 7,507   | 7,622   |
| 16 | 3,112         | 3,130   | 39 | 6,408   | 6,665   |
| 17 | 1,256         | 1,305   | 40 | 2,119   | 2,258   |
| 18 | 1,926         | 2,500   | 41 | 8,185   | 8,102   |
| 19 | 1,090         | 1,043   | 42 | 5,785   | 5,634   |
| 20 | 4,270         | 4,475   | 43 | 3,246   | 3,562   |
| 21 | 4,472         | 4,931   | 44 | 3,485   | 3,469   |
| 22 | 2,405         | 2,487   | 45 | 8,916   | 8,725   |
| 23 | 1,859         | 1,801   |    |         |         |

Очищений набір даних із 44 проектів систем управління вмістом на мові РНР буде використаний для подальшого аналізу. Одна з достатніх умов застосування лінійного регресійного аналізу – тест на нормальність розподілу залишків  $\varepsilon$ . Для вибірок із великим обсягом ( $n > 30$ ) це можна зробити за допомогою критерію Пірсона [32].

Розрахунок  $\chi^2$  здійснюється за формулою

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - np_j)^2}{np_j}, \quad (2.6)$$

де  $m$  – кількість відрізків для розбиття  $[x_{min}, x_{max}]$ ;  $n_j$  – абсолютна частота на  $j$ -му відрізку;  $p_j$  – ймовірність того, що значення випадкової величини  $X$  попадають у  $j$ -ий відрізок [32].

Кількість інтервалів  $m$  можна визначити за формулами:

– Старджеса:  $m = \log_2 n + 1 = 3,3 \lg n + 1$ .

– Брукса і Карузера:  $m = 5 \lg n$ .

Ймовірності  $p_j$  обчислюються як [32]

$$p_j = \int_{x_{j-1}}^{x_j} f(x) dx, \quad (2.7)$$

де  $x_{j-1}$  та  $x_j$  – нижня та верхня границі  $j$ -го відрізка,  $f(x)$  – функція щільності ймовірності.

В свою чергу функції щільності  $f(x)$  для  $p_j$  (2.7) обчислюється як [32]

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} * e^{\frac{-(x-m_x)^2}{2\sigma_x^2}}, \quad (2.8)$$

де  $m_x$  – математичне сподівання випадкової величини,  $\sigma_x$  – середьоквадратичне відхилення випадкової величини.

Значення ступенів вільності  $\nu$  визначається як  $\nu = m - k - 1$ , де  $k = 2$  для нормального розподілу. Значення  $\chi^2$ , отримане для залишків, порівнюється з критичним. Якщо  $\chi^2 < \chi_{кр}^2$ , нульова гіпотеза про гаусівський розподіл приймається, інакше – відхиляється.

## 2.2 Побудова нелінійної регресійної моделі для оцінювання розміру CMS на РНР

Отримавши набір даних без викидів, будемо здійснювати ітераційний процес побудови багатофакторної нелінійної регресійної моделі для оцінювання розміру систем керування вмістом на мові РНР. Лінійна регресійна модель для нормалізованих даних, модель отримується за (1.7) та має вигляд

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \quad (2.9)$$

де  $Z_Y = \lg Y$ ;  $Z_i = \lg X_i, i = \overline{1, 3}$ .

Оцінки параметрів моделі:  $\hat{b}_0 = -1,9364, \hat{b}_1 = 1,0027, \hat{b}_2 = 1,3563, \hat{b}_3 = -0,1354$

Використовуючи формулу (2.6), для залишків регресії було визначено емпіричне значення критерію Пірсона  $\chi^2 = 15,54$ . Критичне значення критерію Пірсона для трьох ступенів свободи та рівня значущості 0,05 становить  $\chi^2 = 7,814$ . Оскільки  $\chi^2 > \chi_{кр}^2$ , нульова гіпотеза про гаусівський розподіл залишків відхиляється. Рядок даних 26 слід вважати викидом за залишками регресії. Перейдемо до першого кроку ітераційного процесу побудови нелінійної регресійної моделі – пошуку викидів у початкових нормалізованих даних без рядка 26.

На другій ітерації викидів за розрахунком SMD не було виявлено. Для даних без викидів побудовано лінійну регресійну модель для нормалізованих даних (без рядків 44 та 26) за (1.5), залишки регресії перевірено на нормальність, отримано  $\chi^2 = 4,83$ . На основі лінійного рівняння (2.9) та перетворення (1.6) було побудовано нелінійну регресійну модель

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}. \quad (2.10)$$

Інтервальні оцінки для побудованої нелінійної моделі (2.10) визначено за формулами (1.8) і (1.9). Матриця  $(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+$ , що використовується у формулах (1.8) та (1.9) отримується за (1.4) та має вигляд

$$(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = \begin{pmatrix} 9,3085 & 0,8874 & -0,5690 \\ 0,8874 & 0,9527 & -0,0045 \\ 0,5690 & -0,0045 & 0,2431 \end{pmatrix}$$

За побудованими інтервалами прогнозування визначено два викиди регресії – рядки 15 та 41, відкинемо їх та перейдемо до першого кроку ітераційного процесу.

На третій ітерації виявлено, що рядок 6 є викидом за розрахунком SMD, відкинемо його та перейдемо до наступного кроку ітераційного процесу.

На четвертій ітерації виявлено, що рядок 45 є викидом за розрахунком SMD, відкинемо його та перейдемо до наступного кроку ітераційного процесу.

На п'ятій ітерації викидів за розрахунком SMD не виявлено, для даних без викидів побудовано лінійну регресійну модель для нормалізованих даних (без рядків 44, 26, 15, 41, 6 та 45), залишки регресії перевірено на нормальність, отримано  $\chi^2 = 6,1$ . За визначеними інтервалами передбачення нелінійної регресії виявлено два викиди – рядки 13 та 32, відкинемо їх та перейдемо до першого кроку ітераційного процесу.

На шостій ітерації викидів за розрахунком SMD не виявлено, для даних без викидів побудовано лінійну регресійну модель для нормалізованих даних (без рядків 44, 26, 15, 41, 6, 45, 13 та 32), залишки регресії перевірено на нормальність, отримано  $\chi^2 = 3,5$ . За визначеними інтервалами передбачення нелінійної регресії виявлено, що рядок 10 – викид. відкинемо його та перейдемо до першого кроку ітераційного процесу.

На шостій ітерації викидів за розрахунком SMD не виявлено, для даних без викидів побудовано лінійну регресійну модель для нормалізованих даних (без рядків 44, 26, 15, 41, 6, 45, 13, 32 та 10), залишки регресії перевірено на нормальність, отримано  $\chi^2 = 4,36$ . За визначеними інтервалами передбачення викидів не виявлено, отже модель побудовано.

Оцінки параметрів фінальної моделі:  $\hat{b}_0 = -1,9928, \hat{b}_1 = 0,9963, \hat{b}_2 = 1,5259, \hat{b}_3 = -0,4007$ , значення матриці  $(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+$

$$(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = \begin{pmatrix} 6,7114 & 0,8402 & -0,3515 \\ 0,8402 & 0,8119 & -0,0023 \\ -0,3515 & -0,0023 & 0,2125 \end{pmatrix}$$

Оцінювання якості моделі дало наступні результати:

- коефіцієнт детермінації  $R^2 = 0,970$ ;
- середня величина відносної похибки  $MMRE = 0,148$ ;
- відсоток прогнозованих результатів  $PRED(0,25) = 0,806$ .

Усі показники відповідають критеріям прийнятності ( $MMRE \leq 0,25$ ,  $PRED(0,25) > 0,75$ ,  $R^2 > 0,75$ ), що підтверджує високу якість моделі [33, 34].

У цьому дослідженні побудовано трьохфакторну лінійну регресійну модель для оцінювання розміру CMS на PHP за (1.1). Ця модель побудована з припущенням про нормальність розподілу метрик і має вигляд

$$Y = \hat{Y} + \varepsilon = \hat{b}_0 + \hat{b}_1 X_1 + \hat{b}_2 X_2 + \hat{b}_3 X_3 + \varepsilon. \quad (2.11)$$

Оцінки параметрів моделі:  $\hat{b}_0 = -58,332, \hat{b}_1 = 0,0873, \hat{b}_2 = 21,2801, \hat{b}_3 = -20,0229$ .

Для перевірки нормальності розподілу залишків регресії  $\varepsilon$  застосовано критерій Пірсона. Емпіричне значення критерію становить  $\chi^2 = 17,348$ , тоді як критичне значення для 3 ступенів свободи на рівні значущості 0,05 дорівнює  $\chi^2 = 7,814$ . Оскільки  $\chi^2 > \chi_{кр}^2$ , нульова гіпотеза відхиляється, що свідчить про відхилення від гаусівського розподілу. Розрахунки показали, що математичне сподівання  $\varepsilon$  дорівнює 80,61, а середньоквадратичне відхилення – 92,092.

Оцінювання якості моделі дало наступні результати:

- коефіцієнт детермінації  $R^2 = 0,874$ ;
- середня величина відносної похибки  $MMRE = 0,804$ ;
- відсоток прогнозованих результатів  $PRED(0,25) = 0,444$ .

Значення  $MMRE$  і  $PRED(0,25)$  не відповідають прийнятним межам ( $MMRE \leq 0,25$ ,  $PRED(0,25) > 0,75$ ), що разом із ненормальністю розподілу залишків свідчить про низьку якість цієї моделі.

Розрахунки інтервальних оцінок передбачають застосування матриці  $(\mathbf{X}^+)^T \mathbf{X}^+$

$$(\mathbf{X}^+)^T \mathbf{X}^+ = \begin{pmatrix} 22929722,5 & 4146,3722 & -2227,8283 \\ 4146,3722 & 132,7938 & -0,5344 \\ -2227,8283 & -0,5344 & 2,4729 \end{pmatrix}$$

Таблиця 2.3 містить дані про інтервальні оцінки, їх ширини для багатofакторних регресійних моделей (лінійної та нелінійної), розраховані за формулами (1.2) і (1.8). Результати показали, що у 83,33% випадків (30 з 36 рядків) ширина довірчих інтервальних оцінок для нелінійної регресії менша, ніж для лінійної.

Таблиця 2.3 – Довірчі інтервали лінійної та нелінійної регресії

| №  | KLOC<br>(Y) | Межі довірчих інтервалів |         |                    |        | Ширина інтервалу |                    |
|----|-------------|--------------------------|---------|--------------------|--------|------------------|--------------------|
|    |             | Лінійна регресія         |         | Нелінійна регресія |        | Лінійна регресія | Нелінійна регресія |
|    |             | LB                       | UB      | LB                 | UB     |                  |                    |
| 1  | 2           | 3                        | 4       | 5                  | 6      | 7                | 8                  |
| 1  | 3,476       | -22,950                  | 13,415  | 2,674              | 3,734  | 36,365           | <b>1,059</b>       |
| 2  | 3,598       | -41,361                  | 8,790   | 2,855              | 3,966  | 50,151           | <b>1,111</b>       |
| 3  | 4,547       | -32,176                  | 17,208  | 3,765              | 5,378  | 49,384           | <b>1,613</b>       |
| 4  | 5,54        | -37,125                  | 1,340   | 4,077              | 5,316  | 38,465           | <b>1,239</b>       |
| 5  | 6,719       | -32,121                  | 31,216  | 7,803              | 10,932 | 63,336           | <b>3,129</b>       |
| 6  | 6,797       | 1,086                    | 43,833  | 6,623              | 9,423  | 42,747           | <b>2,801</b>       |
| 7  | 11,298      | -0,915                   | 29,941  | 15,178             | 17,972 | 30,856           | <b>2,793</b>       |
| 8  | 11,556      | -18,457                  | 17,029  | 9,334              | 11,528 | 35,486           | <b>2,193</b>       |
| 9  | 11,556      | -18,457                  | 17,029  | 9,334              | 11,528 | 35,486           | <b>2,193</b>       |
| 10 | 14,756      | -9,610                   | 37,306  | 10,946             | 14,317 | 46,917           | <b>3,370</b>       |
| 11 | 19,497      | 3,554                    | 45,180  | 21,094             | 27,579 | 41,627           | <b>6,485</b>       |
| 12 | 20,931      | 0,208                    | 37,745  | 17,456             | 21,411 | 37,537           | <b>3,955</b>       |
| 13 | 24,66       | -6,378                   | 31,102  | 16,845             | 21,492 | 37,480           | <b>4,647</b>       |
| 14 | 24,9        | 30,213                   | 58,303  | 31,872             | 36,762 | 28,091           | <b>4,890</b>       |
| 15 | 32,281      | 12,923                   | 62,709  | 29,511             | 39,571 | 49,786           | <b>10,059</b>      |
| 16 | 40,427      | 12,254                   | 61,481  | 29,818             | 41,022 | 49,227           | <b>11,204</b>      |
| 17 | 43,668      | 71,904                   | 108,455 | 46,776             | 59,047 | 36,550           | <b>12,272</b>      |

## Продовження таблиці 2.3

| 1  | 2       | 3       | 4       | 5       | 6       | 7       | 8             |
|----|---------|---------|---------|---------|---------|---------|---------------|
| 18 | 44,775  | 43,070  | 74,622  | 33,683  | 40,573  | 31,553  | <b>6,890</b>  |
| 19 | 46,029  | 43,582  | 76,402  | 45,555  | 55,229  | 32,820  | <b>9,674</b>  |
| 20 | 49,238  | 70,602  | 106,201 | 49,574  | 61,899  | 35,599  | <b>12,325</b> |
| 21 | 59,851  | 59,166  | 86,206  | 56,015  | 65,120  | 27,039  | <b>9,105</b>  |
| 22 | 60,045  | 61,362  | 100,457 | 54,801  | 69,559  | 39,096  | <b>14,758</b> |
| 23 | 60,694  | 54,014  | 88,135  | 51,807  | 65,013  | 34,121  | <b>13,206</b> |
| 24 | 65,159  | 66,130  | 91,202  | 62,943  | 74,183  | 25,073  | <b>11,239</b> |
| 25 | 67,812  | 108,740 | 160,800 | 83,004  | 110,797 | 52,060  | <b>27,793</b> |
| 26 | 98,994  | 89,871  | 122,111 | 90,066  | 110,798 | 32,241  | <b>20,732</b> |
| 27 | 113,294 | 113,852 | 161,503 | 100,288 | 130,715 | 47,651  | <b>30,428</b> |
| 28 | 117,984 | 68,255  | 106,844 | 70,089  | 88,972  | 38,589  | 18,883        |
| 29 | 120,181 | 130,464 | 164,150 | 143,852 | 179,155 | 33,686  | 35,303        |
| 30 | 169,691 | 115,131 | 169,615 | 128,813 | 179,807 | 54,483  | <b>50,994</b> |
| 31 | 178,023 | 131,033 | 193,977 | 110,102 | 152,430 | 62,944  | <b>42,328</b> |
| 32 | 184,02  | 136,985 | 199,396 | 127,324 | 175,592 | 62,411  | <b>48,268</b> |
| 33 | 249,572 | 165,810 | 204,691 | 207,347 | 263,505 | 38,881  | 56,158        |
| 34 | 286,531 | 303,057 | 406,986 | 245,401 | 361,380 | 103,929 | 115,979       |
| 35 | 317,919 | 185,534 | 233,483 | 265,950 | 347,251 | 47,948  | 81,300        |
| 36 | 326,03  | 251,699 | 324,288 | 275,667 | 377,614 | 72,589  | 101,947       |

У таблиці 2.4 наведено інтервали передбачення й їх ширини, визначені за формулами (1.3) та (1.9). Встановлено, що нелінійна регресія забезпечує менші інтервали передбачення у 88,88% випадків (32 з 36 рядків).

