

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

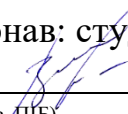
_____ проф. Приходько С.Б.

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»


на тему: Нелінійна регресійна модель для оцінювання якості web-застосунків, що створюються з використанням фреймворку Spring, та розробка програми для її реалізації

Виконав: студент групи 6151з


_____ Вірчак С.А.
(підпис, ПІБ)

Керівник роботи:

_____ ДОЦЕНТ, К.Т.Н.
(посада, науковий ступень вчене звання)


_____ Каіров В.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 Вірчаку Сергію Анатолійовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: Нелінійна регресійна модель для оцінювання якості web-застосунків, що створюються з використанням фреймворку Spring, та розробка програми для її реалізації

Керівник роботи Каіров Володимир Олексійович, доцент, к.т.н.

Затверджені наказом ректора № 1070-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; Розділ з розробки 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. Актуальність теми, 3. Мета та завдання дослідження, 4. Об'єкт і предмет дослідження, 5. Методи дослідження, 6. Наукова новизна одержаних результатів, 7. Практичне значення одержаних результатів, 8. Особистий внесок здобувача 9. Апробація результатів досліджень та публікації, 10. Вихідні дані, 11. Нелінійна регресійна модель для оцінювання метрики RFC web-застосунків на основі Spring-фреймворку, 12. Нелінійна регресійна модель для оцінювання метрики RFC застосунків з відкритим кодом на Java. 13. Порівняння якості моделей, 14. Діаграма варіантів використання, 15. Діаграма діяльності. 16. Діаграма класів, 17. Діаграма послідовності, 18. Інтерфейс користувача. 19.Висновки.

6. Консультанти розділів роботи

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

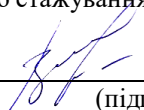
7. Дата видачі завдання 14.10.2024 р.

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) за результатами власних досліджень	21.10.2024	НС*
4.	Підготовка розділу з розробки 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	19.12.2024	

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

Студент


_____ (підпис)

С.А. Вірчак

_____ (ПІБ)

Керівник роботи


_____ (підпис)

В.О. Каіров

_____ (ПІБ)

РЕФЕРАТ

Вірчак Сергій Анатолійович

«Нелінійна регресійна модель для оцінювання якості web-застосунків, що створюються з використанням фреймворку Spring, 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: 83 стор., 12 табл., 8 рис., 16 використаних джерел, 5 додатків.

Актуальність теми дослідження. В даний час існує кілька груп методів та моделей для вимірювання якості ПЗ та процесу його розробки. Проблема полягає в тому, що при побудові якісної моделі необхідно враховувати тип програмних додатків, платформи розробки, мови програмування. Через відсутність моделі оцінювання якості web-застосунків на основі Spring-фреймворку дане дослідження є актуальним.

Мета дослідження. Метою є підвищення достовірності оцінювання метрики RFC web-застосунків на основі Spring-фреймворку для визначення їх якості.

Об'єкт дослідження. Об'єктом дослідження є процес оцінювання якості web-застосунків на основі Spring-фреймворку.

Предмет дослідження. Предметом дослідження є регресійні моделі оцінювання якості web-застосунків на основі Spring-фреймворку, що використовують метрики RFC, CBO, WMC для прогнозування.

Методи дослідження. В даній роботі застосовуються методи математичної статистики та теорії ймовірностей, лінійного та нелінійного регресійного аналізу, математичного програмування.

Наукова новизна одержаних результатів. Удосконалено нелінійну регресійну модель оцінювання метрики RFC для визначення якості web-

застосунків на основі Spring-фреймворку з використанням нормалізуючого перетворення десятковий логарифм та вилучення викидів. Це дозволяє підвищити достовірність оцінювання метрики RFC web-застосунків на основі Spring-фреймворку в порівнянні з існуючою нелінійною регресійною моделлю для Java.

Практичне значення одержаних результатів. Розроблено програмне забезпечення для реалізації нелінійної регресійної моделі оцінювання метрики RFC та визначення якості web-застосунків на основі Spring-фреймворку.

Апробація результатів досліджень. Результати роботи пройшли апробацію на VII Всеукраїнській науково-практичній інтернет конференції студентів, аспірантів та молодих вчених «СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» (29 листопада 2024 р., м. Херсон, м. Хмельницький).

Публікації. Основні результати кваліфікаційної роботи опубліковано у 1 науковій праці – тезах VII Всеукраїнської науково-практичної інтернет конференції студентів, аспірантів та молодих вчених.

Ключові слова: оцінювання метрики RFC, якість web-застосунків, нелінійні регресійні моделі, нормалізуючі перетворення, Spring-фреймворк.

ABSTRACT

Virchak Serhii Anatoliiovych

«A nonlinear regression model for estimating the quality of web applications created using the Spring framework, and the development of a program for its implementation»

Qualification work for obtaining a master's degree in specialty 121 – "Software Engineering". Admiral Makarov National University of Shipbuilding. Mykolaiv, 2022.

Volume: 83 p., 12 tables, 8 figures, 16 references, 5 appendices.

Relevance of the topic. Currently, there are several groups of methods and models for measuring the quality of software and its development process. The problem is that when building a quality model, it is necessary to take into account the type of software applications, development platforms, and programming languages. Due to the lack of a model for assessing the quality of web applications based on the Spring framework, this study is relevant.

Purpose of the research. The goal is to increase the reliability of evaluating RFC metrics of Spring framework-based web applications to determine their quality.

Object of the research. The object of the study is the process of assessing the quality of web applications based on the Spring framework.

Subject of the research. The subject of the study is regression models for assessing the quality of web applications based on the Spring framework, which use RFC, CBO, WMC metrics for prediction.

Methods of the research. This work uses methods of mathematical statistics and probability theory, linear and nonlinear regression analysis, and mathematical programming.

Scientific novelty of the obtained results. The nonlinear regression model for evaluating RFC metrics for determining the quality of Spring framework based web applications has been improved using a normalizing logarithm transformation and outlier removal. This allows for increased reliability in evaluating RFC metrics for

Spring framework based web applications compared to the existing nonlinear regression model for Java.

Practical significance of the obtained results. Розроблено програмне забезпечення для реалізації моделі нелінійної регресії для оцінки метрик RFC і визначення якості веб-додатків на основі Spring framework.

Approbation of research results. The results of the work were tested at the VII All-Ukrainian Scientific and Practical Internet Conference of Students, Postgraduate Students and Young Scientists "MODERN INFORMATION SYSTEMS AND TECHNOLOGIES" (November 29, 2024, Kherson, Khmelnytskyi).

Publications. The main results of the qualification work were published in 1 scientific paper – theses of the VII All-Ukrainian Scientific and Practical Internet Conference of Students, Postgraduate Students and Young Scientists.

Keywords: RFC metric evaluation, web application quality, nonlinear regression models, normalizing transformations, Spring framework.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ВИКОРИСТАННЯ РЕГРЕСІЙНИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ.....	12
1.1 Аналіз використання моделей регресії для оцінювання якості програмного забезпечення.....	12
1.2 Обґрунтування необхідності проведення досліджень	16
1.3 Висновки до розділу 1	18
2 ПОБУДОВА МОДЕЛІ НЕЛІНІЙНОЇ РЕГРЕСІЇ ДЛЯ ОЦІНЮВАННЯ МЕТРИКИ RFC SPRING-ФРЕЙМВОРК WEB-ЗАСТОСУНКІВ ДЛЯ ВИЗНАЧЕННЯ ЇХ ЯКОСТІ	19
2.1 Аналіз даних для побудови моделі нелінійної регресії для оцінювання метрики RFC для Spring-фреймворк web-застосунків визначення їх якості	19
2.2 Розробка моделі нелінійної регресії для оцінювання метрики RFC як функції метрик СВО та WMC для Spring-фреймворк web-застосунків визначення їх якості	22
2.3 Висновки до розділу 2	27
3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ WEB-ЗАСТОСУНКІВ НА ОСНОВІ SPRING-ФРЕЙМВОРКУ	29
3.1 Постановка задачі на розробку проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку	29
3.2 Розробка ескізного проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку	30
3.3 Розробка технічного проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку	39
3.4 Розробка робочого проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку	41
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ WEB-ЗАСТОСУНКІВ НА ОСНОВІ SPRING-ФРЕЙМВОРКУ	48

4.1 Вступ	48
4.2 Розрахунок витрат на створення й експлуатацію програми для оцінювання якості web-застосунків на основі Spring-фреймворку	48
4.3 Розрахунок економічної ефективності розробки і впровадження ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку	52
4.4 Висновки до розділу 4	53
5 ОХОРОНА ПРАЦІ	54
5.1 Вступ	54
5.2 Аналіз небезпечних і шкідливих факторів, характерних при роботі з персональним комп'ютером	54
5.3 Розробка заходів по зменшенню дії небезпечних і шкідливих факторів	55
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	59
6.1 Вступ	59
6.2 Вирішення проблем охорони навколишнього середовища шляхом застосування геоінформаційних систем	60
ВИСНОВКИ	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТОК А – Технічне завдання на розробку програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку	67
ДОДАТОК Б – Програма та методика випробувань програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку	72
ДОДАТОК В – Інструкція користувача програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку	75
ДОДАТОК Г – Текст програми	78
ДОДАТОК Д – Опис програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку	82

ВСТУП

Актуальність теми дослідження. В даний час в індустрії розробки програмного забезпечення (ПЗ) багато уваги приділяється оцінюванню обсягу майбутніх робіт, можливого ризику, необхідних ресурсів, плану робіт, якому бажано слідувати, необхідних зусиль (вартості), якості. Отримані оцінки допомагають контролювати як процес розробки продукту, так і сам товар. Вимірювання процесу виконуються з метою його покращення, вимірювання продукту – для підвищення його якості.