Таблиця 2.4 – Інтервали передбачення лінійної та нелінійної регресії

| № | KLOC<br>(Y) | Межі інтервалів передбачення |        |                    |        | Ширина інтервалу |                    |
|---|-------------|------------------------------|--------|--------------------|--------|------------------|--------------------|
|   |             | Лінійна регресія             |        | Нелінійна регресія |        | Лінійна регресія | Нелінійна регресія |
|   |             | LB                           | UB     | LB                 | UB     |                  |                    |
| 1 | 2           | 3                            | 4      | 5                  | 6      | 7                | 8                  |
| 1 | 3,476       | -76,736                      | 67,202 | 2,049              | 4,874  | 143,938          | <b>2,825</b>       |
| 2 | 3,598       | -90,297                      | 57,726 | 2,184              | 5,185  | 148,024          | <b>3,001</b>       |
| 3 | 4,547       | -81,366                      | 66,399 | 2,904              | 6,972  | 147,765          | <b>4,067</b>       |
| 4 | 5,54        | -90,134                      | 54,349 | 3,055              | 7,095  | 144,483          | <b>4,040</b>       |
| 5 | 6,719       | -76,950                      | 76,045 | 5,984              | 14,255 | 152,994          | <b>8,270</b>       |
| 6 | 6,797       | -50,382                      | 95,300 | 5,103              | 12,230 | 145,682          | <b>7,127</b>       |
| 7 | 11,298      | -56,810                      | 85,836 | 10,975             | 24,855 | 142,646          | <b>13,879</b>      |
| 8 | 11,556      | -72,573                      | 71,145 | 6,860              | 15,687 | 143,719          | <b>8,827</b>       |

## Продовження таблиці 2.4

| 1  | 2       | 3       | 4       | 5       | 6       | 7       | 8              |
|----|---------|---------|---------|---------|---------|---------|----------------|
| 9  | 11,556  | -72,573 | 71,145  | 6,860   | 15,687  | 143,719 | <b>8,827</b>   |
| 10 | 14,756  | -59,632 | 87,328  | 8,211   | 19,087  | 146,959 | <b>10,877</b>  |
| 11 | 19,497  | -48,311 | 97,045  | 15,820  | 36,773  | 145,357 | <b>20,953</b>  |
| 12 | 20,931  | -53,143 | 91,096  | 12,796  | 29,210  | 144,239 | <b>16,415</b>  |
| 13 | 24,66   | -59,750 | 84,474  | 12,526  | 28,901  | 144,224 | <b>16,375</b>  |
| 14 | 24,9    | -26,779 | 115,295 | 22,803  | 51,382  | 142,074 | <b>28,579</b>  |
| 15 | 32,281  | -36,134 | 111,766 | 22,321  | 52,319  | 147,900 | <b>29,999</b>  |
| 16 | 40,427  | -36,989 | 110,724 | 22,739  | 53,793  | 147,713 | <b>31,053</b>  |
| 17 | 43,668  | 18,187  | 162,172 | 34,652  | 79,707  | 143,985 | <b>45,055</b>  |
| 18 | 44,775  | -12,553 | 130,245 | 24,520  | 55,735  | 142,798 | <b>31,216</b>  |
| 19 | 46,029  | -11,550 | 131,534 | 33,245  | 75,681  | 143,084 | <b>42,436</b>  |
| 20 | 49,238  | 16,528  | 160,275 | 36,579  | 83,889  | 143,747 | <b>47,310</b>  |
| 21 | 59,851  | 1,751   | 143,621 | 40,206  | 90,726  | 141,869 | <b>50,520</b>  |
| 22 | 60,045  | 8,583   | 153,236 | 40,677  | 93,712  | 144,652 | <b>53,035</b>  |
| 23 | 60,694  | -0,620  | 142,768 | 38,297  | 87,947  | 143,388 | <b>49,651</b>  |
| 24 | 65,159  | 7,912   | 149,420 | 45,429  | 102,783 | 141,508 | <b>57,354</b>  |
| 25 | 67,812  | 60,429  | 209,110 | 62,686  | 146,710 | 148,681 | <b>84,024</b>  |
| 26 | 98,994  | 34,515  | 177,467 | 66,092  | 150,989 | 142,952 | <b>84,897</b>  |
| 27 | 113,294 | 64,080  | 211,275 | 75,134  | 174,477 | 147,195 | <b>99,343</b>  |
| 28 | 117,984 | 15,291  | 159,807 | 52,026  | 119,862 | 144,516 | <b>67,835</b>  |
| 29 | 120,181 | 75,664  | 218,949 | 106,044 | 243,029 | 143,285 | <b>136,986</b> |
| 30 | 169,691 | 67,600  | 217,146 | 98,678  | 234,717 | 149,547 | <b>136,040</b> |
| 31 | 178,023 | 86,089  | 238,921 | 84,130  | 199,487 | 152,832 | <b>115,358</b> |
| 32 | 184,02  | 91,883  | 244,497 | 97,171  | 230,079 | 152,614 | <b>132,907</b> |
| 33 | 249,572 | 112,953 | 257,547 | 153,973 | 354,847 | 144,594 | 200,874        |
| 34 | 286,531 | 268,135 | 441,908 | 190,979 | 464,360 | 173,773 | 273,381        |
| 35 | 317,919 | 135,863 | 283,154 | 199,366 | 463,227 | 147,292 | 263,861        |
| 36 | 326,03  | 209,468 | 366,519 | 209,937 | 495,841 | 157,051 | 285,904        |

Варто зазначити, що в лінійних моделях у деяких випадках трапляються від'ємні значення як у довірчих інтервалах, так і в інтервалах передбачення, що є неприйнятним.

У таблиці 2.5 представлені результати оцінювання якості лінійної та нелінійної регресійних моделей, побудованих для визначення розміру систем управління вмістом на РНР.

Таблиця 2.5 – Порівняння оцінок лінійної та нелінійної моделей

| Оцінки якості моделі | Модель  |           |
|----------------------|---------|-----------|
|                      | Лінійна | Нелінійна |
| $R^2$                | 0,874   | 0,970     |
| $MMRE$               | 0,804   | 0,148     |
| $PRED(0,25)$         | 0,444   | 0,806     |

З таблиці видно, що нелінійна модель демонструє значно кращі результати за метриками  $MMRE$  та  $PRED(0,25)$ , у порівнянні з лінійною моделлю. Хоча для обох моделей отримано високе значення  $R^2$ , достовірність оцінювання у нелінійній моделі вища.

Таблиця 2.6 містить порівняння оцінок якості для різних нелінійних моделей, розроблених для PHP-фреймворків. Моделі побудовані з використанням аналогічних метрик-предикторів: кількість класів, середня кількість методів на клас, глибина дерев успадкування, та кількість рядків коду у тисячах. Маємо, що вони можуть бути застосовані для аналізу емпіричних даних, отриманих у цьому дослідженні.

Таблиця 2.6 – Оцінки для різних типів проектів

| Оцінки якості моделі | Модель       |         |       |         |
|----------------------|--------------|---------|-------|---------|
|                      | CMS          | CakePHP | Yii   | Symfony |
| $R^2$                | <b>0,970</b> | 0,919   | 0,595 | 0,727   |
| $MMRE$               | <b>0,148</b> | 0,208   | 0,348 | 0,249   |
| $PRED(0,25)$         | <b>0,806</b> | 0,722   | 0,389 | 0,556   |

З таблиці видно, що застосування моделей, розроблених для інших типів проектів (наприклад, для фреймворків), призводить до гірших результатів за метриками  $R^2$ ,  $MMRE$  та  $PRED(0,25)$ . Це підтверджує необхідність розробки регресійних моделей, адаптованих до конкретного фреймворку, типу проектів або мови програмування.

## 2.3 Висновки до розділу 2

У даному розділі була виконана розробка нелінійної регресійної моделі для прогнозування розміру CMS на PHP із використанням логарифмічного нормалізуючого перетворення  $\log_{10}$ .

Детально в цьому розділі виконано наступне:

- Зібрано дані з проектів на GitHub. Метрики діаграми класів отримано за допомогою PhpMetrics.

- Проведено перевірку метрик на наявність мультиколінеарності, яка не була виявлена, та підтверджено негаусівський характер розподілу за допомогою квадрата відстані Махаланобіса.

- Видалено один викид із початкового набору даних із 45 проектів за методами нормалізації та аналізу відстані Махаланобіса.

- Побудовано та проаналізовано лінійну та нелінійну регресійні моделі. Нелінійна модель перевершує лінійну за всіма ключовими показниками якості.

- Виконано розрахунок довірчих інтервалів і інтервалів передбачення. Нелінійна модель демонструє вищу достовірність завдяки меншим ширинам інтервалів.

- Здійснено порівняння побудованої моделі з існуючими моделями для PHP-фреймворків, зокрема CakePHP, Yii та Symfony. Побудована модель має значно кращі результати.

## **3 РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ СИСТЕМ КЕРУВАННЯ ВМІСТО НА МОВІ PHP**

### **3.1 Формулювання завдання на створення ПЗ для оцінювання розміру CMS на PHP**

Одним із ключових завдань цієї роботи є створення програмного забезпечення для кінцевого користувача, яке забезпечить можливість оцінювання розміру систем керування вмістом на мові PHP. Важливо забезпечити сумісність цього ПЗ із сучасними популярними операційними системами. Для цього слід ретельно обрати мову програмування, бібліотеки та інші необхідні утиліти. У даному контексті оптимальним вибором є мова Python (<https://www.python.org/>).

Python – це мова програмування високого рівня з відкритим кодом, що підтримує динамічну строгість типізації. Основні причини вибору Python:

- наявність бібліотек NumPy, SciPy, Pandas, які дозволяють виконувати математичні обчислення на концептуальному рівні;
- графічна бібліотека Qt для створення GUI;
- відсутність обмежень у використанні для комерційних цілей.

Таким чином, Python є оптимальним інструментом для виконання розробки в межах цієї кваліфікаційної роботи. Розробка програми для оцінювання розміру систем керування вмістом на мові PHP буде здійснюватись відповідно до технічного завдання, поданого у додатку А. У роботі також будуть створені ескізний, технічний та робочий проекти.

### **3.2 Розробка ескізного проекту ПЗ для оцінювання розміру CMS на PHP**

Ескізний проект передбачає концептуальне моделювання створюваної системи, для чого буде розроблено діаграму прецедентів та діаграму активності.

### 3.2.1 Розробка моделі сценаріїв використання ПЗ для оцінювання розміру CMS на PHP

Діаграма прецедентів (Use Case Diagram) – будується аби концептуально показати операції, які можна виконувати з системою та акторів, які виконують ці операції. Так як ця діаграма є концептуальною, на ній відсутні будь-які деталі реалізації. До переваг такого моделювання можна віднести зрозумілість цих діаграм людям, які далекі від розробки ПЗ. Таблиця 3.1 містить перелік виявлених прецедентів.

Таблиця 3.1 – Виявлені прецеденти

| Основні актори | Найменування                                   | Формулювання                                                   |
|----------------|------------------------------------------------|----------------------------------------------------------------|
| Користувач     | Ввести дані з метрик CMS                       | Задає файл із метриками CMS                                    |
| Користувач     | Ввести кількість класів                        | Вказує кількість класів у CMS                                  |
| Користувач     | Ввести середню кількість методів на кожен клас | Вводить середню кількість методів для класів CMS               |
| Користувач     | Ввести глибину дерева наслідування             | Вказує глибину наслідування CMS                                |
| Користувач     | Ввести довірчу ймовірність                     | Задає значення довірчої ймовірності                            |
| Користувач     | Нормалізувати дані                             | Отримує нормалізовані дані через log10-перетворення            |
| Користувач     | Побудувати нелінійну регресійну модель         | Отримує результати побудови моделі                             |
| Користувач     | Виконати оцінювання розміру                    | Отримує розмір CMS, інтервал передбачення та довірчий інтервал |
| Користувач     | Виконати оцінювання інтервалу передбачення     | Отримує інтервал передбачення CMS                              |
| Користувач     | Виконати оцінювання довірчого інтервалу        | Визначає довірчий інтервал для CMS                             |

Діаграму прецедентів показано на рисунку 3.1.

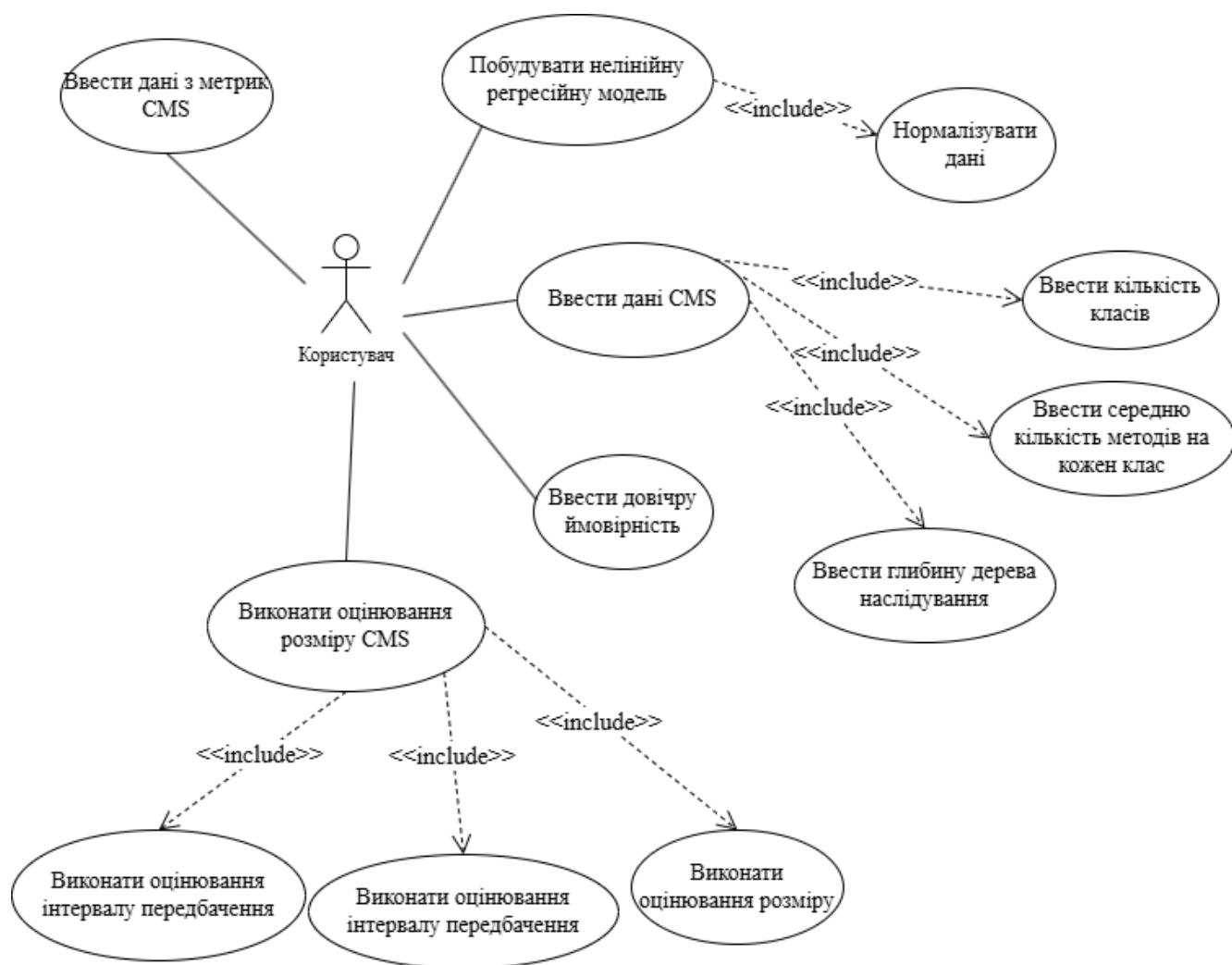


Рисунок 3.1 – Діаграма варіантів використання

3.2.2 Розробка сценарію виконання ПЗ для оцінювання розміру CMS на PHP за допомогою діаграми діяльності

Діаграма діяльності є одним із типів діаграм в UML (Unified Modeling Language) і використовується для моделювання послідовності дій у системі. Вона відображає, як виконуються певні процеси або задачі, показуючи потік управління та передачу даних між різними об'єктами чи етапами. Основними компонентами

діаграми є дії, рішення, початкові та кінцеві стани, а також потоки переходів між ними.