В даний час існує кілька груп методів та моделей для оцінювання якості програмного забезпечення та процесу його розробки. Проте не існує універсальної методики такого оцінювання [1-2]. Дослідженням в галузі оцінювання якості ПЗ присвячено ряд сучасних робіт [1 - 13].

В об'єктно-орієнтованому програмуванні (ООП) для вимірювання якості програмних систем застосовуються, в тому числі, і метрики Chidamber and Kemerer (СК Metrics). До складу СК Metrics входить шість метрик, що призначені для вимірювання якості дизайну та коду, а саме: WMC, DIT, NOC, CBO, RFC, LCOM [3]. В свою чергу відомо, що не всі метрики є незалежними і що метрика RFC залежить від CBO та WMC [4 - 5].

Уже розроблені як лінійні так і нелінійні моделі регресії для оцінювання метрики RFC на основі метрик CBO та WMC [7-12]. Проблема полягає в тому, що при побудові якісної моделі необхідно враховувати тип програмних додатків, платформи розробки, мови програмування. А для web-застосунків на основі Spring-фреймворку такої моделі не існує. Тому побудова моделі регресії для оцінювання метрики RFC, використовуючи як фактори метрики CBO та WMC web-застосунків, розроблених з використанням Spring-фреймворку, для визначення їх якості є актуальною.

Мета та завдання дослідження. Метою є підвищення достовірності оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків на

основі Spring-фреймворку для визначення їх якості. Для досягнення мети були сформульовані наступні завдання:

– Аналіз регресійних методів та моделей для оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків, розроблених з використанням Spring-фреймворку, для визначення їх якості.

– Збір даних, що будуть використовуватися для побудови моделі нелінійної регресії для оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків, розроблених з використанням Spring-фреймворку.

– Перевірка зібраних багатовимірних даних на наявність викидів.

– Удосконалення моделі нелінійної регресії для оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків, розроблених з використанням Spring-фреймворку, для визначення їх якості.

– Розробка ПЗ для оцінювання, на основі удосконаленої моделі, метрики RFC, використовуючи як фактори, CBO та WMC, а також визначення якості web-застосунків, розроблених з використанням Spring-фреймворку.

Об'єкт дослідження. Об'єктом дослідження є процес оцінювання якості web-застосунків на основі Spring-фреймворку.

Предмет дослідження. Предметом дослідження є регресійні моделі оцінювання якості web-застосунків на основі Spring-фреймворку, що використовують метрики RFC, CBO, WMC для прогнозування.

Методи дослідження. В даній роботі застосовуються методи математичної статистики та теорії ймовірностей, лінійного та нелінійного регресійного аналізу, математичного програмування.

Наукова новизна одержаних результатів. Удосконалено нелінійну регресійну модель оцінювання метрики RFC для визначення якості web-застосунків на основі Spring-фреймворку з використанням нормалізуючого перетворення десятковий логарифм та вилучення викидів. Це дозволяє підвищити достовірність оцінювання метрики RFC web-застосунків на основі Spring-фреймворку в порівнянні з існуючою нелінійною регресійною моделлю для Java.

Практичне значення одержаних результатів. Розроблено програмне забезпечення для реалізації нелінійної регресійної моделі оцінювання метрики RFC та визначення якості web-застосунків на основі Spring-фреймворку.

Особистий внесок здобувача. Робота є самостійно виконаним дослідженням. У праці [13], що опублікована у співавторстві, здобувачу належить нелінійна регресійна модель для оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків на основі Spring-фреймворку, що розроблена з використанням нормалізуючого перетворення на основі десяткового логарифму та вилучення викидів.

Апробація результатів досліджень. Результати роботи пройшли апробацію на VII Всеукраїнській науково-практичній інтернет конференції студентів, аспірантів та молодих вчених «СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ І ТЕХНОЛОГІЇ» (29 листопада 2024 р., м. Херсон, м. Хмельницький).

Публікації. Основні результати кваліфікаційної роботи опубліковано у 1 науковій праці – тезах VII Всеукраїнської науково-практичної інтернет конференції студентів, аспірантів та молодих вчених.

1 АНАЛІЗ ВИКОРИСТАННЯ РЕГРЕСІЙНИХ МОДЕЛЕЙ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ І ОБГРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ

1.1 Аналіз використання моделей регресії для оцінювання якості програмного забезпечення

В сучасній індустрії розробки ПЗ значна увага приділяється оцінюванню обсягу майбутніх робіт, можливого ризику, необхідних ресурсів, плану робіт, якому бажано слідувати, необхідних зусиль (вартості), якості. Отримані оцінки допомагають контролювати як процес розробки продукту, так і сам товар. Вимірювання процесу виконуються з метою його покращення, вимірювання продукту – для підвищення його якості.

В даний час існує кілька груп методів та моделей для оцінювання якості ПЗ та процесу його розробки. Проте, незважаючи на зусилля кращих фахівців, не існує фундаментальної теорії та методології оцінювання та забезпечення якості програмних продуктів [1, 2]. З цієї причини індустрія оцінювання якості ПЗ потребує нових наукових досліджень. Проблемам оцінювання якості та підходам до їх вирішення присвячено ряд сучасних робіт [1 - 13].

В об'єктно-орієнтованому програмуванні (ООП) для вимірювання якості програмних систем застосовуються, в тому числі, і метрики Chidamber and Kemerer (СК Metrics). До складу СК Metrics входить шість метрик, що призначені для вимірювання якості дизайну та коду, а саме: WMC, DIT, NOC, CBO, RFC, LCOM. В свою чергу відомо про залежність метрики RFC від CBO та WMC [4 - 5].

Серед моделей, що використовуються для оцінювання якості ПЗ, можна виділити як лінійні так і нелінійні моделі регресії, що дозволяють оцінити метрики RFC через CBO та WMC для визначення якості ПЗ [7-12, 14].

В роботі [7] запропоноване лінійне рівняння регресії для оцінювання середньої величини RFC для проекту як функції від CBO та WMC:

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2,$$

де \hat{Y} – результат оцінювання RFC,

b_i ($i = \overline{0, k}$) – параметри лінійної регресії,

X_1 – значення метрики СВО,

X_2 – значення метрики WMC.

Однак для використання лінійної регресійної теорії повинні виконуватися певні умови, однією з яких є нормальний розподіл залишків. Ця умова на практиці справедлива досить рідко.

Лінійна регресійна модель для даного дослідження при двох факторах має вигляд [15]:

$$Y = \hat{Y} + \varepsilon = b_0 + b_1 X_1 + b_2 X_2 + \varepsilon,$$

де Y – значення метрики RFC із вибірки,

ε – залишки.

При нормально розподілених залишках довірчий інтервал лінійної регресії визначається із співвідношення [15]:

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

а інтервал прогнозування – із співвідношення [15]:

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

де $S_Y^2 = \frac{1}{v} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$,

$v = N - 2 - 1$,

N – кількість точок багатовимірних даних,

$t_{\alpha/2, v}$ – квантіль t -розподілу Стюдента з $\alpha/2$ рівнем значимості та v степенями вільності,

$(X^+)^T X^+$ – матриця 2×2

$$(X^+)^T X^+ = \begin{pmatrix} S_{X_1 X_1} & S_{X_1 X_2} \\ S_{X_1 X_2} & S_{X_2 X_2} \end{pmatrix},$$

де $S_{X_q X_r} = \sum_{i=1}^N [X_{qi} - \bar{X}_q][X_{ri} - \bar{X}_r]$, $q, r = \overline{1, 2}$.

Якщо залишки не мають нормального розподілу, то використовують нелінійну регресію [8-14]. Нелінійна регресія використовується для моделювання нелінійних явищ, оскільки вона пропонує більшу гнучкість, надійність, точність, ніж лінійна регресія. Використання нелінійної регресії дає можливість краще зрозуміти відношення між предикторами та зробити більш точні прогнози. Також відомо, що в багатьох випадках, навіть якщо виконані всі умови для побудови моделі лінійної регресії, розроблена нелінійна регресійна модель має кращу якість.

Незважаючи на існування регресійних моделей для оцінювання середньої величини RFC на рівні проекту як функції від CBO та WMC, в роботі планується розробити нелінійну регресійну модель для web-застосунків на основі Spring-фреймворку. Необхідність розробки обумовлена тим, що при побудові якісної моделі необхідно враховувати тип програмних застосунків, платформи розробки, мови програмування. Проте моделі для оцінювання метрики RFC на рівні проекту як функції від CBO та WMC для web-застосунків на основі Spring-фреймворку знайдено не було.

Отримана модель буде використовуватися для визначення якості відповідних застосунків за методикою, запропонованою в [8]. Згідно цієї методики, спочатку з допомогою моделі оцінюється значення метрики RFC застосунку залежно від CBO і WMC та визначаються межі інтервалів довіри та прогнозу для RFC. Якщо отримане значення RFC знаходиться всередині довірчого інтервалу, то якість застосунку середня, якщо між довірчим та

прогнозними інтервалами – якість висока, вище верхньої межі прогнозного інтервалу – якість низька.

Для побудови нелінійної регресійної моделі для оцінювання метрики RFC web-застосунків на основі Spring-фреймворку буде використано метод нормалізуючих перетворень та вилучення викидів з наступними основними етапами:

- 1) вибір нормалізуючого перетворення;
- 2) нормалізація багатовимірних даних;
- 3) перевірка нормалізованих даних на наявність викидів з використанням квадрату відстані Махаланобіса та вилучення викидів у разі їх наявності;
- 4) побудова моделі лінійної регресійної для даних без викидів;
- 5) перевірка залишків лінійної регресії на підпорядкування нормальному розподілу (якщо залишки не нормально розподілені, то вилучається застосунок з найбільшим по модулю залишком і повертаємося на 2);
- 6) побудова довірчих інтервалів та інтервалів прогнозування для лінійної регресії;
- 7) перехід до нелінійної регресії через зворотне перетворення;
- 8) перехід до довірчих інтервалів та інтервалів прогнозування нелінійної регресії через зворотне перетворення.
- 9) перевірка попадання значень метрики RFC для емпіричних даних в інтервал прогнозування нелінійної регресії (якщо існують значення RFC, що виходять за межі інтервалу, то вони вважаються викидами, відповідні застосунки вилучаються з набору емпіричних даних і повертаємося на 2).

Позначимо нормалізуюче перетворення випадкових величин Y, X_1, X_2 у нормалізовані дані Z_Y, Z_1, Z_2 через ψ . Тоді [15]

$$T = \psi(P), \quad (1.1)$$

де $P = \{Y, X_1, X_2\}^T$ – негаусівський випадковий вектор,

$T = \{Z_Y, Z_1, Z_2\}^T$ – гаусівський випадковий вектор,

$\psi = \{\psi_Y, \psi_1, \psi_2\}^T$ – нормалізуюче перетворення.

З (1.1) через зворотне перетворення ψ^{-1} отримаємо [15]:

$$P = \psi^{-1}(T). \quad (1.2)$$

Для нормалізованих величин Z_Y, Z_1, Z_2 модель лінійної регресії має вигляд [15]

$$Z_Y = \hat{Z}_Y + \varepsilon = b_0 + b_1 Z_1 + b_2 Z_2 + \varepsilon, \quad (1.3)$$

довірчий інтервал обчислюється за формулою [15]

$$\hat{Z}_Y \pm t_{\alpha, v} S_{Z_Y} \left\{ \frac{1}{N} + (z_X^+)^T [(z_X^+)^T z_X^+]^{-1} (z_X^+) \right\}^{1/2}, \quad (1.4)$$

а інтервал прогнозування представляється співвідношенням [15]

$$\hat{Z}_Y \pm t_{\alpha, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (z_X^+)^T [(z_X^+)^T z_X^+]^{-1} (z_X^+) \right\}^{1/2}, \quad (1.5)$$

де $S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N (Z_{Y_i} - \hat{Z}_{Y_i})^2$.

$$v = N - 2 - 1,$$

$(z_X^+)^T z_X^+ - 2 \times 2$ матриця

$$(z_X^+)^T z_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} \end{pmatrix},$$

де $S_{Z_q Z_r} = \sum_{i=1}^N [Z_{q_i} - \bar{Z}_q][Z_{r_i} - \bar{Z}_r]$, $q, r = \bar{1}, \bar{2}$.

Через зворотне перетворення з (1.3), (1.4), (1.5) отримуємо нелінійну регресійну модель, її інтервали довіри та прогнозу.

1.2 Обґрунтування необхідності проведення досліджень

В результаті проведеного аналізу сучасного стану проблеми оцінювання якості ПЗ та наявних інструментів для її оцінювання, впливає необхідність та

актуальність подальших досліджень у вказаній сфері. Проблемам оцінювання якості та підходам до їх вирішення присвячено ряд сучасних робіт [1 - 12].

У попередньому підрозділі було підкреслено, що в об'єктно-орієнтованому програмуванні (ООП) для вимірювання якості програмних систем застосовуються, в тому числі, і метрики Chidamber and Kemerer (СК Metrics): WMC, DIT, NOSC, CBO, RFC, LCOM. В свою чергу відомо про залежність метрики RFC від CBO та WMC [4 - 5].

Також у попередньому підрозділі було зазначено, що для оцінювання якості ПЗ використовуються лінійні так і нелінійні моделі регресії, що дозволяють оцінити метрики RFC через CBO та WMC для визначення якості ПЗ [7-12, 14]. Проте існує ряд припущень, при яких можливо використовувати лінійну регресійну теорію. Одним з них є нормальний розподіл залишків. Однак це припущення виконується тільки в окремих випадках. Якщо залишки не нормально розподілені, то використовують нелінійну регресію [8-14].

Використання нелінійної регресії дає можливість краще зрозуміти відносини між предикторами та зробити більш точні прогнози. Також відомо, що в більшості випадків, навіть якщо виконані всі умови для побудови лінійної регресійної моделі, побудована модель нелінійної регресії має кращу якість. Тому зазвичай будують нелінійні регресійні моделі, застосовуючи лінеаризуючі, нормалізуючі перетворення або метод простого перебору. На практиці досить вживаними є нормалізуючі перетворення, серед яких можна виділити перетворення десятковий логарифм, яке використовувалося при розробці ряду відомих регресійних моделей.

Незважаючи на існування нелінійних регресійних моделей для оцінювання середньої величини RFC на рівні проекту як функції від факторів CBO і WMC, в роботі планується розробити нелінійну регресійну модель для web-застосунків на основі Spring-фреймворку. Необхідність розробки обумовлена тим, що при побудові якісної моделі необхідно враховувати тип програмних застосунків, платформи розробки, мови програмування. Проте моделі нелінійної регресії для

оцінювання метрики RFC з CBO та WMC у якості незалежних змінних для web-застосунків на основі Spring-фреймворку знайдено не було.

Все вище перераховане обґрунтовує необхідність розробки моделі нелінійної регресії для оцінювання метрики RFC як функції від факторів CBO та WMC для Spring-фреймворк web-застосунків, використавши як нормалізуюче перетворення десятковий логарифм, для визначення якості застосунків.

1.3 Висновки до розділу 1

В даному розділі було виконано аналіз використання моделей регресії для оцінювання якості програмного забезпечення, наведено обґрунтування доцільності виконання досліджень за темою кваліфікаційної роботи та розробки моделі нелінійної регресії для оцінювання метрики RFC через CBO та WMC для Spring-фреймворк web-застосунків для визначення їх якості.

Також в розділі:

- Визначено, що для оцінювання метрики RFC як функції від факторів CBO та WMC існують моделі як лінійної, так і нелінійної регресії.
- Визначено, що моделі для оцінювання метрики RFC для web-застосунків на основі Spring-фреймворку не існує.
- Визначено, що моделі нелінійної регресії в багатьох випадках мають кращу якість, ніж лінійні. Тому зазвичай будують моделі нелінійної регресії.
- Визначено, що для побудови нелінійних регресійних моделей застосовують лінеарізуючі, нормалізуючі перетворення або метод простого перебору. Однак на практиці досить вживаними є нормалізуючі перетворення, серед яких можна виділити перетворення десятковий логарифм, яке використовувалося при розробці ряду відомих регресійних моделей.