Діаграма діяльності дозволяє аналізувати складні процеси, виявляти проблемні місця або дублювання дій, а також оптимізувати робочі процеси. Вона досить розповсюджена у застосуванні в розробці ПЗ, управлінні проектами, бізнес-аналітиці та документуванні. Така діаграма допомагає не лише розробникам, але й зацікавленим сторонам (наприклад, менеджерам чи клієнтам) зрозуміти, як функціонує система.

Діаграму діяльності для розроблюваного програмного забезпечення показано на рисунку 3.2.

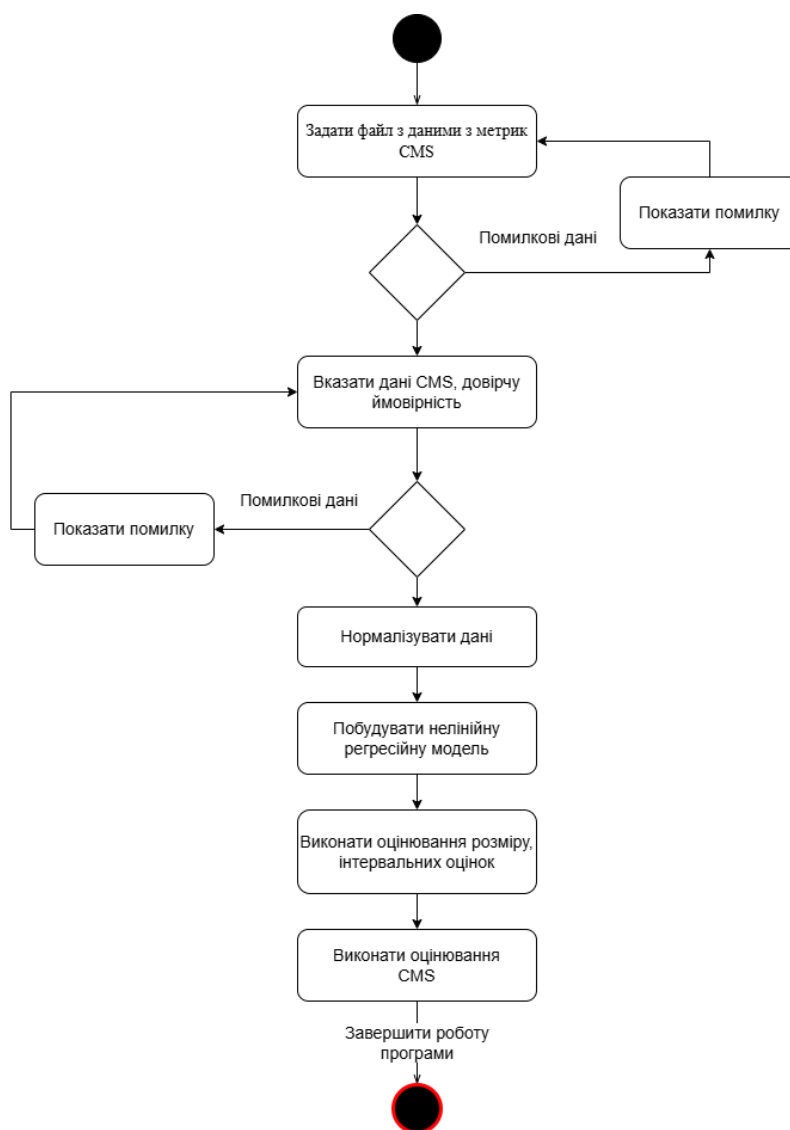


Рисунок 3.2 – Діаграма діяльності

### 3.3 Розробка технічного проекту ПЗ для оцінювання розміру CMS на PHP

#### 3.3.1 Статична модель ПЗ для оцінювання розміру CMS на PHP у вигляді діаграми класів

Діаграма класів – це один із основних типів діаграм у UML (Unified Modeling Language), який використовується для моделювання структури системи. Вона відображає класи, їхні атрибути, методи (функції) та взаємозв'язки між класами. Основними компонентами діаграми класів є:

- Клас – об'єкт, який описує певну сутність із набором атрибутів (характеристик) і методів (поведінки).
- Асоціації – відношення між класами, що показують, як вони взаємодіють.
- Залежності, агрегація та композиція – різновиди зв'язків між класами, які відображають ступінь і тип їхньої взаємодії.
- Успадкування – зв'язок між базовим класом і похідними, що показує ієрархію.

Діаграма класів допомагає структурувати дані, спроектувати взаємодію між компонентами системи, а також створити чітке уявлення про архітектуру програмного забезпечення. Вона особливо корисна на етапі проектування й розробки системи.

Діаграму класів, що показує статичну структуру програмного забезпечення для оцінювання розміру систем керування вмістом на мові PHP наведено на рисунку 3.3.

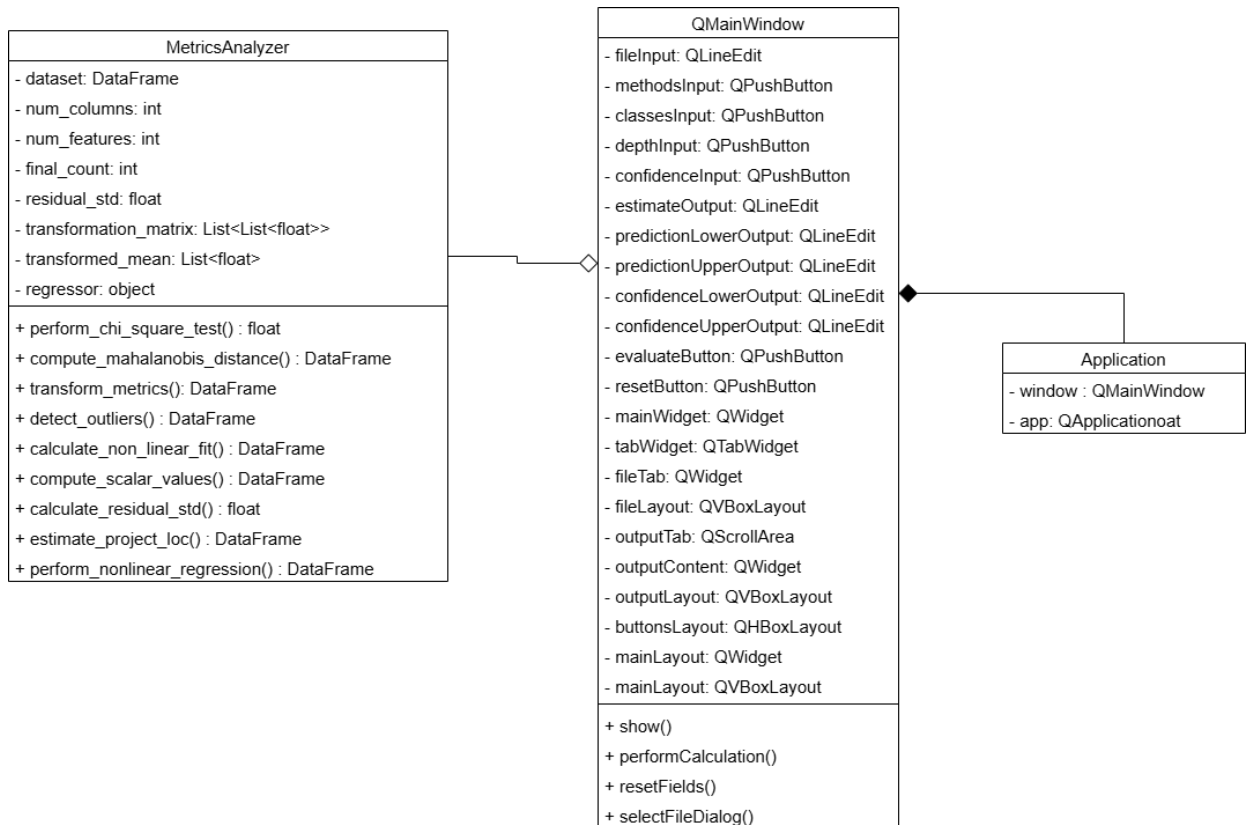


Рисунок 3.3 – Діаграма класів

### 3.3.2 Специфікації класів ПЗ для оцінювання розміру CMS на PHP

– Клас **QMainWindow** – головний клас, який відповідає за основне вікно програми.

– Клас **QWidget** – базовий клас бібліотеки Qt, що дозволяє створювати елементи графічного інтерфейсу.

– Клас **QLineEdit** – клас, який використовується для створення текстових полів у графічному інтерфейсі.

– Клас **QValidator** – клас, що забезпечує перевірку даних у реальному часі, запобігаючи введенню некоректної інформації.

– Клас **Application** – основний клас програми, який відповідає за її функціонування.

– Клас `MetricsAnalyzer` – клас, який використовується для створення регресійних моделей, оцінювання розміру застосунку та розрахунку інтервальних оцінок.

### 3.3.3 Розробка динамічного зображення ПЗ для оцінювання розміру CMS на PHP у вигляді діаграм послідовностей

Діаграма послідовностей – це один із типів діаграм UML (Unified Modeling Language), який використовується для моделювання взаємодії об'єктів у часі. Вона показує, як об'єкти обмінюються повідомленнями та як відбувається потік інформації між ними. Основними елементами цієї діаграми є:

- Об'єкти — учасники взаємодії, які представлені вертикальними лініями.
- Повідомлення — дії або виклики методів, що передаються між об'єктами.
- Часова шкала — показує послідовність і тривалість взаємодії.
- Активність об'єкта — проміжки часу, коли об'єкт виконує певну дію.

Діаграму послідовності для прецедента «Ввести дані CMS» наведено на рисунку 3.4.

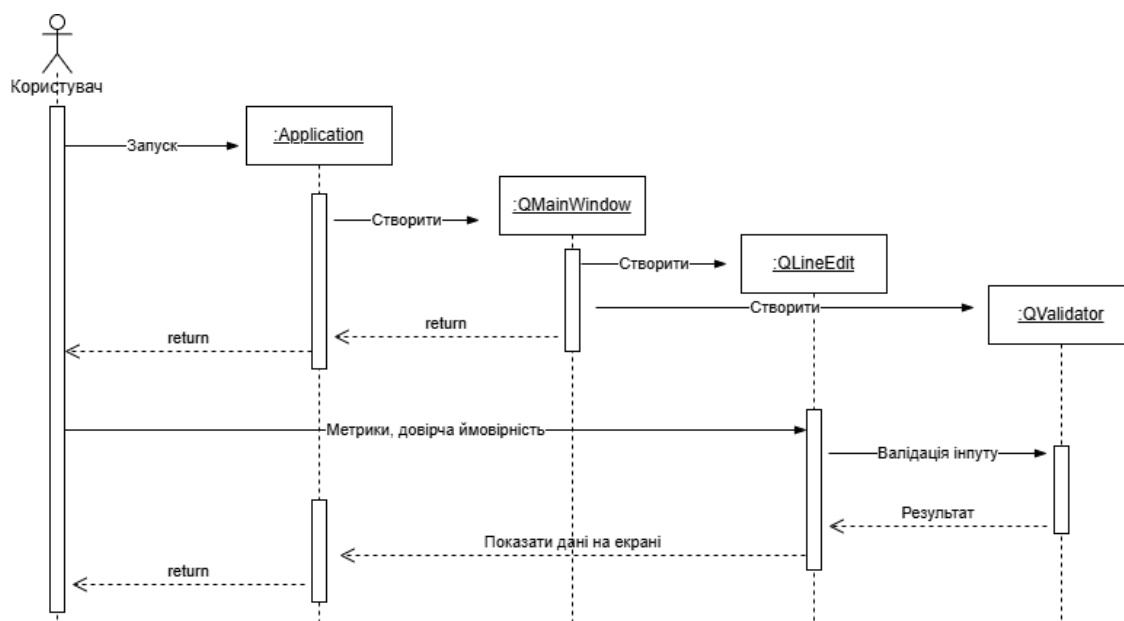


Рисунок 3.4 – Діаграма послідовності для варіанту прецедента «Ввести дані CMS»

Діаграму послідовності для прецедента «Виконати оцінювання розміру CMS» наведено на рисунку 3.5.

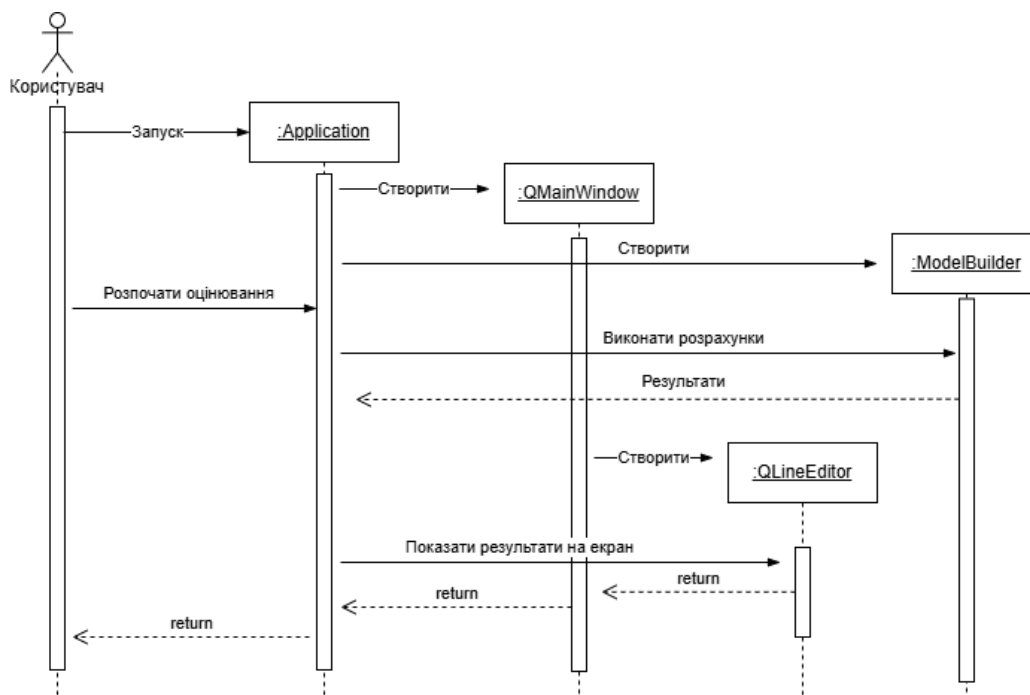


Рисунок 3.5 – Діаграма послідовності прецедента «Виконати оцінювання розміру CMS»

### 3.4 Розробка робочого проекту ПЗ для оцінювання розміру CMS на PHP

#### 3.4.1 Кодування ПЗ для оцінювання розміру CMS на PHP

У рамках робочого проєкту було розроблено програму для оцінювання розміру CMS на PHP у тисячах рядків коду. Проведено тестування та перевірку працездатності програми.

Програму реалізовано з на Python, а для створення графічного інтерфейсу користувача використано бібліотеку Qt, сумісну з Python. Повний лістинг коду наведено в додатку Г.

### 3.4.2 Тестування ПЗ для оцінювання розміру CMS на PHP

Еквівалентне розбиття – це техніка тестування ПЗ, що вбачає розділення набору вхідних даних на класи еквівалентності. Усі значення в одному класі вважаються еквівалентними, оскільки вони повинні оброблятися програмою однаково. Ця методика дозволяє скоротити кількість тестових випадків, зберігаючи їх ефективність.

Класи еквівалентності діляться на:

- Позитивні – дані, які відповідають очікуваним умовам.
- Негативні – дані, які знаходяться за межами допустимих умов.

Мета еквівалентного розбиття – підвищити ефективність тестування за рахунок зменшення дублюючих перевірок і охоплення найважливіших сценаріїв.

Для перевірки коректності роботи застосунку виконано тестування з розробкою супровідної документації.

Прикладом є варіант використання «Ввести кількість класів». Таблиця 3.2 демонструє виявлені класи еквівалентності, а результати тестування подано у таблицях 3.3 та 3.4, відповідно для правильних та неправильних класів еквівалентності.