Тому прийнято рішення розробляти модель нелінійної регресії, що дасть змогу оцінити метрику RFC як функцію від факторів CBO та WMC для Spring-фреймворк web-застосунків, для визначення якості застосунків. В якості нормалізуючого перетворення буде використовуватися десятковий логарифм.

2 ПОБУДОВА МОДЕЛІ НЕЛІНІЙНОЇ РЕГРЕСІЇ ДЛЯ ОЦІНЮВАННЯ МЕТРИКИ RFC SPRING-ФРЕЙМВОРК WEB-ЗАСТОСУНКІВ ДЛЯ ВИЗНАЧЕННЯ ЇХ ЯКОСТІ

2.1 Аналіз даних для побудови моделі нелінійної регресії для оцінювання метрики RFC для Spring-фреймворк web-застосунків для визначення їх якості

Для побудови моделі нелінійної регресії для оцінювання метрики RFC було зібрано дані з метрик 40 web-застосунків на основі Spring-фреймворку, які було розміщено на ресурсі GitHub (<https://github.com/>). В роботі зібрано дані з наступних метрик:

- RFC (Y),
- CBO (X_1),
- WMC (X_2).

Дані зібрані з використанням ck-metrics (<https://github.com/mauricioaniche/ck>) і представлені в таблиці 2.1.

Таблиця 2.1 – Емпіричні дані для розробки моделі нелінійної регресії для оцінювання метрики RFC для web-застосунків на основі Spring-фреймворку та квадрат відстані Махаланобіса (SMD) для емпіричних даних та нормалізованих емпіричних даних

№	Y (RFC)	X_1 (CBO)	X_2 (WMC)	SMD	SMD (норм)
1	2	3	4	6	6
1	6,4861	6,6042	3,7708	0,7760	0,8520
2	3,7447	3,3830	3,8298	3,6959	3,8734
3	7,3026	4,8750	5,2171	1,1560	0,9069
4	8,6131	6,6488	5,6190	0,7007	0,7895
5	0,9231	3,8846	1,2308	5,4477	16,9969
6	4,7872	6,5851	2,9574	1,5506	1,4578
7	7,7778	2,8889	6,4444	6,3589	8,7341
8	4,0000	7,3462	2,8846	3,6163	3,0497
9	8,8421	4,8816	7,4211	1,3759	1,0281
10	4,3333	3,9048	12,3333	4,4751	5,0863

Продовження табл. 2.1

1	2	3	4	6	6
11	5,2000	3,2000	3,0000	5,7204	8,5992
12	5,0000	3,7500	12,2500	3,7141	3,8530
13	9,7317	5,6585	9,4634	1,1169	1,0419
14	18,5604	9,3956	7,8462	17,9301	7,0436
15	7,7846	3,7231	8,7538	2,3683	2,6429
16	5,5600	7,1600	3,5200	1,8199	1,3834
17	8,2083	6,0833	3,9167	1,2796	1,7657
18	7,5135	5,4189	4,5135	0,9866	0,9945
19	9,0588	7,0588	12,9412	5,3133	4,2353
20	7,6000	5,4615	4,5385	0,9830	0,9976
21	9,4615	5,5299	7,7692	1,0490	0,8171
22	4,4915	6,0339	2,2373	1,5404	2,4675
23	7,3256	4,2558	6,3721	1,6871	1,4498
24	6,0621	5,4969	6,5342	0,0919	0,2103
25	11,0750	4,7583	17,3417	7,6332	3,5688
26	3,3171	8,0976	2,4634	6,7732	5,7468
27	6,0227	5,6591	7,6364	0,3806	0,8389
28	6,8750	7,2500	7,3500	2,0850	2,2170
29	4,3871	5,3226	5,3226	0,7623	0,9615
30	4,9630	7,0741	2,8307	2,1333	1,8901
31	6,6440	7,2745	3,2092	1,5957	1,7585
32	7,9038	6,8077	7,2692	0,7758	1,0783
33	4,3000	4,0500	12,8000	5,1141	5,6025
34	5,9545	3,9545	14,7273	5,6841	4,0365
35	8,2813	6,3125	7,4063	0,3368	0,6344
36	7,3200	7,3200	4,6800	1,3190	1,1255
37	12,1196	5,6304	14,1087	4,9399	2,7981
38	9,1750	7,6000	4,6000	1,9539	1,9042
39	5,2174	5,4720	2,6460	1,2814	2,0912
40	4,2963	3,4259	9,0741	2,4779	3,4716

В таблиці 2.1 наведено результати розрахунку квадрату відстані Махаланобіса (SMD) для емпіричних даних та нормалізованих емпіричних даних. SMD для кожної точки вхідних багатовимірних даних визначається за формулою [15, 16]

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

де X_i – i -а багатовимірна точка, $X_i = \{X_{1i}, X_{2i}, X_{3i}\}^T$,

N – кількість багатовимірних точок,

\bar{X} – середній вектор вибірки,

S_N – коваріаційна матриця:

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

Емпіричні дані не розподілені за нормальним законом. Це підтверджується значеннями квадрату відстані Махаланобіса для 4 проекту із таблиці 2.1, який більший за $\chi_{m,\alpha}^2$ (квантіль розподілу χ^2 з рівнем значимості α), $\alpha = 0,005$, $\chi_{m,\alpha}^2 = 12,8382$.

Також для підтвердження негаусівського розподілу вихідних даних розраховано оцінку багатовимірного ексцесу $\hat{\beta}_2$ [17] згідно співвідношення

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

Відомо, що для нормально розподілених даних β_2 розраховується за формулою $\beta_2 = m(m+2)$ ($m = 3$ для двохфакторної моделі). Якщо $\hat{\beta}_2 > \beta_2$, то дані мають негаусівський розподіл. Для вихідних даних отримано: $\hat{\beta}_2 = 16,4088$; $\beta_2 = 15$, що підтверджує негаусівський розподіл.

Так як вихідні дані не мають гаусівського розподілу, то будемо розробляти нелінійну регресійну модель. Першим кроком побудови нелінійної регресії була нормалізація емпіричних даних та вилучення викидів з використанням SMD. В якості нормалізуючого перетворення обрано десятковий логарифм. Тоді перетворення (1.1) $\psi = \{\psi_Y, \psi_1, \psi_2\}^T$ набуде вигляду $\psi = \{\lg Y, \lg X_1, \lg X_2\}^T$. Для вилучення викидів з нормалізованих даних будемо розраховувати SMD для кожної багатовимірної точки i ($i = \overline{1, N}$) за формулою [15]:

$$d_i^2 = (Z_i - \bar{Z})^T S_N^{-1} (Z_i - \bar{Z}), \quad (2.2)$$

де Z_i – i -а багатовимірних точка нормалізованих даних, яка має три координати в двохфакторній моделі

N – кількість багатовимірних точок,

\bar{Z} – середній вектор нормалізованих даних,

S_N – коваріаційна матриця:

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

Для кожного d_i^2 також розраховано статистику T_{S_i} [16]

$$T_{S_i} = (N - m)Nd_i^2 / ((N^2 - 1)m). \quad (2.3)$$

При підпорядкуванні даних закону багатовимірного нормального розподілу, d_i^2 (2.2) не повинен перевищувати $\chi_{m,\alpha}^2$ (квантіль розподілу χ^2 з рівнем значимості α) та T_{S_i} (2.3) не повинна перевищувати $F_{m,N-m,\alpha}$ (квантіль F -розподілу з рівнем значимості α). В даному дослідженні $\alpha = 0,005$. Ті проекти, для яких дані вимоги не виконуються, вважаються викидами.

На першій ітерації знайдено та вилучено 1 викид. Тобто залишилося 39 проектів. На наступній ітерації, після перерахунку d_i^2 та T_{S_i} , викиди відсутні.

2.2 Розробка моделі нелінійної регресії для оцінювання метрики RFC як функції метрик CBO та WMC для Spring-фреймворк web-застосунків для визначення їх якості

Емпіричні дані, які зібрані для побудови моделі нелінійної регресії для оцінювання метрики RFC як функції від метрик CBO та WMC, наведені в таблиці 2.1, мали негаусівський розподіл. Для побудови моделі нелінійної регресії,

використовуючи метод нормалізуючих перетворень, виконано нормалізацію даних, використовуючи в якості нормалізуючого перетворення десятковий логарифм, та вилучено 1 викид. Для подальшого дослідження залишився набір із 39 багатовимірних даних.

Наступним етапом створення моделі нелінійної регресії є розробка моделі лінійної регресії для нормалізованих даних без викидів, яка має вигляд (1.3). В (1.3), враховуючи обране нормалізуюче перетворення, $Z_Y = \lg Y$; $Z_1 = \lg X_1$; $Z_2 = \lg X_2$. Оцінки параметрів b_0 , b_1 , b_2 для (1.3) розраховані методом найменших квадратів та відповідно дорівнюють 0,1074, 0,5775, 0,3720. Модель лінійної регресії для оцінювання метрики RFC як функції від метрик СВО та WMC для нормалізованих даних без викидів має вигляд:

$$Z_Y = \hat{Z}_Y + \varepsilon = 0,1074 + 0,5775Z_1 + 0,3720Z_2 + \varepsilon. \quad (2.4)$$

Для обґрунтування використання лінійної регресійної теорії необхідно підтвердити гіпотезу H_0 про гаусівський розподіл випадкової величини (ВВ) ε із (2.4). Перевірка гіпотези виконана з використанням одного із критеріїв згоди, а саме критерію χ^2 Пірсона.

При використанні критерію Пірсона необхідно обчислити випадкову величину χ^2 із співвідношення [15]:

$$\chi^2 = \sum_{j=1}^m \frac{(n_j - np_j)^2}{np_j}, \quad (2.5)$$

де m – кількість підінтервалів розбиття інтервалу випадкової величини,

n_j – кількість значень ВВ, що потрапила в підінтервал j ,

n – всього значень ВВ,

p_j – ймовірність того, що ВВ попаде в підінтервал j .

Для визначення кількості підінтервалів m традиційно використовують одну із формул:

– Стерджеса: $m = 3,32 \lg n + 1$;

– Брукса і Карузера: $m = 5 \lg n$.

За формулою Стерджеса отримали 6 підінтервалів.

Ймовірність того, що ВВ попаде в підінтервал j обчислюється по формулі [15]:

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

де $f(x)$ – функція щільності ймовірності гаусівського розподілу,

x_{j-1} , x_j – ліва та права границі підінтервалу j .

Функції щільності ймовірності визначається за формулою [15]:

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

де σ_x – вибіркове середньоквадратичне відхилення,

m_x – математичне сподівання.

Згідно (2.5) отримали $\chi^2 = 3,9271$. На наступному кроці по таблиці χ^2 Пірсона знаходимо критичне значення $\chi_{кр}^2$, яке залежить від двох параметрів α та ν , де α – заданий рівень значимості (у нас 0,05, тобто з ймовірністю 0,95 можна гарантувати прийняття чи спростування гіпотези), ν – число степеней вільності. ν знаходиться по формулі $\nu = m - k - 1$, де k – число параметрів розподілу (в даному випадку $k = 2$).

Після знаходження $\chi_{кр}^2$, необхідно порівняти χ^2 та $\chi_{кр}^2$. Якщо χ^2 менше $\chi_{кр}^2$, то нульова гіпотеза приймається, в протилежному випадку – спростовується. Отримали $\chi^2 = 3,9271$, а $\chi_{кр}^2 = 7,8147$, тобто приймається нульова гіпотеза про

нормальний розподіл випадкової величин ε . Це теоретично обґрунтовує використання моделі лінійної регресії.

Інтервал довіри та інтервал прогнозу лінійної регресії (2.4) визначається за відомими співвідношеннями (1.4) та (1.5). Матриця, що використовується для розрахунку інтервалів, має вигляд:

$$S_z = \begin{pmatrix} 0,5933 & -0,3935 \\ -0,3935 & 2,2389 \end{pmatrix}.$$

На наступних етапах розробки моделі нелінійної регресії для оцінювання метрики RFC як функції від СВО та WMC для web-застосунків на основі Spring-фреймворку через зворотнє перетворення (1.2) здійснюємо перехід до моделі нелінійної регресії, інтервалів довіри та інтервалів прогнозу.

Модель нелінійної регресії наступна:

$$Y = 10^{\varepsilon+0,1074} X_1^{0,5775} X_2^{0,3720}. \quad (2.6)$$

На наступному етапі створення моделі нелінійної регресії для оцінювання метрики RFC як функції від СВО та WMC Spring-фреймворк web-застосунків необхідно виконати перевірку попадання значень метрики RFC для емпіричних даних в прогнозний інтервал нелінійної регресії. Під час перевірки не було знайдено значень RFC, що виходять за інтервал, тобто модель нелінійної регресії з її довірчими інтервалами та інтервалами прогнозування побудована.

Таким чином остаточна модель нелінійної регресії має вигляд (2.6). Для розрахунку границь довірчих інтервалів та інтервалів прогнозування лінійної регресії використовуються нормалізовані дані, квантиль розподілу Стьюдента=2,028094001, залишкове стандартне відхилення : 0,128409561 та обернена коваріаційна матриця: $\begin{pmatrix} 1,908072374 & 0,335364555 \\ 0,335364555 & 0,505590197 \end{pmatrix}$ (1.4, 1.5).

Інтервали нелінійної регресії отримуються через зворотнє перетворення.

Якість побудованої регресійної моделі перевірялася за наступними показниками: R^2 , $MMRE$ та $PRED(0,25)$ [15].

Коефіцієнт детермінації R^2 розраховано за формулою [15]

$$R^2 = 1 - \frac{SS_E}{SS_T}, \quad (2.7)$$

де SS_E – сума квадратів залишків;

SS_T – загальна сума квадратів.

R^2 характеризує частку варіації ознаки Y , яка врахована в моделі та обумовлена впливом на неї факторів. R^2 приймає значення від 0 до 1. Значення $R^2 \geq 0,5$ вважається прийнятним. При $R^2 \geq 0,75$ модель достатньо ефективна та результативна, при $R^2=1$ модель достовірна та адекватна.

Для обчислення середньої величини відносної похибки $MMRE$ та рівня прогнозування $PRED(l)$ використовується величина відносної похибки MRE_i [15]

$$MRE_i = \left| \frac{y_i - \hat{y}_i}{y_i} \right|.$$

Тоді $MMRE$ дорівнює [15]:

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i, \quad (2.8)$$

а $PRED(0,25)$ визначається з [15] як кількість тих MRE_i , які не перевищують 0,25.

Прийнятне значення $MMRE \leq 0,25$, а $PRED(0,25) - \geq 0,75$.

В нашому випадку згідно формул (2.7) - (2.9) отримано $R^2=0,4043$, $MMRE=0,2493$ та $PRED(0,25) = 0,6923$. Таким чином отримано, враховуючи прийнятні значення розрахованих показників якості, що задовільним є лише

MMRE. Для покращення якості моделі необхідно спробувати інше нормалізуюче перетворення.

Для порівняльного аналізу була обрана модель нелінійної регресії для оцінювання метрики RFC як функції від CBO та WMC Java застосунків з відкритим кодом виду [10]:

$$Y = 10^{\varepsilon - 0,0548} X_1^{0,6723} X_2^{0,4415}.$$

Застосувавши дану модель до даних нашого дослідження, отримали наступні показники якості моделі: $R^2 = 0,3458$, $MMRE = 0,2416$ та $PRED(0,25) = 0,5128$. Тобто по коефіцієнту детермінації R^2 та рівню прогнозування $PRED(0,25)$ модель нелінійної регресії для оцінювання метрики RFC як функції від CBO та WMC Spring-фреймворк web-застосунків є кращою, а по середній величині відносної похибки *MMRE* обидві моделі мають задовільні значення, проте кращою є модель для оцінювання метрики RFC Java застосунків з відкритим кодом [10].

Як висновок, можна зазначити про підтвердження залежності якості моделі від типу програмних додатків, платформ розробки, мов програмування, для яких модель була розроблена. Для покращення якості моделі необхідно спробувати застосувати більш складне нормалізуюче перетворення.

2.3 Висновки до розділу 2

У даному розділі виконано наступне:

- Зібрано 40 web-застосунків, що розроблені на основі Spring-фреймворку та розміщені на GitHub, з яких отримано значення метрик RFC, CBO та WMC.
- Перевірено вихідні дані на нормальний розподіл з використанням SMD та β_2 (багатовимірного ексцесу). Виявлено відсутність гаусівського розподілу.

– Виконано нормалізацію вихідних даних з допомогою нормалізуючого перетворення десятковий логарифм та вилучення викидів з використанням SMD. Вилучено 1 проект.

– Побудовано нелінійну регресійну модель для оцінювання метрики RFC як функції від CBO та WMC для web-застосунків на основі Spring-фреймворку для визначення якості застосунків.