Таблиця 3.2 – Виявлені класи еквівалентності для тестування введення кількості класів

| <i>Вхідні данні</i> | <i>Правильні класи</i> | <i>Неправильні класи</i>                                                           |
|---------------------|------------------------|------------------------------------------------------------------------------------|
| Classes             | Натуральне число (1)   | Дійсне число (2).<br>Від’ємне число (3).<br>Не число (4)<br>Значення відсутнє (5). |

Таблиця 3.3 – Тести правильних класів еквівалентності

| № тесту | Покритий клас | Вхідні дані | Вихідні данні               | Результат тестування |
|---------|---------------|-------------|-----------------------------|----------------------|
| 1       | 1             | 15          | Продовження роботи програми | Пройдено             |

Таблиця 3.4 – Тести неправильних класів еквівалентності

| № тесту | Покритий клас | Вхідні дані | Вихідні данні                                | Результат тестування |
|---------|---------------|-------------|----------------------------------------------|----------------------|
| 2       | 2             | 2,7         | Повідомлення про помилку                     | Пройдено             |
| 3       | 3             | -1,8        | Повідомлення про помилку                     | Пройдено             |
| 4       | 4             | Qwerty12345 | Повідомлення про помилку                     | Пройдено             |
| 5       | 5             | -           | Повідомлення про необхідність заповнити поле | Пройдено             |

Іншим прикладом є варіант використання «Ввести кількість методів на кожен клас». Таблиця 3.5 демонструє виявлені класи еквівалентності, а результати тестування подано у таблицях 3.6 та 3.7, відповідно для правильних та неправильних класів еквівалентності.

Таблиця 3.5 – Виявлені класи для варіанту використання «Ввести кількість методів на кожен клас»

| <i>Вхідні данні</i>                          | <i>Правильні класи</i>   | <i>Неправильні класи</i>                                      |
|----------------------------------------------|--------------------------|---------------------------------------------------------------|
| Methods by class (depth of inheritance tree) | Дійсне додатне число (1) | Від’ємне число (2).<br>Не число (3)<br>Значення відсутнє (4). |

Таблиця 3.6 – Тести правильних класів еквівалентності

| № тесту | Покритий клас | Вхідні дані | Вихідні данні               | Результат тестування |
|---------|---------------|-------------|-----------------------------|----------------------|
| 1       | 1             | 5,90        | Продовження роботи програми | Пройдено             |

Таблиця 3.7 – Тести неправильних класів еквівалентності

| № тесту | Покритий клас | Вхідні дані | Вихідні данні                                | Результат тестування |
|---------|---------------|-------------|----------------------------------------------|----------------------|
| 2       | 2             | -123,45     | Повідомлення про помилку                     | Пройдено             |
| 3       | 3             | Qwerty1!    | Повідомлення про помилку                     | Пройдено             |
| 4       | 4             | -           | Повідомлення про необхідність заповнити поле | Пройдено             |

Тестування введення глибини дерева наслідування виконується аналогічно сценарію для середньої кількості методів. Процес перевірки введення довірчої

ймовірності відбувається подібно до інших варіантів використання, за винятком того, що це значення має бути дійсним числом у діапазоні від 0 до 1.

Результати тестів свідчать про відповідність роботи програми заявленим вимогам.

### 3.4.3 Проведення випробувань ПЗ для оцінювання розміру CMS на РН

Згідно з розробленою та затвердженою методикою випробувань, описаною у Додатку Б, було проведено оцінювання розміру систем керування вмістом, реалізованих на мові РНР. Зокрема, досліджено такі функції:

- введення метрик CMS;
- введення даних, що визначити оцінку розміру CMS;
- застосування нормалізуючого перетворення до даних;
- побудова нелінійної регресійної моделі;
- розрахунок оцінок розміру CMS, довірчих інтервалів та інтервалів передбачення.

На рисунках 3.6 та 3.7 представлено результати роботи програмного застосунку. Аналіз отриманих результатів підтверджує відповідність програмного забезпечення висунутим вимогам.

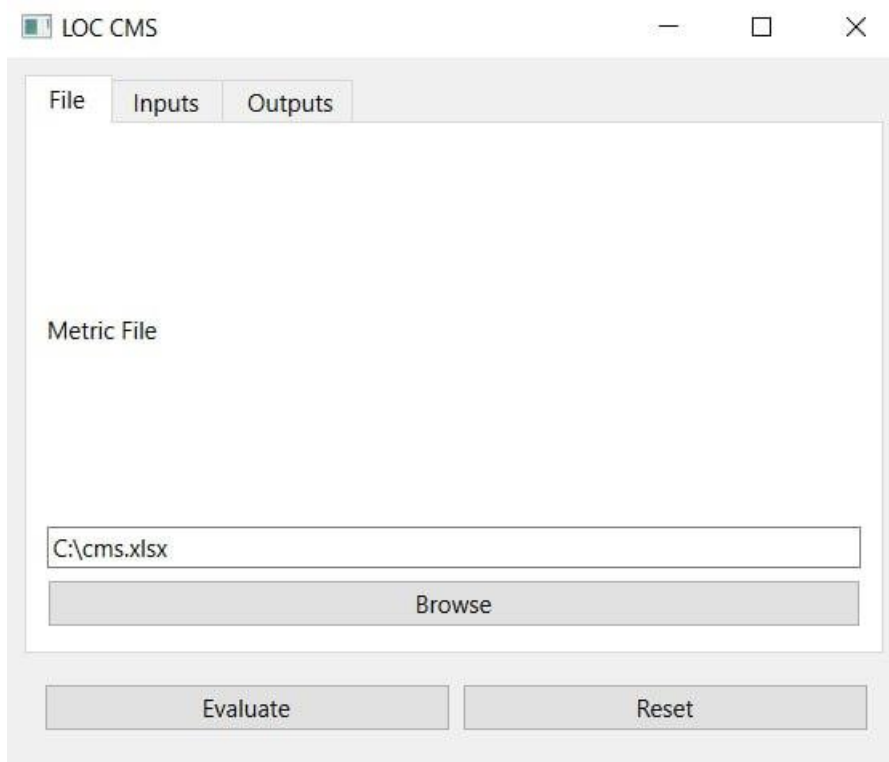


Рисунок 3.6 – Стартове вікно програми

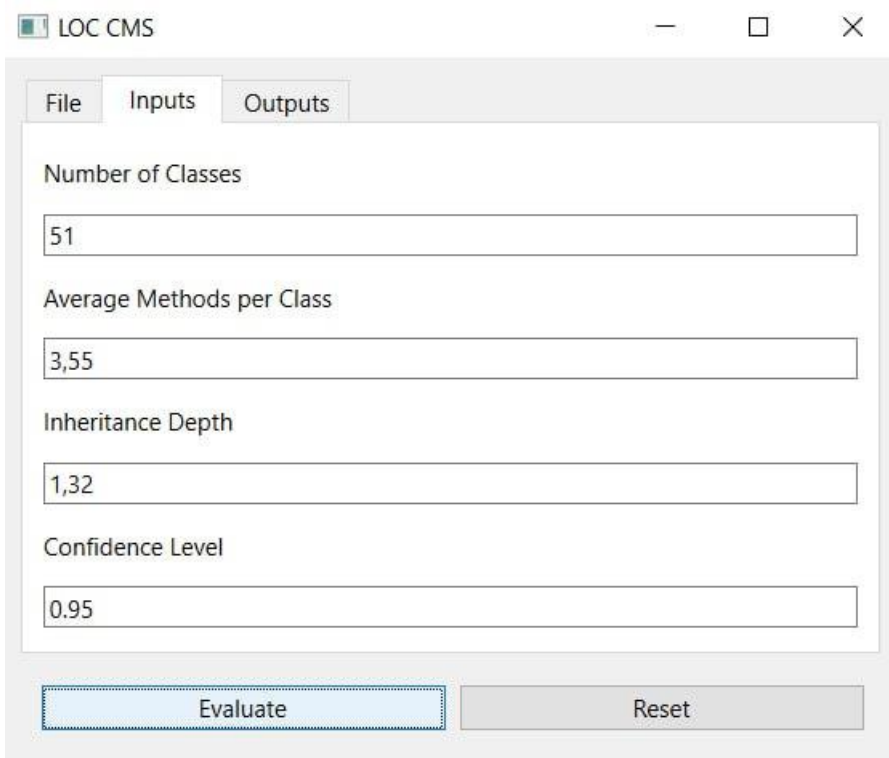


Рисунок 3.7 – Вікно програми до виконання оцінювання

The screenshot shows a window titled 'LOC CMS' with three tabs: 'File', 'Inputs', and 'Outputs'. The 'Outputs' tab is active, displaying a 'Size Estimate' section with the following values:

| Category                    | Value |
|-----------------------------|-------|
| Size Estimate               | 3.16  |
| Confidence Interval (Lower) | 2.67  |
| Confidence Interval (Upper) | 3.73  |
| Prediction Interval (Lower) | 2.05  |
| Prediction Interval (Upper) | 4.87  |

At the bottom of the window, there are two buttons: 'Evaluate' and 'Reset'.

Рисунок 3.8 – Результат оцінювання CMS на мові PHP

Отже вимоги, висунуті у технічному завданні, виконані.

### 3.5 Висновки до розділу 3

У цьому розділі представлено проект ПЗ для оцінювання розміру CMS на PHP.

У межах розділу:

- було підібрано необхідні програмні засоби для проведення розробки;
- розроблено ескізний проект, що містить діаграму перцедентів та діаграму діяльності;
- виконано технічний проект із побудовою діаграми класів, специфікаціями до них та діаграмами послідовності для визначених сценаріїв;
- реалізовано робочий проект, включаючи етапи кодування, тестування й перевірки створеного ПЗ.

## **4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ І ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ СИСТЕМ КЕРУВАННЯ ВМІСТОМ НА МОВІ PHP**

### **Вступ**

Розробку програмного забезпечення доцільно оцінювати з економічної точки зору. Це можна зробити, розрахувавши собівартість розробки та очікуваний прибуток. Головним показником, що свідчить про доцільність витрат на створення програмного продукту, є річний економічний ефект.

Економічна обґрунтованість розробки визначається через аналіз собівартості та прогнозування прибутків від реалізації. Основний критерій, який підтверджує доцільність витрат, – це річний економічний ефект.

### **4.1 Розрахунок витрат на створення й експлуатацію програмного забезпечення**

Витрати на розробку програмного забезпечення складаються з витрат на заробітну плату розробника, на амортизацію та експлуатацію ЕОМ, на якій виконується розробка, на засоби розробки та витрати на матеріали і комплектуючі. Розробка програмного забезпечення виконується спеціалістом, місячний оклад якого складає 20000 грн. Додаткова заробітна плата складає 20% від основної. Виходячи з цього, основна і додаткова заробітна плата розроблювача системи 24000 грн./міс , а вартість сучасної ПЗВМ складає 23000 грн. Вартість кіловат-години електроенергії дорівнює 4,32 грн. Витрати на допоміжні матеріали наведені в таблиці 4.1

Таблиця 4.1 – Витрати на допоміжні матеріали

| № | Пункти витрат                  | Сума, грн. |
|---|--------------------------------|------------|
| 1 | Папір для принтера             | 400        |
| 2 | Картридж та тонер для принтера | 350        |
| 3 | Використання флеш-накопичувача | 220        |
| 4 | Непередбачувані витрати        | 1500       |
|   | Разом                          | 2470       |

Вартість розробки системи розраховується за формулою:

$$C_{\text{пр}} = (Z_{\text{зп}} + Z_{\text{зс}} + Z_{\text{зг}} + Z_{\text{е}}) * T + Z_{\text{м}},$$

де  $T$  – тривалість розробки у місяцях

$Z_{\text{зп}}$  – основна і додаткова заробітна плата, грн.;

$Z_{\text{зс}}$  – відрахування на соціальні заходи (38% від основної і додаткової заробітної плати), грн.;

$Z_{\text{зг}}$  – загальногосподарські витрати (10% від заробітної плати), грн.;

$Z_{\text{е}}$  – витрати на електроенергію, грн.

$Z_{\text{м}}$  – витрати на допоміжні матеріали, грн.

$Z_{\text{е}}$  при витратах в 0,5 кВт з тривалістю роботи на місяць  $22 * 8 = 176$  годин та вартості електроенергії 4,32 грн. становить

$$Z_{\text{е}} = 176 * 4,32 * 0,5 = 380,16 \text{ грн.} \quad (4.1)$$

Представимо всі поточні витрати на розробку програмного забезпечення в таблиці 4.2

Таблиця 4.2 – Витрати на розробку програмного забезпечення

| № | Пункти витрат                                            | Одиниця | Кількість |
|---|----------------------------------------------------------|---------|-----------|
| 1 | Тривалість розробки ( $T$ )                              | Міс.    | 2         |
| 2 | Основна та додаткова заробітна плата ( $Z_{\text{зп}}$ ) | Грн.    | 24000     |
| 3 | Відрахування на соціальні витрати ( $Z_{\text{зс}}$ )    | Грн.    | 9120      |
| 4 | Загальногосподарські витрати ( $Z_{\text{зг}}$ )         | Грн.    | 2400      |
| 5 | Витрати на допоміжні матеріали ( $Z_{\text{м}}$ )        | Грн.    | 2470      |
| 6 | Витрати на електроенергію $*(Z_{\text{е}})$              | Грн.    | 380,16    |

За формулою (4.1) виконаємо розрахунок вартості програми:

$$C_{\text{пр}} = (24000 + 9120 + 2400 + 380,16) * 2 + 2470 = 74270,32 \text{ грн.}$$

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$A_{\text{об}} = 23000 * 0,6 = 13800 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$B_{\text{м}} = 23000 * 0,05 = 1150 \text{ грн.}$$

Річний обсяг робіт ПК у годинах можна визначити як

$$\Phi_{\text{м}} = 264,5 * T_{\text{з}}, \quad (4.2)$$

де  $T_{\text{з}}$  – середнє навантаження на устаткування в місяць (6 годин), 264,5 – середня кількість робочих днів на рік.

Отже, річний обсяг роботи ПК складає:

$$\Phi_{\text{м}} = 264,5 * 6 = 1587 \text{ годин}$$

Витрати на електроенергію  $З_{\text{е}}$  при 1587 годинах роботи устаткування на рік становлять:

$$З_{\text{е}} = 1587 * 4,32 = 6555,84$$

Експлуатаційні витрати на ПК на рік становлять:

$$З_{\text{зр}} = 13800 + 1150 + 6555,84 = 21805,84$$

Таким чином витрати на розробку та експлуатацію програмного забезпечення в перший рік становлять:

$$З_{\text{р}} = 74270,32 + 21805,84 = 96076,16$$

#### 4.2 Економічна ефективність розробки та впровадження програмного забезпечення

Визначити пряму економічну ефективність можна, ґрунтуючись на тому, що впровадження програмного забезпечення вивільняє одного працівника (за експертною оцінкою фахівців підприємства). Зарплата одного працівника в рік складає

$$З = 20000 * 12 * 1 = 240000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$E_{\text{рік}} = \Delta C_n - E_n * k, \quad (4.3)$$

де  $\Delta C_n$  – кошти, що буде вивільнено при впровадженні програмного забезпечення (240000) без експлуатаційних витрат (21805,84), тобто, 218149,16 грн.;

$E_n$  – коефіцієнт ефективності (такий же як й коефіцієнт амортизації – 0,6)

$k$  – одноразові витрати на впровадження програмного забезпечення (74270,32 грн.).

$$E_{\text{рік}} = 218149,16 - 0,6 * 74270,32 = 218149,16 - 28362,19 = 189786,94$$

Термін, за який програмне забезпечення окупиться наступний:

$$T = \frac{k}{\Delta C_n} = \frac{74270,32}{218149,16} = 0,34 \text{ року}$$

Отже термін, за який програмне забезпечення окупиться 4 місяці.

#### 4.3 Висновки до розділу 4

У цьому розділі виконано розрахунки, що стосуються створення та впровадження програмного забезпечення для оцінювання розміру систем управління вмістом мовою РНР. Після цього було визначено економічну ефективність і термін окупності розробленого продукту. За результатами розрахунків, програмне забезпечення окупиться протягом 4 місяців у перший рік використання. Таким чином, розробка програмного забезпечення є економічно вигідною.

## 5 ОХОРОНА ПРАЦІ

Охорона праці – це система заходів правового, соціально-економічного, організаційно-технічного, санітарно-гігієнічного і лікувально-профілактичного характеру, спрямованих на забезпечення безпечних умов праці, збереження здоров'я і працездатності працівників у процесі виконання їхніх професійних обов'язків.

До цієї системи входять:

- аналіз шкідливих і небезпечних чинників, що впливають на людину;
- впровадження заходів для зменшення або усунення негативного впливу та забезпечення комфортних умов праці;
- дотримання вимог з техніки безпеки та виробничої санітарії;
- спеціальні правила для охорони праці жінок, неповнолітніх і працівників зі зниженою працездатністю;
- норми для працівників у складних, небезпечних та шкідливих умовах праці.

Безпека праці визначається багатьма чинниками, серед яких головними є рівень кваліфікації та відповідальності керівників. Суттєву роль у підвищенні безпеки виконують стандарти охорони праці, що включають взаємопов'язані нормативні документи для запобігання травмам і професійним захворюванням. Закон України "Про охорону праці" гарантує права працівників на безпечні умови праці та регламентує відносини між роботодавцем і працівником з питань безпеки, гігієни праці та санітарії.

Поліпшення умов праці сприяє не лише підвищенню її безпеки, а й позитивно впливає на продуктивність, якість продукції та економічні результати підприємства. Соціальні результати включають зміцнення здоров'я працівників, підвищення їхнього задоволення роботою, зростання престижу професій і покращення дисципліни.

Також заходи з охорони праці сприяють охороні довкілля, адже очищення стічних вод, боротьба з шумом і вібрацією чи захист від випромінювання одночасно забезпечують комфортні умови для життя людини.

### 5.1 Аналіз шкідливих та небезпечних факторів в офісі фірми

При роботі співробітники відділу зазнають впливу небезпечних і шкідливих факторів, як-то:

- недостатня освітленість робочого місця;
- підвищений рівень шуму і вібрації від роботи вентиляторів, принтерів;
- висока ймовірність виникнення пожеж в зв'язку з наявністю великої кількості паперу;
- підвищена напруга в електричній мережі, замикання якої може відбутися через тіло людини;
- вплив неіонізуючих електромагнітних випромінювань, електростатичних і магнітних полів, які виникають при роботі комп'ютерів;
- недостатній рівень параметрів мікроклімату і температури, відносної вологості повітря, швидкості руху повітря.

Робота співробітника вимагає від нього значної зорової уваги, і як наслідок, у приміщенні повинно бути правильно спроектоване і розташоване освітлення.

В залежності від часу доби і джерела світла на підприємстві використовується освітлення трьох видів: природне, штучне і комбіноване.

У денний час і ясну погоду використовується природне освітлення через світлові віконні пройми. Для нього характерна висока розсіяність і це є сприятливим фактором для здорової діяльності.

У передвечірній час і за несприятливих погодних умовах використовується комбіноване освітлення. Оскільки денне світло не забезпечує достатню освітленість робочого місця, то використовуються настільні лампи місцевого освітлення.

У вечірній час взимку використовується штучне освітлення. В якості джерел використовуються лампи накаливання.

Рекомендована і дійсна освітленість у приміщенні відділу товарноматеріальних цінностей складає 750лк.

Одним із найшкідливіших факторів, що чинить негативний вплив на працездатність робітників, є шум. Джерелами шуму у відділі є вентилятори, ЕОМ.

Для забезпечення нормативного рівня шуму у виробничому приміщенні відділу і на робочих місцях використовуються шумопоглинаючі засоби: важкогорючі спеціальні перфоровані плити, мінеральна вата з максимальним коефіцієнтом звукопоглинання в межах частот 31,5-8000Гц.

Рівень вібрації при виконанні робіт з використанням ЕОМ не повинен перевищувати допустимих значень. Оскільки в бюро використовуються 2 ПК з одним струйним принтером, то рівень вібрації, який створюють один принтер і два процесори, знаходиться в межах допустимих норм.

У відділі є меблі, які являються швидкозагораючим предметом. Існує ризик у випадку виникнення пожежі, швидке її поширення по приміщенню.

Електричний струм, проходячи через тіло людини справляє тепловий, хімічний, біологічний вплив, порушуючи нормальну життєдіяльність організму. Будь-який з цих впливів може привести до електротравм.

У відділі є холодильник, електрочайник, кондиціонер, касовий апарат. Електровмикання здійснюється через трьохштирьові штепсельні вилки, які заземлені на штатну систему заземлення.

Робота комп'ютера супроводжується електромагнітним і електростатичним випромінюванням. Навколо електростатично-зарядженого монітору підвищується концентрація пилу. Такий електризований пил може викликати запалення шкіри, а електромагнітне випромінювання може призвести до зниження загальної продуктивності співробітника.

Потужність дози рентгенівського випромінювання у відділі на відстані 0,05м від екрану не перевищує  $7,74 \cdot 10^{-12}$  А/кг, а вміст озону в повітрі робочої зони відповідає еквівалентній дозі 0,1мг/м<sup>3</sup>, окислів азоту – 5мг/м<sup>3</sup>, вміст пилу – 4мг/м<sup>3</sup>.

Мікрокліматичні параметри (температура, відносна вологість повітря, швидкість руху повітря) впливають у першу чергу на продуктивність праці і

самопочуття людини, а також на надійність роботи обчислювальної техніки. Забезпечення достатньої роботи системи вентиляції і опалення також підтримують працездатність співробітника.

Приміщення відділу обладнане системою опалення, кондиціонування повітря. Параметри мікроклімату, які діють у відділу, відповідають встановленим нормам

## 5.2 Заходи щодо зменшення впливу шкідливих факторів роботи за персональним комп'ютером

У процесі роботи співробітники відділу піддаються впливу небезпечних і шкідливих факторів, серед яких:

Для зменшення впливу небезпечних і шкідливих факторів у приміщенні необхідно здійснити ряд заходів.

- недостатнє освітлення робочого місця;
- підвищені рівні шуму і вібрації, спричинені функціонуванням вентиляторів і принтерів;
- висока ймовірність пожежі через велику кількість паперових матеріалів;
- надмірна напруга в електромережі, яка може спричинити ураження струмом;
- вплив неіонізуючого електромагнітного випромінювання та електростатичних полів, що виникають під час роботи комп'ютерів;
- невідповідні параметри мікроклімату, зокрема температура, вологість і швидкість руху повітря.

Робота співробітників вимагає значного зорового напруження, що робить правильне проектування та розташування освітлення важливим елементом організації робочого місця. Освітлення у приміщенні залежить від часу доби та погодних умов і поділяється на природне, штучне та комбіноване.

У денний час за ясної погоди використовується природне світло через вікна, що сприяє комфортній роботі. Увечері чи за несприятливих умов додається комбіноване освітлення, зокрема настільні лампи. Увечері взимку застосовують виключно штучне освітлення за допомогою ламп розжарювання. Рекомендована яскравість освітлення у відділі становить 750 лк.

Одним із основних шкідливих чинників є шум, джерелами якого виступають вентилятори та комп'ютери. Для зменшення його рівня у приміщенні використовуються звукопоглинаючі матеріали, такі як спеціальні перфоровані плити й мінеральна вата.

Рівень вібрації при роботі з електронно-обчислювальними машинами не перевищує допустимих норм. У відділі встановлено два комп'ютери та струменевий принтер, які забезпечують дотримання встановлених стандартів.

Окрему небезпеку становлять легкозаймисті меблі, які у разі пожежі можуть сприяти її швидкому поширенню. Електричний струм, проходячи через тіло людини, викликає теплові, хімічні та біологічні ефекти, що можуть призвести до травм. Усі електроприлади, такі як холодильник, чайник і касовий апарат, підключені через заземлені трьохконтактні вилки.

Робота комп'ютерів супроводжується електромагнітними та електростатичними випромінюваннями, які можуть підвищувати концентрацію пилу навколо моніторів. Це може спричинити подразнення шкіри та зниження загальної продуктивності праці.

Параметри мікроклімату в приміщенні, такі як температура, вологість і швидкість руху повітря, впливають на самопочуття співробітників і ефективність роботи обладнання. У відділі забезпечені системи опалення та кондиціонування, які відповідають встановленим нормам і сприяють підтриманню продуктивності.

## 6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Охорона довкілля – це сукупність заходів, спрямованих на зменшення негативного впливу людської діяльності на живу і неживу природу. Загальне управління у сфері захисту навколишнього середовища здійснюють Кабінет Міністрів України та місцеві органи виконавчої влади. Спеціалізованим органом державного управління у цій галузі є Міністерство охорони довкілля України та його представництва на місцях.

Основні нормативно-правові документи, що регулюють природоохоронну діяльність, включають Закони України: «Про охорону навколишнього природного середовища» (1991 р.), «Про охорону атмосферного повітря» (1992 р.), «Про природно-заповідний фонд України» (1992 р.), «Про тваринний світ» (1993 р.), «Про карантин рослин» (1993 р.) та інші.

Правові засади охорони довкілля базуються на нормах земельного, водного, лісового законодавства, а також законодавства про надра.

Система охорони навколишнього середовища включає:

- Правову охорону, що закріплює екологічні принципи у вигляді юридичних норм.
- Матеріальне стимулювання, яке сприяє економічній вигідності природоохоронної діяльності.
- Інженерну охорону, що впроваджує технології збереження ресурсів і охорони природи.

Мета охорони довкілля полягає у покращенні екологічного стану шляхом зменшення викидів у повітря, створення заповідників і національних парків для збереження природних комплексів, обмеження вилову риби та полювання на окремі види, а також боротьби з несанкціонованим викиданням відходів.

Особливу стурбованість викликає забруднення атмосфери шкідливими газами, що сприяє утворенню озонових «дірок» і посиленню парникового ефекту. Озонові «дірки» знижують захисну функцію атмосфери від ультрафіолетового

випромінювання, що призводить до збільшення захворювань людей і тварин. Парниковий ефект викликає глобальне потепління, танення льодовиків, підвищення рівня Світового океану, зміни у кліматичних процесах і деградацію екосистем.

У період науково-технічної революції значно зросли викиди газів і аерозолів техногенного походження. Найбільшу хімічну активність у взаємодії з компонентами атмосфери проявляють сполуки сірки, азоту, фосфору, галогенів, фенолів і формальдегідів. Щорічно в атмосферу потрапляють сотні мільйонів тонн цих речовин, що робить питання захисту повітря критично важливим.

### 6.1 Забруднення навколишнього середовища

Комп'ютерна компанія – це офісна організація, яка спеціалізується на розробці та підтримці програмного забезпечення. Незважаючи на відсутність промислового виробництва, що супроводжується викидами шкідливих речовин у повітря, воду чи ґрунт, у діяльності такої компанії можуть бути джерела забруднення довкілля.

До таких джерел належать:

- дизельна електростанція;
- твердопаливні котли;
- побутові відходи;
- електромагнітне випромінювання;
- відпрацьована оргтехніка та обладнання.

В офісі компанії працюють до 50 комп'ютерів, телефонів і велика кількість серверів. Аварійне або планове відключення електроенергії може значно ускладнити роботу співробітників. Щоб цього уникнути, на території компанії встановлено дизельну електростанцію, яка функціонує як у резервному, так і в аварійному режимах. Її робота супроводжується викидами продуктів згоряння дизельного палива та шумовим забрудненням.

У холодний період року для опалення офісу використовуються твердопаливні котли, які працюють на деревних пелетах. Хоча вони екологічніші за вугілля чи мазут, повністю екологічно чистим цей вид палива не є.

Побутові відходи, що утворюються у процесі роботи компанії, бувають рідкими (стічні води, залишки з миття посуду чи туалетні відходи) та твердими (папір, пластик, харчові відходи). Їх накопичення порушує санітарні норми, тому важливо організувати правильне збирання та утилізацію.

Застаріле комп'ютерне обладнання також стає джерелом екологічних проблем. Оргтехніка містить багато деталей, що підлягають переробці. Сьогодні утилізація таких пристроїв є нагальною необхідністю. До комп'ютерного лому належать монітори, системні блоки, принтери, флеш-накопичувачі, жорсткі диски, клавіатури та миші. Їх компоненти вимагають спеціальної обробки, адже багато з них містять небезпечні речовини, які при неправильному поводженні отруюють воду й ґрунт.

Крім того, офісні приміщення є джерелом електромагнітного забруднення через велику кількість комп'ютерної техніки, що працює по 8 годин на день, а іноді й цілодобово. Випромінювачами є монітори, процесори, а також мікрохвильові печі, які використовуються у кухнях. Електромагнітне випромінювання негативно впливає на здоров'я людей, викликаючи головний біль, порушення сну та інші проблеми.

## 6.2 Розробка заходів щодо зменшення забруднення

Охорона природи, раціональне використання ресурсів та забезпечення екологічної безпеки є невіддільною умовою сталого економічного та соціального розвитку України. В Україні екологічні відносини регулюються Законом «Про охорону навколишнього природного середовища», а також іншими законодавчими актами: земельним, водним, лісовим законодавством, нормами про надра, охорону атмосферного повітря, захист і раціональне використання рослинного і тваринного світу.

Для зменшення забруднення довкілля необхідно впроваджувати ефективні заходи, які мінімізують цей вплив у випадках, коли його неможливо повністю уникнути:

- Очищення димових газів – електрофільтри на великих енергоблоках (300 МВт і більше) забезпечують глибоке очищення димових газів. Мокрі золоуловлювачі, що працюють із низькими витратами води і невеликим перепадом тиску, ефективно вловлюють частинки золи на стінках із плівкою води завдяки відцентровій силі. Ефективність таких установок сягає 90%.

- Перехід на природний газ – для опалення офісів доцільно використовувати природний газ замість деревних пелетів, адже рівень шкідливих викидів від нього значно нижчий. Однак економічна доцільність такого переходу залежить від вартості палива.

- Раціональне поводження з відходами – важливо відправляти вторинну сировину (папір, скло, пластик) на переробку та мінімізувати обсяги відходів, повторно використовуючи ресурси, наприклад, скляні та пластикові пляшки. Зменшити кількість сміття допомагає також скорочення споживання пакування.

- Захист від електромагнітного випромінювання – для зменшення його впливу слід вимикати техніку, коли вона не використовується, та робити перерви в роботі з комп'ютерами. Мікрохвильові печі необхідно перевіряти на рівень випромінювання у майстрів.

- Утилізація електронної техніки – переробка старих комп'ютерів та оргтехніки дозволяє відновлювати до 95% матеріалів, зменшуючи обсяг відходів, які надходять на сміттєзвалища.

Фактично будь-який комп'ютер чи оргтехніку можна переробити та використати повторно. При правильному підході до утилізації близько 95% матеріалів із технічних відходів повертаються у виробництво, тоді як решта 5% відправляється на полігони чи на заводи з переробки побутових відходів.

Співвідношення ручної та автоматизованої праці на підприємствах із переробки залежить від типу техніки. Наприклад, для моніторів воно складає

приблизно 50 на 50, оскільки демонтаж кінескопів є складним процесом. У випадку системних блоків і оргтехніки частка автоматизованих операцій є більшою.

Якісна переробка комп'ютерної техніки відповідно до КВЕД передбачає підготовчі роботи:

- Низькорівневе форматування жорстких дисків.
- Очищення картриджів від тонера, що містить небезпечні хімічні речовини.

Загалом, електронний лом розбирається, сортується та обробляється. Пластикові, металеві й скляні компоненти оргтехніки переробляються окремо.

Перший етап переробки завжди виконується вручну. На цьому етапі видаляють усі небезпечні компоненти. У сучасних настільних комп'ютерах і принтерах таких елементів практично немає. Проте на переробку часто потрапляють пристрої, виготовлені наприкінці 1990-х – на початку 2000-х років, коли рідкокристалічні монітори ще не були поширені. У кінескопних моніторах міститься велика кількість свинцевих сполук. Ще одна категорія техніки, що містить небезпечні компоненти, – це ноутбуки. У батареях і дисплеях застарілих моделей є ртуть, яка є надзвичайно шкідливою для людини. У сучасних моделях ноутбуків ці компоненти вже не використовуються.