– Побудовано інтервали довіри та прогнозу для нелінійної регресії.

– Розраховано показники якості створеної моделі нелінійної регресії для оцінювання метрики RFC як функції від CBO та WMC для web-застосунків на основі Spring-фреймворку з допомогою метрик R^2 , $MMRE$ та $PRED(0,25)$, значення яких дорівнюють: $R^2 = 0,4043$, $MMRE = 0,2493$ та $PRED(0,25) = 0,6923$.

– Застосовано нелінійну регресійну модель для оцінювання метрики RFC Java застосунків з відкритим кодом [10] до емпіричних даних даного дослідження та отримано наступні показники якості моделі $R^2 = 0,3458$, $MMRE = 0,2416$ та $PRED(0,25) = 0,5128$.

– Виконано порівняння якості побудованої моделі з моделлю нелінійної регресії для оцінювання метрики RFC Java застосунків з відкритим кодом [10]. Розроблена модель має кращі оцінки якості. Порівняльний аналіз показників якості моделей підтвердив доцільність побудови моделей під конкретну мову програмування, фреймворк, тип ПЗ. Побудова більш якісної моделі для оцінювання RFC для web-застосунків на основі Spring-фреймворку дасть можливість більш точно оцінити якість відповідних застосунків згідно методики [8].

3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ WEB-ЗАСТОСУНКІВ НА ОСНОВІ SPRING-ФРЕЙМВОРКУ

3.1 Постановка задачі на розробку проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Виходячи з викладеного вище, наведемо постановку задачі на створення ПЗ. В роботі повинно бути розроблене ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку, яке буде задовольняти вказаним вимогам:

Вимоги до функціональних характеристик

ПЗ повинно надавати наступні функціональні можливості:

- Введення у форму значення метрики CBO Spring-фреймворк web-застосунку для оцінювання значення його метрики RFC.
- Введення у форму значення метрики WMC Spring-фреймворк web-застосунку для оцінювання значення його метрики RFC.
- Введення у форму довірчої ймовірності.
- Оцінювання значення метрики RFC на основі введених значень метрик, використовуючи побудовану модель нелінійної регресійної.
- Обчислення меж (лівої та правої) інтервалу довіри для RFC.
- Обчислення меж (лівої та правої) інтервалу прогнозу для RFC.
- Визначення якості web-застосунку на основі Spring-фреймворку.

Вимоги до складу і параметрів технічних засобів

Необхідні технічні засоби з наступними мінімальними параметрами:

- Оперативна пам'ять – 1 Гб.
- Дисковий простір – 500 Мб.

Вимоги до інформаційної та програмної сумісності

Необхідна операційна система Windows 7 та вище.

Розгорнута постановка задачі сформульована в технічному завданні, що наведене в Додатку А.

3.2 Розробка ескізного проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Ескізний проект розробляється на основі документів передпроектної стадії: обстеження предметної галузі та технічного завдання.

Наведемо перелік робіт на етапі ескізного проектування: виявлення акторів, виявлення варіантів використання, побудова діаграми варіантів використання, розробка специфікацій прецедентів, розробка діаграм діяльності.

Розробка діаграми прецедентів

У даного ПЗ буде одна дійова особа, яку названо користувач. В таблиці 3.1 наведено опис акторів ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку.

Таблиця 3.1 – Виявлення акторів ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Виявлені актори	Опис акторів
Користувач	Особа, яка з допомогою ПЗ буде визначати якість web-застосунків на основі Spring-фреймворку

ПЗ, що розробляється, повинно надавати певні функціональні можливості, які в UML представляються варіантами використання (ВВ). Таблиці 3.2 містить виявлені прецеденти.

Таблиця 3.2 – Виявлення варіантів використання

Виявлені актори	Виявлені ВВ	Опис ВВ
1	2	3
Користувач	Введення даних для оцінювання	ВВ містить в собі 3 варіанти: «Введення значення довірчої ймовірності», «Введення значення метрики СВО Spring-фреймворк web-застосунку», «Введення значення метрики WMC Spring-фреймворк web-застосунку», призначенням яких є задання вхідних даних для виконання оцінювання
	Введення значення довірчої ймовірності	Цей ВВ слугує для введення в форму значення довірчої ймовірності, яка необхідна для обчислення меж інтервалу довіри та інтервалу прогнозу
	Введення значення метрики СВО web-застосунку	Цей ВВ слугує для введення в форму значення метрики СВО Spring-фреймворк web-застосунку
	Введення значення метрики WMC web-застосунку	Цей ВВ слугує для введення в форму значення метрики WMC Spring-фреймворк web-застосунку
	Оцінювання	Цей ВВ містить в собі 3 варіанти: «Оцінювання значення метрики RFC», «Визначення меж інтервалу довіри RFC», «Визначення меж інтервалу прогнозу RFC», «Визначення якості web-застосунку на основі Spring-фреймворку».

Продовження табл. 3.2

1	2	3
	Оцінювання значення метрики RFC web-застосунку	Цей ВВ слугує для оцінювання значення метрики RFC Spring-фреймворк web-застосунку
	Визначення меж інтервалу довіри RFC web-застосунку	Цей ВВ слугує для визначення меж інтервалу довіри RFC Spring-фреймворк web-застосунку
	Визначення меж інтервалу прогнозу RFC web-застосунку	Цей ВВ слугує призначений для визначення меж інтервалу прогнозу RFC Spring-фреймворк web-застосунку
	Визначення якості web-застосунку	Цей ВВ слугує для визначення якості Spring-фреймворк web-застосунку.

Після аналізу варіантів використання та акторів розроблено діаграму ВВ ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку (рис. 3.1). В цілому, діаграма варіантів використання є зручним інструментом для візуалізації функціональності системи і забезпечення чіткого розуміння вимог та взаємодії системи з її користувачами.

На діаграмі між акторами та варіантами використання зв'язки асоціації, а між варіантами використання – включення.

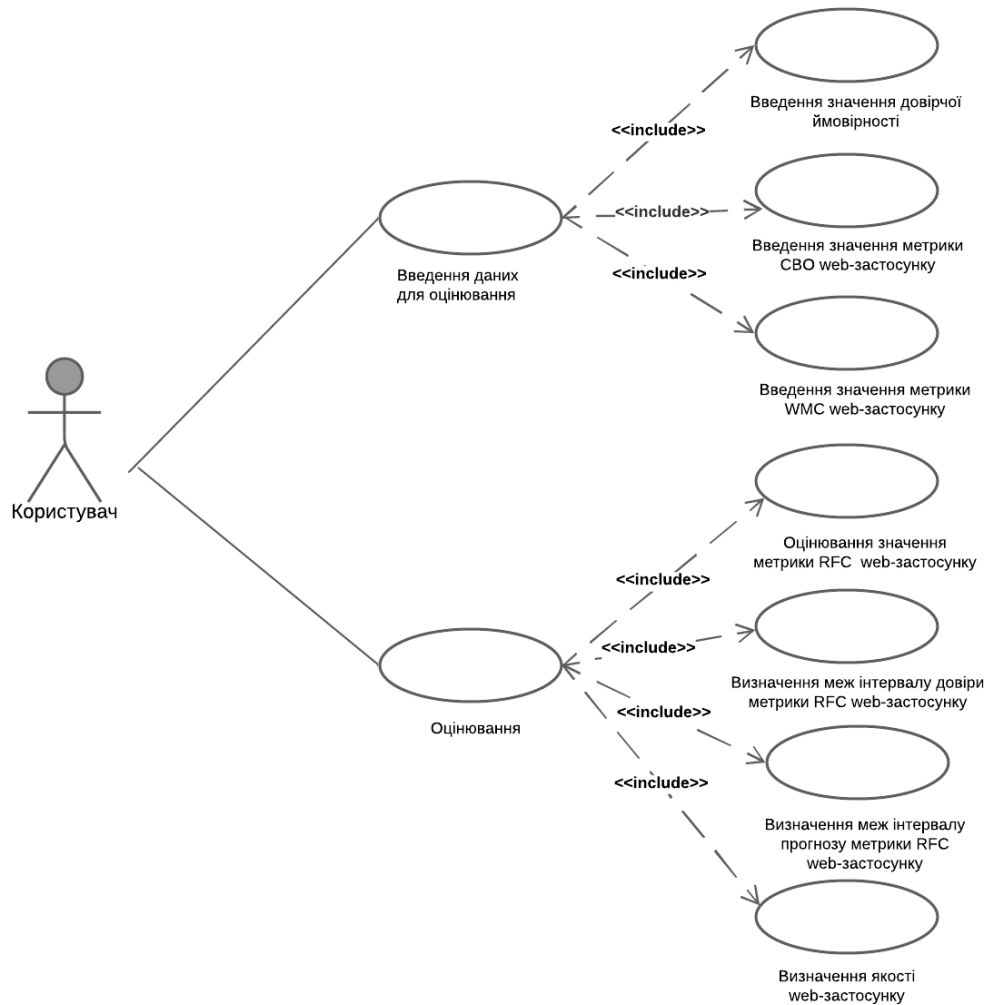


Рисунок 3.1 – Діаграма ВВ ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Розробка специфікацій для ВВ

Наступним кроком ескізного проектування є розробка специфікацій ВВ. Специфікація ВВ – це детальне описання кожного варіанту використання, який відображений на діаграмі ВВ. Вона розкриває, як саме користувач (актор) взаємодіє з системою для досягнення певної мети. Специфікація ВВ включає різні аспекти взаємодії, які допомагають чітко сформулювати функціональні вимоги до системи.