Далі видаляють великі пластикові деталі. Цей процес зазвичай також виконується вручну. Пластик сортують за типами та подрібнюють, щоб його можна було повторно використовувати. Інші залишки відправляють у великий подрібнювач-шредер, де всі подальші операції відбуваються автоматично. Багато технологій переробки запозичено з гірничої справи, де схожі методи використовуються для вилучення цінних металів із руди.

Подрібнені залишки комп'ютерів у вигляді гранул підлягають сортуванню. Спершу магнітами вилучають усі частини із заліза. Далі переходять до виділення кольорових металів, яких у ПК міститься значно більше. Наприклад, алюміній отримують із брухту методом електролізу. У сухому залишку залишається суміш пластику та міді. Мідь виділяють за допомогою флотації: гранули занурюють у спеціальну рідину, де пластик спливає, а мідь осідає на дні.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було вирішено завдання побудови нелінійної регресійної моделі для оцінювання розміру CMS на PHP.

Досягнуті результати:

- Виконано аналіз існуючих моделей оцінювання розміру веб-застосунків на PHP. Виявлено, що найбільш ефективними є нелінійні регресійні моделі, засновані на метриках діаграм класів. Доведено доцільність побудови моделі для CMS на PHP.

- Проведено пошук проектів на GitHub і зібрано метрики діаграм класів за допомогою PhpMetrics. Метрики включають: кількість класів, середню кількість методів на клас, глибину дерева наслідування.

- Виконано тестування емпіричних даних на нормальність, яке виявило негаусівський розподіл і наявність викидів. Викиди усунено за допомогою нормалізуючого перетворення  $\log_{10}$  та розрахунку квадрата відстані Махаланобіса.

- Побудовано трьохфакторні лінійну та нелінійну регресійні моделі. Нелінійна модель продемонструвала кращі значення  $R^2$ ,  $MMRE$  та  $PRED(0,25)$ , порівняно з лінійною.

- Розраховано інтервальні оцінки для обох моделей. Нелінійна модель показала менші ширини інтервалів, що свідчить про її вищу достовірність.

- Побудовану нелінійну регресійну модель порівняно з моделями до фреймворків Yii, Symfony, CakePHP. Порівняння демонструє, що побудова моделі для певного типу проектів дає кращі оцінки  $R^2$ ,  $MMRE$  та  $PRED(0,25)$ .

- Розраховано економічні показники щодо ефективності розробки ПЗ оцінювання розміру CMS на PHP, доведено економічну ефективність розробки.

- Розроблено розділи з охорони праці та охорони навколишнього середовища.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Dan S. N. Success Factors That Influence Agile Software Development Project Success. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*. 2016. Vol. 17, No 1, P. 172–222.
2. Boehm B. W. Software engineering economics. Englewood Cliffs, N.J : Prentice-Hall, 1981. 767 p.
3. International Software Benchmarking Standards Group. ISBSG Data Repository. URL: <https://www.isbsg.org/> (дата звернення: 12.10.2023).
4. Software Cost Estimation with COCOMO II / B. W. Boehm et al. Prentice Hall, 2000. 544 p.
5. Laird L. M., Brennan M. C. Software Measurement and Estimation: A Practical Approach (Quantitative Software Engineering Series). Wiley-IEEE Computer Society Pr, 2006. 280 p.
6. Zifen Yang. An improved software size estimation method based on object-oriented approach. *2012 IEEE Symposium on Electrical & Electronics Engineering (EEESYM)*, Kuala Lumpur, Malaysia, 24–27 June 2012. 2012. URL: <https://doi.org/10.1109/eeesym.2012.6258733> (date of access: 10.10.2023).
7. Trendowicz A., Jeffery R. Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success. Springer, 2014.
8. Kiewkanya M., Surak S. Constructing C++ software size estimation model from class diagram. *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, 13–15 July 2016. 2016. URL: <https://doi.org/10.1109/jcsse.2016.7748880> (date of access: 10.10.2023).
9. Prykhodko N. V., Prykhodko S. B. THE NON-LINEAR REGRESSION MODEL TO ESTIMATE THE SOFTWARE SIZE OF OPEN SOURCE JAVA-BASED SYSTEMS. *Radio Electronics, Computer Science, Control*. 2018. No. 3. URL: <https://doi.org/10.15588/1607-3274-2018-3-17> (date of access: 10.10.2023)

10. Prykhodko N. V., Prykhodko S. B. NON-LINEAR REGRESSION MODEL TO ESTIMATE THE SOFTWARE SIZE OF VB-BASED INFORMATION SYSTEMS. *Information Technology And Computer Engineering*. 2018. Vol. 43, no. 3. P. 37–42. URL: <https://doi.org/10.31649/1999-9941-2018-43-3-37-42> (date of access: 10.10.2023).

11. Prykhodko S. B., Prykhodko N. V., Koltsov A. V. A NONLINEAR REGRESSION MODEL FOR EARLY LOC ESTIMATION OF OPEN-SOURCE KOTLIN-BASED APPLICATIONS. *Radio Electronics, Computer Science, Control*. 2024. No. 1. P. 85. URL: <https://doi.org/10.15588/1607-3274-2024-1-8> (date of access: 15.11.2024).

12. Макарова Л. М., Латанська Л. О., Нікітін О. В., Нікітіна О. Ю. МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ РОЗМІРУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КРОС-ПЛАТФОРМНОЇ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ ІЗ ВИКОРИСТАННЯМ ПЛАТФОРМИ XAMARIN. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2022. Т. 33 (72), № 1. С. 150–156.

13. Tan H. B. K., Zhao Y., Zhang H. Estimating LOC for information systems from their conceptual data models. *Proceeding of the 28th international conference, Shanghai, China*, 20–28 May 2006. New York, New York, USA, 2006. URL: <https://doi.org/10.1145/1134285.1134331> (date of access: 10.10.2023).

14. Tan H. B. K., Zhao Y., Zhang H. Conceptual data model-based software size estimation for information systems. *ACM Transactions on Software Engineering and Methodology*. 2009. Vol. 19, no. 2. P. 1–37. URL: <https://doi.org/10.1145/1571629.1571630> (date of access: 10.10.2023).

15. Prykhodko S. B., Prykhodko N. V., Makarova L. M. Estimating the Software Size of Open-Source PHP-Based Systems Using Non-Linear Regression Analysis. *Proceedings of International Conference “Advanced Computer Information Technologies” (ACIT-2018)* : CEUR Workshop Proceedings, Ceske Budejovice, CZECH REPUBLIC. Ceske Budejovice, 2019. P. 199–202.

16. Приходько С.Б., Приходько Н.В. Нелінійне регресійне рівняння для оцінювання розміру програмного забезпечення інформаційних систем з відкритим

кодом на PHP. *Вісник східноукраїнського національного університету імені Володимира Даля*. 2018. № 6 (247). С. 135–139.

17. Prykhodko S. B., Shutko I. S., Prykhodko A. S. A NONLINEAR REGRESSION MODEL TO ESTIMATE THE SIZE OF WEB APPS CREATED USING THE CAKEPHP FRAMEWORK. *Radio Electronics, Computer Science, Control*. 2022. No. 4. P. 129–139. URL: <https://doi.org/10.15588/1607-3274-2021-4-12> (date of access: 10.10.2023).

18. Prykhodko S., Shutko I., Prykhodko A. Early LOC Estimation of Web Apps Created Using Yii Framework by Nonlinear Regression Models. *WSEAS TRANSACTIONS ON COMPUTERS*. 2021. Vol. 20. P. 321–328. URL: <https://doi.org/10.37394/23205.2021.20.35> (date of access: 10.10.2023).

19. Prykhodko S., Shutko I., Prykhodko A. Early size estimation of web apps created using CodeIgniter framework by nonlinear regression models. *RADIOELECTRONIC AND COMPUTER SYSTEMS*. 2022. No. 3. P. 84–94. URL: <https://doi.org/10.32620/reks.2022.3.06> (date of access: 10.10.2023).

20. Приходько С. Б., Приходько Н. В., Ворона М. В., Беловол І. О. Нелінійна регресійна модель для оцінювання розміру Web-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*. 2021. Т. 50, № 1. С. 115–121.

21. Латанська Л. О., Савчук Г. Ю. РАННЄ ОЦІНЮВАННЯ РОЗМІРУ CMS МООВОЮ PHP. *Інформаційні технології: моделі, алгоритми, системи* : зб. тез V всеукраїнської науково-практичної інтернет конференції, 30-31 жовтня. 2024 р. С 55–56

22. Usage statistics and market share of WordPress. URL: <https://w3techs.com/technologies/details/cm-wordpress>. (дата звернення: 12.10.2023).

23. Usage statistics of PHP for websites. URL: <https://w3techs.com/technologies/details/pl-php>. (дата звернення: 12.10.2023).

24. Приходько С. Б., Макарова Л. М., Приходько Н. В., Пухалевич А. В. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни «Емпіричні методи програмної інженерії». Миколаїв : НУК, 2023. 56 с.

25. PRYKHODKO N. V., PRYKHODKO S. B. Constructing the Nonlinear Regression Models on the Basis of Multivariate Normalizing Transformations. *Elektronnoe modelirovanie*. 2018. Vol. 40, no. 6. P. 101–110. URL: <https://doi.org/10.15407/emodel.40.06.101> (date of access: 10.10.2023).

26. Prykhodko S., Prykhodko N. Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations. *Advances in Intelligent Systems and Computing*. Cham, 2020. P. 166–174. URL: [https://doi.org/10.1007/978-3-030-58124-4\\_16](https://doi.org/10.1007/978-3-030-58124-4_16) (date of access: 10.10.2023).

27. MARDIA K. V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*. 1970. Vol. 57, no. 3. P. 519–530. URL: <https://doi.org/10.1093/biomet/57.3.519> (date of access: 10.10.2023).

28. Frost J. Regression Analysis: An Intuitive Guide for Using and Interpreting Linear Models. Statistics By Jim Publishing, 2020. 355 p.

29. Detecting bivariate outliers on the basis of normalizing transformations for non-Gaussian data : thesis / S. Prykhodko et al. 2017. URL: <http://essuir.sumdu.edu.ua/handle/123456789/55754> (date of access: 10.10.2023).

30. Prykhodko S. B., Prykhodko N. V., Makarova L. M. Pukhalevych A. V. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate 68 Non-Gaussian Data. Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) : LvivSlavske, Ukraine, February 20–24. Lviv-Slavske, 2018. P. 962–965.

31. Arkes J. Regression Analysis: A Practical Introduction. Taylor & Francis Group, 2019. 342 p.

32. Приходько С. Б., Макарова Л. М., Пугаченко К. С. Методичні вказівки та завдання до виконання лабораторних робіт з дисципліни «Обробка експериментальних даних на комп'ютері». Миколаїв : НУК, 2018. 76 с.

33. Foss T., Stensrud E., Kitchenham B., Myrtveit I. A Simulation Study of the Model Evaluation Criterion MMRE // *IEEE Transactions on Software Engineering*. – 2003. – Vol. 29, No. 11. – P. 985–995.300.

34. Prykhodko S., Prykhodko N. Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate.

## ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

### Вступ

Найменування програми: «LOC CMS».

### 1 Підстави для розробки

Підставою для розробки є наказ на затвердження тем кваліфікаційних робіт магістрів.

### 2 Призначення розробки

#### 2.1 Функціональне призначення

Функціональним призначенням програми є надання користувачу можливості оцінити розмір систем керування вмістом на мові РНР.

#### 2.2 Експлуатаційне призначення

Програма призначена для експлуатації на персональних комп'ютерах користувачів для оцінювання розміру систем керування вмістом на мові РНР.

### 3 Вимоги до програми або програмного продукту

#### 3.1 Вимоги для функціональних характеристик

##### 3.1.1 Вимоги до переліку виконуваних функцій

Розроблена програма має надавати можливість виконувати наступні функції:

- Оцінювання розміру систем керування вмістом на мові РНР у тисячах строк коду.

- Оцінювання довірчого інтервалу значення розміру систем керування вмістом на мові РНР.

- Оцінювання інтервалу передбачення значення розміру систем керування вмістом на мові РНР.

### 3.1.2 Вимоги для організації вхідних та вихідних даних

Введення вхідних даних відбувається за рахунок заповнення полів графічного інтерфейсу користувача.

Результати виводяться у відповідні поля графічного інтерфейсу користувача.

Вхідними даними для побудови нелінійної регресійної моделі є набір метрик з проектів систем керування вмістом на мові PHP.

Вхідними даними для проведення оцінювання розміру CMS є такі метрики програмного забезпечення як кількість класів (Classes), кількість методів на кожен клас (MbC), глибина дерева наслідування (DIT). Також для виконання оцінювання необхідно вказати довірчу ймовірність.

Кількість класів CMS— це ціле число, яке більше за одиницю.

Середня кількість методів на кожен клас, глибина дерева наслідування та довірча ймовірність, фактичний розмір – дійсні додатні числа. Довірча ймовірність приймає значення не більші за одиницю.

Вихідними даними є отримана оцінка розміру CMS у тисячах рядків коду, оцінки довірчого інтервалу та інтервалу передбачення, які буде відображено у відповідних полях графічного інтерфейсу програми.

### 3.2 Вимоги до надійності

Надійне (стійке) функціонування програмного забезпечення повинно бути забезпечено виконанням сукупності організаційно-технічних заходів, виконанням валідації вхідних даних згідно вимог та відображення повідомлень про виявлення некоректних даних з закликом ввести коректні.

### 3.3 Умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, що пред'являються до технічних засобів в частині умов їх експлуатації.

### 3.4 Вимоги до технічної та програмної сумісності

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2Гб (1 Гб мінімум), простір на жорсткому диску 100 Мб мінімум.

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

### 3.5 Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм не висуваються.

### 3.6 Вимоги до маркування та упаковки

Вимоги до маркування та упаковки не висуваються.

### 3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

## 4 Вимоги до програмної документації

Склад програмної документації має включати в себе:

- 1) Технічне завдання;
- 2) Програму та методику випробувань;
- 3) Інструкцію користувача;
- 4) Опис програмного забезпечення.
- 5) Текст програми

## 5 Техніко-економічні показники

Економічна ефективність від впровадження програмного забезпечення була розрахована у розділі 4. Виходячи з одержаних результатів, впровадження даного програмного забезпечення є доцільним, приблизний термін окупності – 4 місяці.

## 6 Стадії та етапи розробки програмного забезпечення

Стадії та етапи виконання розробки програмного забезпечення можна представити наступним чином:

Таблиця А.1 – Стадії та етапи розробки

| Стадії розробки   | Назва етапу                      | Термін виконання |
|-------------------|----------------------------------|------------------|
| Технічне завдання | Розробка технічного завдання     | 29.09.2024       |
|                   | Затвердження технічного завдання | 08.10.2024       |
| Ескізний проект   | Розробка ескізного проекту       | 24.10.2024       |
|                   | Затвердження ескізного проекту   | 31.10.2024       |
| Технічний проект  | Розробка технічного проекту      | 14.11.2024       |
|                   | Затвердження технічного проекту  | 29.11.2024       |
| Робочий проект    | Розробка програми                | 30.11.2024       |
|                   | Розробка програмної документації | 04.12.2024       |
|                   | Випробування програми            | 05.12.2024       |

## 7 Порядок контролю та прийомки

Приймально-здавальні роботи мають проводитись під наглядом Замовника або уповноваженої ним особи.

Приймально-здавальні випробування повинні проводитись згідно документу «Програма та методика випробувань», який узгоджується між стороною розробника та стороною замовника.

Хід проведення приймально-здавальних робіт Замовник та Виконавець документують в «Протокол проведення випробувань».

На основі Протоколу проведення випробувань Виконавець та Замовник підписують «Акт прийому-здачі програми у експлуатацію».

## ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

### 1 Об'єкт випробувань

Об'єктом дослідження є програмне забезпечення для оцінювання розміру систем керування вмістом на мові РНР

### 2 Мета випробувань

Мета випробувань – перевірка працездатності програмного застосунку, перевірка відповідності фактичних характеристик програмного застосунку тим вимогам, які було викладено у технічному завданні.

### 3 Вимоги до застосунку

Під час виконання випробувань перевіряються функціональні характеристики (вимоги), що були викладені в технічному завданні.

Розроблена програма має надавати можливість виконувати наступні функції:

- Оцінювання систем керування вмістом на мові РНР у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру систем керування вмістом на мові РНР.
- Оцінювання інтервалу передбачення значення систем керування вмістом на мові РНР.

### 4 Вимоги до програмної документації

Програмна документація включає в себе наступні документи:

- 1) технічне завдання;
- 2) програму та методику випробувань;
- 3) інструкцію користувача;
- 4) опис програмного забезпечення.
- 5) текст програми

## 5 Засоби і порядок випробувань

### 5.1 Технічні засоби, що використовуються під час випробувань

Склад технічних засобів, що використовувались під час випробувань:

HP Victus Gaming Laptop 15-fa0022ua.

– Восьмиядерний Intel Core i5-12450H (2.0 - 4.4 ГГц).

– Оперативна пам'ять: 16 Гбайт, DDR4-3200 МГц.

– Накопичувач даних: SSD 512 Гбайт.

– Графіка: nVidia GeForce RTX 3050, 4 Гбайт GDDR6.

– Екран: 15.6" IPS Full HD (1920x1080), 144 Гц.

### 5.2 Програмні засоби, що використовувались під час випробувань

Операційна система Windows 10 Pro 21H2 204.2125

### 5.3 Порядок проведення випробувань

Порядок проведення випробувань – перевірка відповідності функціональних характеристик програмного застосунку, перевірка вимог до програмної документації.

## 6 Методи випробувань

### 6.1 Методика проведення перевірки вимог до програмної документації

За відсутності конкретних вимог до оформлення та подання програмної документації, розроблені програмні документи перевіряються представником замовника візуально. Уповноважена особа перевіряє відповідність розроблених програмних документів з тими, що були вказані у пункті «Склад програмної документації, що пропонується на випробування».

### 6.2 Методика проведення перевірки вимог до функціональних характеристик програмного продукту

Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» технічного завдання.

Перевірку можна вважати успішно завершеною у випадку відповідності виконуваних функцій технічному завданню.

Порядок випробувань:

- Перевірка роботи застосунку на моніторах з різною розподільною здатністю.
- Перевірка відображення шрифтів.
- Перевірка кожної форми графічного інтерфейсу на можливість введення даних.
- Перевірка кожної форми графічного інтерфейсу на введення невалідних даних.
- Перевірка можливості виконання оцінювання розміру, визначення довірчого інтервалу та інтервалу передбачення для систем керування вмістом на мові РНР.

## 7 Результати випробувань

За результатами випробувань було підтверджено відповідність функціональних можливостей висунутим вимогам до програмного забезпечення.

## 8 Відомості про відмови, збої та аварійні ситуації

Під час проведення випробувань відмов, збоїв та аварійних ситуацій помічено не було

## 9 Відомості про коригування параметрів об'єкта випробувань та технічної документації

Коригування параметрів об'єкта випробувань та технічної документації в процесі виконання випробувань не проводилось

## ДОДАТОК В – ІНСТРУКЦІЯ КОРИСТУВАЧА

### Вступ

Інструкція користувача призначена для ознайомлення користувача з правилами та порядком виконання операцій, які забезпечуються роботою програмного забезпечення.

Ця програма призначена для оцінювання розміру та визначення інтервальних оцінок розміру у тисячах строк коду систем керування вмістом на мові РНР. Програма є вільним програмним забезпеченням з графічним інтерфейсом користувача з можливістю вводити дані та переглядати результати.

### Загальні відомості про програму

Найменування програми: «LOC CMS».

### Використані мови програмування

Програма написана на мові Python, для розробки графічного інтерфейсу користувача було використано сумісну з Python бібліотеку Qt.

### Призначення програми

Програма призначена для оцінювання розміру (в тисячах рядків коду) систем керування вмістом на мові РНР на основі таких метрик як кількість класів (Classes, Number of Classes), середня кількість методів на клас (MBC, Methods by Class), глибина дерева наслідування (DIT, Depth of Inheritance Tree).

### Функціональні можливості програми

- Оцінювання розміру систем керування вмістом на мові РНР у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру систем керування вмістом на мові РНР.
- Оцінювання інтервалу передбачення значення розміру систем керування вмістом на мові РНР.

### Обмеження області застосування програми

Область застосування програми обмежена галуззю розробки програмного

забезпечення.

### **Умови, необхідні для використання програми**

Спеціальні умови для використання програми відсутні

### **Вимоги до технічної сумісності та програмного середовища**

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2Гб (1 Гб мінімум), простір на жорсткому диску 100 Мб мінімум.

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

### **Стартове вікно програми та вкладка для введення даних**

Загальний вигляд стартового вікна програми «LOC CMS» наведено на рисунку В.1.

Стартове вікно (вкладка «File») містить поле для вибору файлу, на основі якого буде побудовано нелінійну регресійну модель

На вкладці «Inputs» є наступні поля для введенні даних для виконання оцінювання:

- Поле «Number of Classes» дозволяє вказати кількість класів оцінюваного веб-застосунку. В якості кількості класів можна ввести лише невід'ємне число, більше одиниці.

- Поле «Average Methods per Class» дозволяє вказати середню методів у класах оцінюваного веб-застосунку. В якості середньої кількості класів можна ввести дійсне невід'ємне число, більше одиниці.

- Поле «Inheritance Depth» дозволяє вказати середнє значення глибини дерева наслідування у класах оцінюваного веб-застосунку. В якості глибини дерева наслідування можна ввести дійсне невід'ємне число.

- Поле «Confidence Level» для вказання довірчої ймовірності моделювання. В якості довірчої ймовірності можна ввести дійсне невід'ємне число до одиниці включно.

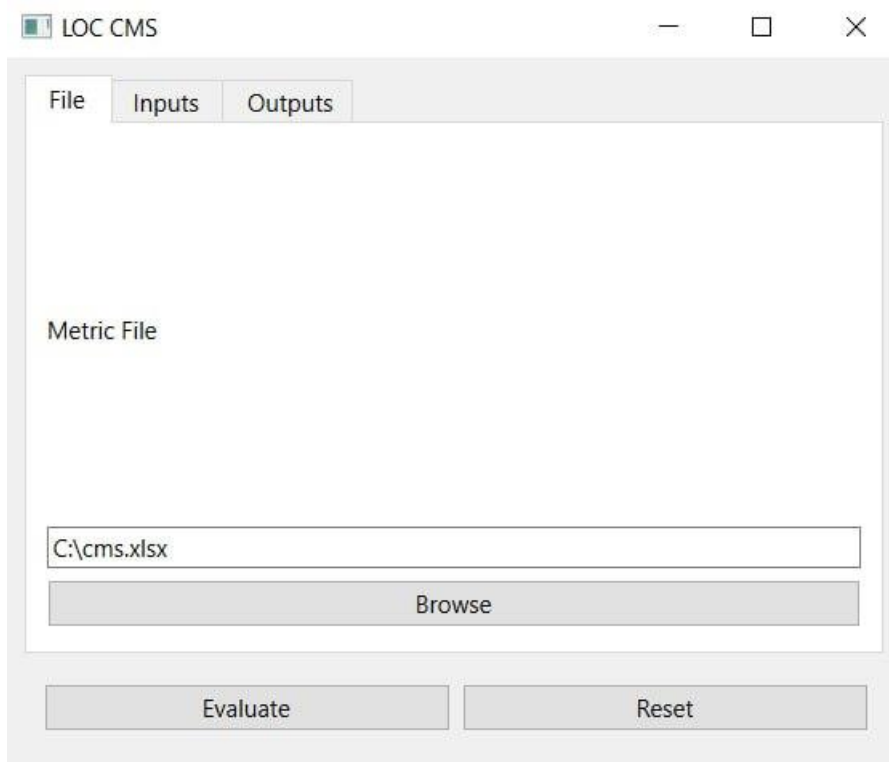


Рисунок В.1 – Стартове вікно програми

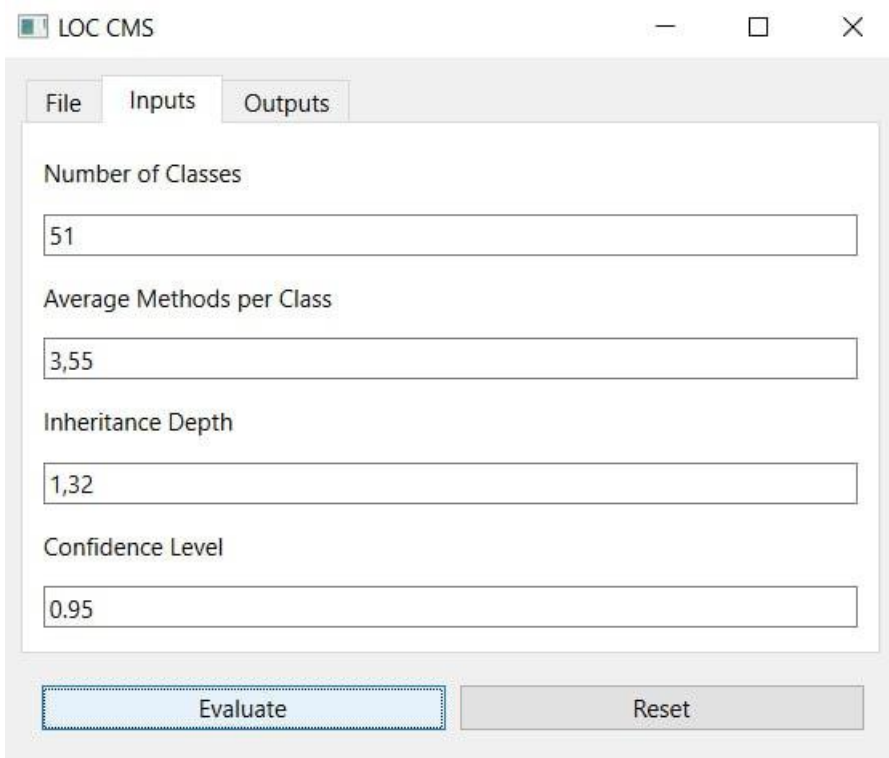
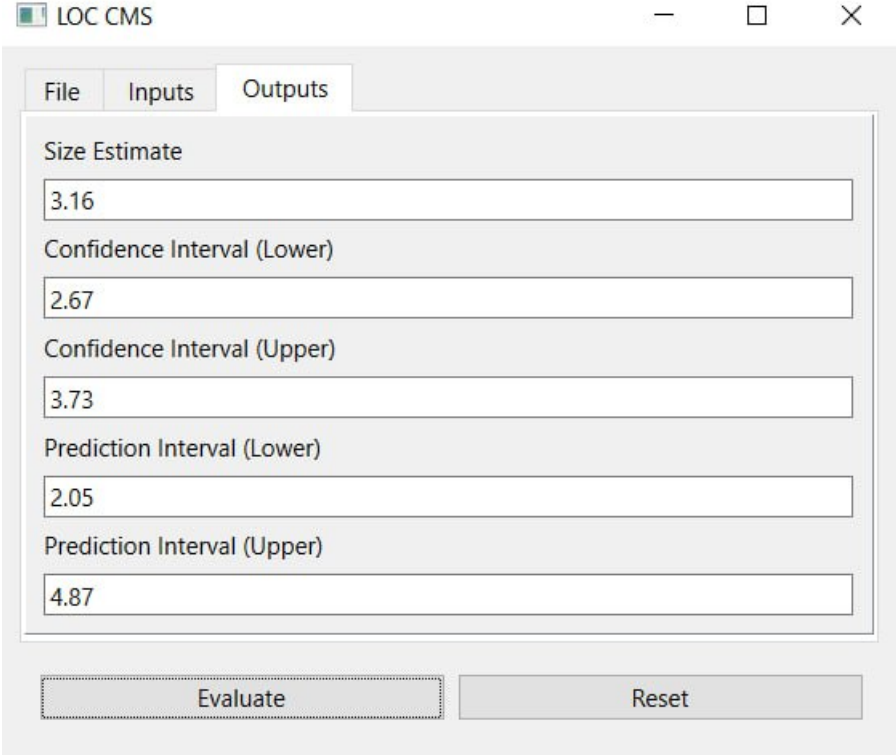


Рисунок В.2 – Вигляд вікна програми до виконання оцінювання

### Результати оцінювання розміру систем керування вмістом на мові PHP

Після введення вхідних даних та ініціювання процедури виконання обчислень за допомогою відповідної кнопки, програмне забезпечення виконає розрахунок відповідних оцінок. Результати оцінювання розміру веб-застосунку (його розмір у тисячах строк коду, довірчі інтервали та інтервали прогнозування нелінійної регресії), будуть показані в полях в нижній частині екрану.

Результат виконання оцінювання можна побачити на рисунку В.3.



The screenshot shows a window titled "LOC CMS" with three tabs: "File", "Inputs", and "Outputs". The "Outputs" tab is active and displays the following results:

| Category                    | Value |
|-----------------------------|-------|
| Size Estimate               | 3.16  |
| Confidence Interval (Lower) | 2.67  |
| Confidence Interval (Upper) | 3.73  |
| Prediction Interval (Lower) | 2.05  |
| Prediction Interval (Upper) | 4.87  |

At the bottom of the window, there are two buttons: "Evaluate" and "Reset".

Рисунок В.3 – Результат оцінювання систем керування вмістом на мові PHP

## ДОДАТОК Г – ТЕКСТ ПРОГРАМИ

```
import pandas as pd

from MetricsAnalyzer import MetricsAnalyzer
import sys
from PyQt6.QtWidgets import (QApplication, QWidget, QFileDialog, QPushButton, QLabel,
QGroupBox, QLineEdit, QVBoxLayout, QMainWindow, QHBoxLayout, QTabWidget, QScrollArea)
from PyQt6.QtGui import (QDoubleValidator, QIntValidator)
from pathlib import Path

class AppWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('LOC CMS')

        # Widgets initialization
        self.fileInput = QLineEdit(self)
        browseButton = QPushButton('Browse')
        browseButton.clicked.connect(self.selectFileDialog)

        self.methodsInput = QLineEdit(self)
        self.classesInput = QLineEdit(self)
        self.depthInput = QLineEdit(self)
        self.confidenceInput = QLineEdit("0.95", self)

        self.methodsInput.setValidator(QDoubleValidator(self))
        self.classesInput.setValidator(QIntValidator(self))
        self.depthInput.setValidator(QDoubleValidator(self))
        self.confidenceInput.setValidator(QDoubleValidator(0.0, 1.0, 2, parent=self))

        self.estimateOutput = QLineEdit(self)
        self.estimateOutput.setReadOnly(True)
        self.predictionLowerOutput = QLineEdit(self)
        self.predictionLowerOutput.setReadOnly(True)
        self.predictionUpperOutput = QLineEdit(self)
        self.predictionUpperOutput.setReadOnly(True)
        self.confidenceLowerOutput = QLineEdit(self)
        self.confidenceLowerOutput.setReadOnly(True)
        self.confidenceUpperOutput = QLineEdit(self)
        self.confidenceUpperOutput.setReadOnly(True)

        self.evaluateButton = QPushButton('Evaluate')
        self.evaluateButton.clicked.connect(self.performCalculation)
        self.resetButton = QPushButton('Reset')
        self.resetButton.clicked.connect(self.resetFields)
```

```

# Layouts
mainWidget = QWidget()
mainLayout = QVBoxLayout(mainWidget)

# Tabs for sections
tabWidget = QTabWidget()

# File tab
fileTab = QWidget()
fileLayout = QVBoxLayout()
fileLayout.addWidget(QLabel("Metric File"))
fileLayout.addWidget(self.fileInput)
fileLayout.addWidget(browseButton)
fileTab.setLayout(fileLayout)
tabWidget.addTab(fileTab, "File")

# Input tab
inputTab = QWidget()
inputLayout = QVBoxLayout()
inputLayout.addWidget(QLabel("Number of Classes"))
inputLayout.addWidget(self.classesInput)
inputLayout.addWidget(QLabel("Average Methods per Class"))
inputLayout.addWidget(self.methodsInput)
inputLayout.addWidget(QLabel("Inheritance Depth"))
inputLayout.addWidget(self.depthInput)
inputLayout.addWidget(QLabel("Confidence Level"))
inputLayout.addWidget(self.confidenceInput)
inputTab.setLayout(inputLayout)
tabWidget.addTab(inputTab, "Inputs")

# Output tab
outputTab = QScrollArea()
outputContent = QWidget()
outputLayout = QVBoxLayout()
outputLayout.addWidget(QLabel("Size Estimate"))
outputLayout.addWidget(self.estimateOutput)
outputLayout.addWidget(QLabel("Confidence Interval (Lower)"))
outputLayout.addWidget(self.confidenceLowerOutput)
outputLayout.addWidget(QLabel("Confidence Interval (Upper)"))
outputLayout.addWidget(self.confidenceUpperOutput)
outputLayout.addWidget(QLabel("Prediction Interval (Lower)"))
outputLayout.addWidget(self.predictionLowerOutput)
outputLayout.addWidget(QLabel("Prediction Interval (Upper)"))
outputLayout.addWidget(self.predictionUpperOutput)
outputContent.setLayout(outputLayout)
outputTab.setWidget(outputContent)
outputTab.setWidgetResizable(True)
tabWidget.addTab(outputTab, "Outputs")

# Buttons section