Специфікація ВВ «Введення даних для оцінювання»

1.0 Найменування варіанту використання: «Введення даних для оцінювання»

1.1 Короткий опис: ВВ активізується користувачем і надає йому можливість вносити вхідні дані для виконання оцінювання метрики RFC і визначення якості ПЗ.

2.0 Потоки подій

2.1 Основний потік: ВВ активізується при переході на форму для введення даних для оцінювання.

Цей варіант використання включає в себе 3 варіанти: «Введення значення довірчої ймовірності», «Введення значення метрики СВО Spring-фреймворк web-застосунку», «Введення значення метрики WMC Spring-фреймворк web-застосунку», призначенням яких є задання вхідних даних для виконання оцінювання.

2.2 Альтернативні потоки: відсутні.

3.0 Передумови: відсутні.

Специфікація ВВ «Введення значення довірчої ймовірності»

1.0 Найменування варіанту використання: «Введення значення довірчої ймовірності»

1.1 Короткий опис: ВВ активізується користувачем і надає йому можливість ввести довірчу ймовірність.

2.0 Потоки подій

2.1 Основний потік: ВВ активізується відразу коли користувач у формі для введення даних для оцінювання починає вводити значення довірчої ймовірності, яка необхідна для визначення меж інтервалу довіри та інтервалу прогнозу. Користувач вводить значення довірчої ймовірності.

2.2 Альтернативні потоки: відсутні.

3.0 Передумови: варіант використання «Введення даних для оцінювання»

Специфікація ВВ «Введення значення метрики СВО web-застосунку»

1.0 Найменування варіанту використання: «Введення значення метрики СВО web-застосунку»

1.1 Короткий опис: ВВ активізується користувачем. Цей варіант використання призначений для введення в форму значення метрики WMC Spring-фреймворк web-застосунку

2.0 Потоки подій

2.1 Основний потік: ВВ активізується відразу коли користувач у формі для введення даних для оцінювання починає вводити значення метрики СВО Spring-фреймворк web-застосунку. Користувач вводить значення метрики СВО.

2.2 Альтернативні потоки: відсутні.

3.0 Передумови: ВВ «Введення даних для оцінювання»

Специфікація ВВ «Введення значення метрики WMC web-застосунку»

1.0 Найменування варіанту використання: «Введення значення метрики WMC web-застосунку»

1.1 Короткий опис: ВВ активізується користувачем. Цей варіант використання призначений для введення в форму значення метрики WMC Spring-фреймворк web-застосунку

2.0 Потоки подій

2.1 Основний потік: ВВ активізується відразу коли користувач у формі для введення даних для оцінювання починає вводити значення метрики WMC Spring-фреймворк web-застосунку. Користувач вводить значення метрики WMC.

2.2 Альтернативні потоки: відсутні.

3.0 Передумови: ВВ «Введення даних для оцінювання»

Специфікація ВВ «Оцінювання»

1.0 Найменування ВВ: «Оцінювання»

1.1 Короткий опис: ВВ активізується користувачем і є загальним варіантом, який включає в себе оцінювання метрики RFC, розрахунок інтервалів довіри та прогнозу та визначення якості web-застосунку.

2.0 Потоки подій

2.1 Основний потік: ВВ активізується користувачем при натисканні на кнопку «Оцінити». Цей варіант використання включає в себе 4 варіанти: «Оцінювання значення метрики RFC web-застосунку», «Визначення меж

інтервалу довіри метрики RFC web-застосунку», «Визначення меж інтервалу прогнозу метрики RFC web-застосунку», «Визначення якості web-застосунку».

2.2 Альтернативні потоки: відсутні.

3.0 Передумови: відсутні.

Специфікація BB «Оцінювання значення метрики RFC web-застосунку»

1.0 Найменування варіанту використання: «Оцінювання значення метрики RFC web-застосунку».

1.1 Короткий опис: BB активізується користувачем та призначений для оцінювання значення метрики RFC Spring-фреймворк web-застосунку.

2.0 Потоки подій

2.1 Основний потік: BB активізується користувачем при запуску процесу оцінювання. Відбувається оцінювання значення метрики RFC Spring-фреймворк web-застосунку. Результат оцінювання виводиться на користувальницький екран.

2.2 Альтернативні потоки: не визначені

3.0 Передумови: BB «Оцінювання».

Специфікація BB «Визначення меж інтервалу довіри метрики RFC web-застосунку»

1.0 Найменування варіанту використання: «Визначення меж інтервалу довіри метрики RFC web-застосунку».

1.1 Короткий опис: BB активізується користувачем та призначений для визначення меж інтервалу довіри метрики RFC web-застосунку.

2.0 Потоки подій

2.1 Основний потік: BB активізується користувачем при запуску процесу оцінювання. Відбувається визначення меж інтервалу довіри метрики RFC Spring-фреймворк web-застосунку. Межі інтервалу довіри виводиться на користувальницький екран.

2.2 Альтернативні потоки: не визначені.

3.0 Передумови: BB «Оцінювання».

Специфікація ВВ «Визначення меж інтервалу прогнозу метрики RFC web-застосунку»

1.0 Найменування варіанту використання: «Визначення меж інтервалу прогнозу метрики RFC web-застосунку».

1.1 Короткий опис: ВВ активізується користувачем та призначений для визначення меж інтервалу прогнозу метрики RFC web-застосунку.

2.0 Потоки подій

2.1 Основний потік: ВВ активізується користувачем при запуску процесу оцінювання. Відбувається визначення меж інтервалу прогнозу метрики RFC Spring-фреймворк web-застосунку. Межі інтервалу прогнозу виводиться на користувальницький екран.

2.2 Альтернативні потоки: не визначені.

3.0 Передумови: ВВ «Оцінювання».

Специфікація ВВ «Визначення якості web-застосунку»

1.0 Найменування варіанту використання: «Визначення якості web-застосунку».

1.1 Короткий опис: ВВ активізується користувачем та призначений для оцінювання якості web-застосунку.

2.0 Потоки подій

2.1 Основний потік: ВВ активізується користувачем при запуску процесу оцінювання. Відбувається визначення якості web-застосунку. Якість web-застосунку виводиться на користувальницький екран.

2.2 Альтернативні потоки: не визначені.

3.0 Передумови: ВВ «Оцінювання».

Розробка діаграми діяльності ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

На рисунку 3.2 представлена діаграма діяльності ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку.