```

```

buttonsLayout = QHBoxLayout()
buttonsLayout.addWidget(self.evaluateButton)
buttonsLayout.addWidget(self.resetButton)
buttonsWidget = QWidget()
buttonsWidget.setLayout(buttonsLayout)

# Adding tabs and buttons to main layout
mainLayout.addWidget(tabWidget)
mainLayout.addWidget(buttonsWidget)

self.setCentralWidget(mainWidget)

def selectFileDialog(self):
    filename, _ = QFileDialog.getOpenFileName(
        self,
        "Select a File",
        "",
        "Data files (*.xlsx *.csv *.txt)"
    )
    if filename:
        path = Path(filename)
        self.fileInput.setText(str(path))

def resetFields(self):
    self.fileInput.setText('')
    self.methodsInput.setText('')
    self.classesInput.setText('')
    self.depthInput.setText('')
    self.confidenceInput.setText('0.95')
    self.estimateOutput.setText('')
    self.predictionLowerOutput.setText('')
    self.predictionUpperOutput.setText('')
    self.confidenceLowerOutput.setText('')
    self.confidenceUpperOutput.setText('')

def performCalculation(self):
    metric_file = self.fileInput.text()
    class_count_text = self.classesInput.text()
    method_avg_text = self.methodsInput.text()
    inheritance_depth_text = self.depthInput.text()
    conf_level_text = self.confidenceInput.text()

    if class_count_text == '' or method_avg_text == '' or inheritance_depth_text == '' or conf_level_text == '':
        return

    class_count = float(classes_text.replace(',', '.'))
    method_avg = float(methods_text.replace(',', '.'))
    inheritance_depth = float(dit_text.replace(',', '.'))
    conf_level = float(confidence_value_text.replace(',', '.'))

```

```

data_frame = pd.read_excel(file_path)
analyzer = MetricsAnalyzer(data_frame)

analyzer.perform_nonlinear_regression()

estimation = analyzer.estimate_project_loc(
    input_metrics=pd.DataFrame({'X1': [class_count], 'X2': [method_avg],
'X3': [inheritance_depth]}),
    confidence=conf_level
)

self.estimateOutput.setText('%.2f' % (estimation['regression_estimate'][0]))
self.predictionLowerOutput.setText('%.2f' %
estimation['pred_lower_bound'][0])
self.predictionUpperOutput.setText('%.2f' %
estimation['pred_upper_bound'][0])
self.confidenceLowerOutput.setText('%.2f' %
estimation['conf_lower_bound'][0])
self.confidenceUpperOutput.setText('%.2f' %
estimation['conf_upper_bound'][0])

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = AppWindow()
    window.show()
    sys.exit(app.exec())

import numpy as np
import pandas as pd
import math
from scipy.stats import norm
from scipy.stats import chi2
from scipy.stats import t
from scipy.stats import chisquare
from scipy.stats import normaltest
from scipy.stats import shapiro
from scipy.stats import chi2_contingency
from statistics import NormalDist
from sklearn import linear_model
import statsmodels.api as sm

class MetricsAnalyzer:
    def __init__(self, input_data):
        self.dataset = input_data.copy()
        self.num_columns = len(self.dataset.columns)
        self.num_features = self.num_columns - 1

    def perform_chi_square_test(self, input_data, num_intervals=6):
        total_entries = len(input_data)

```

```

mean_value = np.mean(input_data)
std_dev = np.std(input_data, ddof=1)

min_val = input_data.min()
max_val = input_data.max()
step_size = (max_val - min_val) / num_intervals

left_bound = min_val
right_bound = left_bound + step_size

chi_square_stat = 0
for i in range(num_intervals):
    if i == num_intervals - 1:
        interval_data = input_data[(input_data <= max_val) & (input_data >=
left_bound)]
    else:
        interval_data = input_data[(input_data < right_bound) & (input_data
>= left_bound)]

        prob = norm(mean_value, std_dev).cdf(right_bound) - norm(mean_value,
std_dev).cdf(left_bound)

        chi_square_stat += (len(interval_data) - total_entries * prob) ** 2 /
(total_entries * prob)

        left_bound = right_bound
        right_bound += step_size

return chi_square_stat

def compute_mahalanobis_distance(self, input_data):
    centered_data = input_data - np.mean(input_data, axis=0)

    covariance_matrix = np.cov(input_data.values.T, bias=True)

    inverse_covariance = np.linalg.inv(covariance_matrix)

    left_product = np.dot(centered_data, inverse_covariance)

    mahalanobis_values = np.dot(left_product, centered_data.T)
    return mahalanobis_values.diagonal()

def transform_metrics(self, transformation_function, data_subset=None):
    if data_subset is None:
        data_subset = self.dataset.copy()

    column_names = data_subset.columns

```

```

num_columns = len(column_names)

for i in range(num_columns):
    new_column_name = 'Transformed_' + column_names[i]
    data_subset[new_column_name] = data_subset.iloc[:, i].map(lambda x:
transformation_function(x))

return data_subset

def detect_outliers(self, data_subset):
    num_columns = self.num_columns
    while True:
        data_subset['distance_scores'] =
self.compute_mahalanobis_distance(input_data=data_subset.iloc[:,
num_columns:num_columns+num_columns])
        critical_value = chi2.ppf(1-0.05, self.num_columns)
        max_index = data_subset['distance_scores'].idxmax()
        max_distance = data_subset['distance_scores'].max()

        if max_distance >= critical_value:
            data_subset = data_subset.drop(index=max_index)
        else:
            break

    return data_subset

def calculate_non_linear_fit(self, data_subset, power_coeff, base_intercept):
    return (data_subset ** power_coeff).prod(axis=1) * 10 ** base_intercept

def compute_scalar_values(self, data_subset=None):
    column_names = data_subset.columns

    num_columns = len(column_names)

    self.transformed_mean = np.mean(data_subset, axis=0)
    centered_data = data_subset - np.mean(data_subset, axis=0)

    for i in range(num_columns):
        new_column_name = column_names[i] + '_square'
        centered_data[new_column_name] = centered_data.iloc[:, i].map(lambda x: x
* x)

    matrix_result = np.zeros((num_columns, num_columns))

    for i in range(num_columns):
        for j in range(num_columns):

```

```

        matrix_result[i][j] = (centered_data.iloc[:, i] *
centered_data.iloc[:, j]).sum()
        feature_data = centered_data.iloc[:, 0:self.num_features]

        output_result = np.array([])

        inverse_matrix = np.linalg.inv(matrix_result)
        self.transformation_matrix = inverse_matrix

        for _, row in feature_data.iterrows():
            temp_result = np.dot(np.dot(row, inverse_matrix), row.T)
            output_result = np.append(output_result, temp_result)

        return pd.DataFrame(output_result)

    def calculate_residual_std(self, residual_data):
        total_entries = len(residual_data)

        residual_sum =
residual_data[['linear_residuals']].pow(2).sum()['linear_residuals']

        return math.sqrt(1.0 / (total_entries - self.num_features - 1) *
residual_sum)

    def estimate_project_loc(self, input_metrics, confidence=0.05):
        input_metrics = self.transform_metrics(transformation_function=math.log10,
data_subset=input_metrics)

        centered_metrics = input_metrics.iloc[:, self.num_features:] -
self.transformed_mean

        input_metrics['regression_estimate'] = self.calculate_non_linear_fit(
            data_subset=input_metrics.iloc[:, 0:self.num_features],
            power_coeff=self.regressor.coef_,
            base_intercept=self.regressor.intercept_
        )

        input_metrics['scalar_product'] = np.dot(np.dot(centered_metrics,
self.transformation_matrix), centered_metrics.T)

        student_quantile = t.isf((1 - confidence) / 2, self.final_count -
self.num_features - 1)

        confidence_delta = student_quantile * self.residual_std * math.sqrt(1 /
self.final_count + float(input_metrics['scalar_product'][0]))
        prediction_delta = student_quantile * self.residual_std * math.sqrt(1 + (1 /
self.final_count) + float(input_metrics['scalar_product'][0]))

```

```

        input_metrics['conf_lower_bound'] = 10 **
(math.log10(input_metrics['regression_estimate'][0]) - confidence_delta)
        input_metrics['conf_upper_bound'] = 10 **
(math.log10(input_metrics['regression_estimate'][0]) + confidence_delta)
        input_metrics['pred_lower_bound'] = 10 **
(math.log10(input_metrics['regression_estimate'][0]) - prediction_delta)
        input_metrics['pred_upper_bound'] = 10 **
(math.log10(input_metrics['regression_estimate'][0]) + prediction_delta)

    return input_metrics

def perform_nonlinear_regression(self):
    data_subset = self.dataset

    while True:
        normalized_data =
self.transform_metrics(transformation_function=math.log10,
data_subset=data_subset.iloc[:, :self.num_columns])

        cleaned_data = self.detect_outliers(data_subset=normalized_data)

        cleaned_data = cleaned_data.reset_index().drop(labels='index', axis=1)

        predictors = cleaned_data.iloc[:, self.num_columns+1:2*self.num_columns]
        target = cleaned_data.iloc[:, self.num_columns]

        self.regressor = linear_model.LinearRegression()
        self.regressor.fit(predictors, target)

        regression_model = sm.OLS(target, predictors).fit()
        predicted_values = self.regressor.predict(predictors)

        cleaned_data['linear_regression'] = pd.DataFrame(predicted_values)
        cleaned_data['linear_residuals'] = cleaned_data['linear_regression'] -
cleaned_data['Transformed_Y']

        chi_critical = chi2.ppf(1-0.05, 3)
        chi_square_stat =
self.perform_chi_square_test(input_data=cleaned_data['linear_residuals'])

        cleaned_data['nonlinear_regression'] = self.calculate_non_linear_fit(
            data_subset=cleaned_data.iloc[:, 1:self.num_columns],
            power_coeff=self.regressor.coef_,
            base_intercept=self.regressor.intercept_
        )

        cleaned_data['scalar'] =
self.compute_scalar_values(data_subset=cleaned_data.iloc[:,
self.num_columns+1:self.num_columns*2])

```

```

self.final_count = len(cleaned_data)

self.residual_std =
self.calculate_residual_std(residual_data=cleaned_data)

student_quantile = t.isf(0.025, self.final_count - self.num_features - 1)

cleaned_data['conf_interval_delta'] = cleaned_data[['scalar']].map(lambda
x: student_quantile * self.residual_std * math.sqrt(1 / self.final_count + x))
cleaned_data['pred_interval_delta'] = cleaned_data[['scalar']].map(lambda
x: student_quantile * self.residual_std * math.sqrt(1 + (1 / self.final_count) + x))

cleaned_data['conf_lower_bound'] = 10 **
(cleaned_data['linear_regression'] - cleaned_data['conf_interval_delta'])
cleaned_data['conf_upper_bound'] = 10 **
(cleaned_data['linear_regression'] + cleaned_data['conf_interval_delta'])
cleaned_data['pred_lower_bound'] = 10 **
(cleaned_data['linear_regression'] - cleaned_data['pred_interval_delta'])
cleaned_data['pred_upper_bound'] = 10 **
(cleaned_data['linear_regression'] + cleaned_data['pred_interval_delta'])

outliers = cleaned_data[(cleaned_data.Y <= cleaned_data.pred_lower_bound)
| (cleaned_data.Y >= cleaned_data.pred_upper_bound)]

if chi_critical > chi_square_stat and outliers.empty:
    break
elif chi_critical < chi_square_stat:
    if abs(cleaned_data['linear_residuais'].max()) >
abs(cleaned_data['linear_residuais'].min()):
        residuals_max_index = cleaned_data['linear_residuais'].idxmax()
    else:
        residuals_max_index = cleaned_data['linear_residuais'].idxmin()

    cleaned_data = cleaned_data.drop(index=residuals_max_index)
    data_subset = cleaned_data
elif not outliers.empty:
    cleaned_data = cleaned_data.drop(outliers.index)
    data_subset = cleaned_data

```

## ДОДАТОК Д – ОПИС ПРОГРАМИ

### 1 Загальні відомості

Найменування: «LOC CMS».

#### 1.2 Програмне забезпечення, необхідне для функціонування програми

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

#### 1.3 Мови програмування

Програмне забезпечення для оцінювання розміру систем керування вмістом на мові PHP, розроблене з використанням мови програмування Python та використанням бібліотек Pandas, NumPy та SciPy для виконання математичних обчислень. Для створення графічного інтерфейсу користувача було використано сумісну з Python бібліотеку Qt.

### 2 Функціональне призначення

Функціональним призначенням програми є надання користувачу можливості оцінити систем керування вмістом на мові PHP.

Функції програмного застосунку, які доступні користувачу:

- Оцінювання розміру систем керування вмістом на мові PHP у тисячах строк коду.
- Оцінювання довірчого інтервалу значення розміру систем керування вмістом на мові PHP.
- Оцінювання інтервалу передбачення значення розміру систем керування вмістом на мові PHP.

## 2.1 Функціональні обмеження

Програмне забезпечення не вимагає використання зовнішніх ресурсів, тому функціональні обмеження відсутні.

## 3 Технічні засоби, що використовуються

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: 64-бітний процесор на архітектурі x86 або ARM, оперативна пам'ять 2Гб (1 Гб мінімум), простір на жорсткому диску 100 Мб мінімум.

## 4 Виклик та завантаження

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7/8/10/11, Ubuntu 17/18/19/20.

Запуск програмного застосунку відбувається за рахунок натискання на ярлик програмного застосунку.

## 5 Вхідні дані

Вхідними даними для побудови нелінійної регресійної моделі є файл з метриками з розширенням .txt/.csv/.xlsx в якому в першій колонці знаходиться залежна величина – кількість рядків коду у тисячах, а у другому-четвертому кількість класів, кількість методів на кожен клас та глибина дерева наслідування відповідно.

Вхідними даними для оцінювання систем керування вмістом на мові РНР є кількість класів, середня кількість методів на кожен класів, глибина дерева наслідування, довірна ймовірність.

Кількість класів CMS – це ціле число, яке більше за одиницю.

Значення середньої кількості методів, глибини дерева наслідування CMS застосунку мають бути дійсними додатними числами.

Довірна ймовірність вказує на відсотки та є дійсним числом від нуля до одиниці включно.

## 6 Вихідні дані

Вихідними даними є отримані оцінки розміру у тисячах рядків коду, оцінки довірчого інтервалу та інтервалу передбачення. Отримані значення буде відображено у відповідних полях графічного інтерфейсу програми.