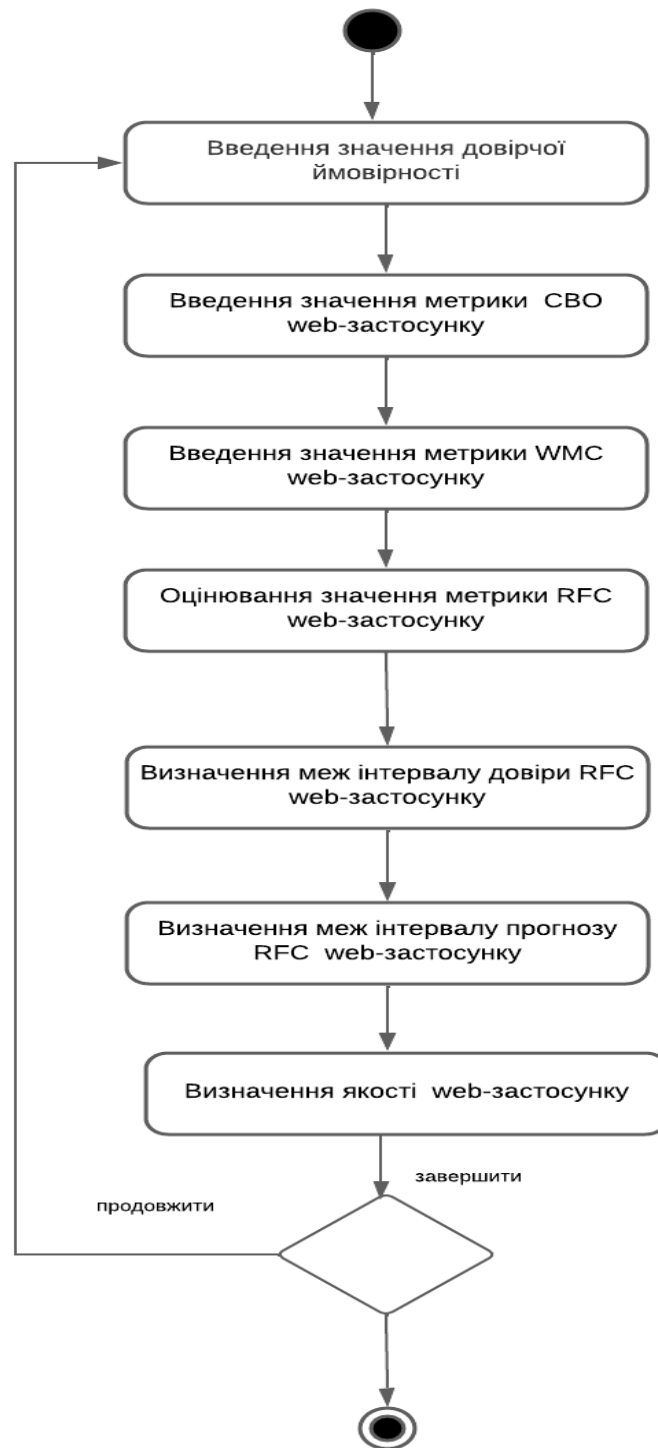


Рисунок 3.2 – Діаграма діяльності ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Діаграма діяльності (Activity Diagram) – це тип діаграми, що використовується в об'єктно-орієнтованому аналізі та проектуванні для моделювання бізнес-процесів, потоків робіт або різних сценаріїв виконання задач в рамках системи. Вона забезпечує чітке та наочне представлення дій та їх

послідовності, допомагаючи зрозуміти, як система виконує різні завдання. Вона є частиною UML (Unified Modeling Language) і описує процеси та їх взаємодії за допомогою графічних елементів.

3.3 Розробка технічного проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Розробка діаграми класів ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Діаграма класів (Class Diagram) – це одна із основних діаграм в UML (Unified Modeling Language). Вона є ключовим елементом в процесі проектування програмного забезпечення. Вона допомагає розробникам, аналітикам і зацікавленим сторонам краще зрозуміти, як побудовані взаємозв'язки між об'єктами та які класи складають систему. Ця діаграма є підґрунтям для виконання реалізації та тестування.

Діаграма класів ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку зображена на рисунку 3.3.

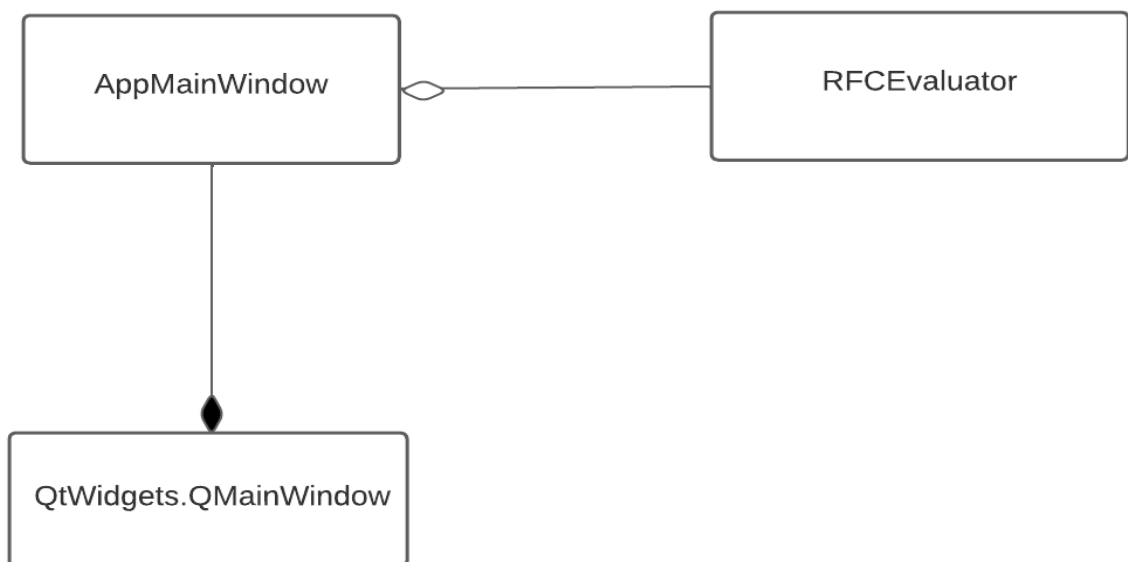


Рисунок 3.3 – Діаграма класів ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Призначення класів:

– Клас `AppMainWindow` – відповідає за введення даних для оцінювання метрики RFC, розрахунку інтервалу довіри та інтервалу прогнозу, якості web-застосунку на основі Spring-фреймворку

– Клас `QtWidgets` – клас, що є базовим відносно компонентів інтерфейсу користувача бібліотеки Qt.

– Клас `RFCEvaluator` – клас, що виконує оцінювання метрики RFC, інтервалу довіри, інтервалу прогнозу та якості web-застосунку на основі Spring-фреймворку

Динамічна модель ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

На рисунку 3.4 представлена діаграма послідовності для ВВ «Введення значення метрики CBO (Coupling between object classes) web-застосунку на основі Spring-фреймворку.

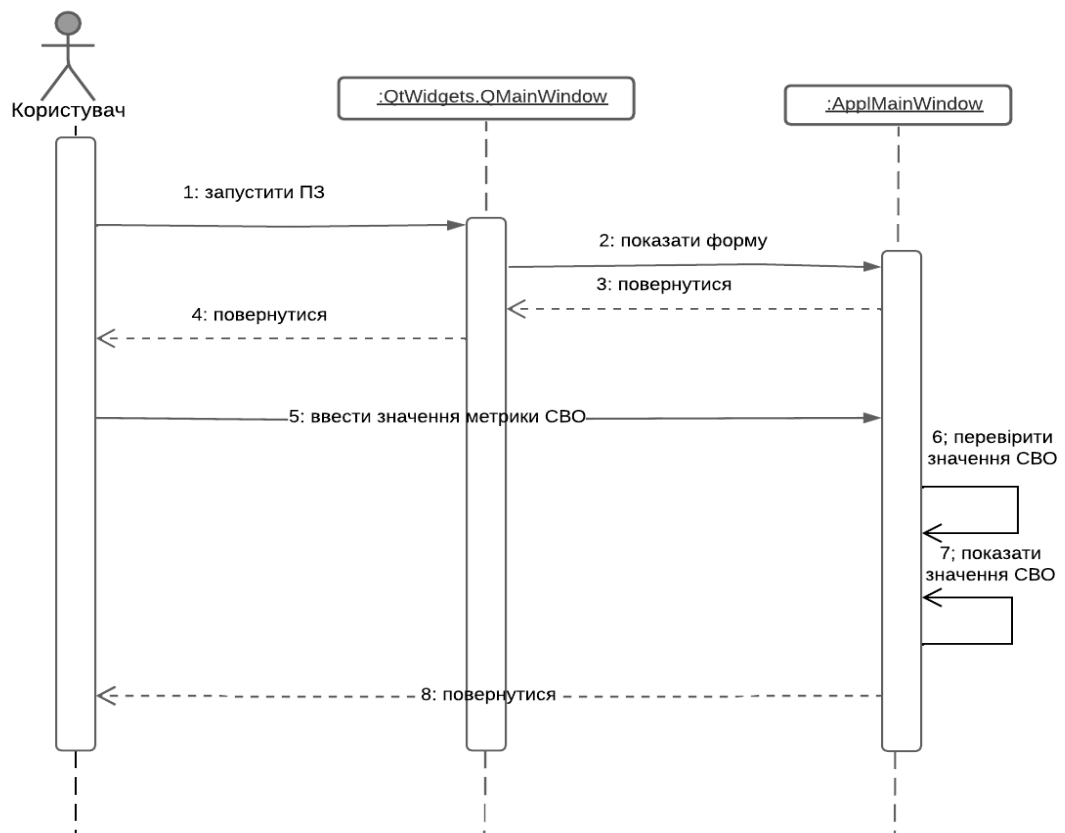


Рисунок 3.4 – Діаграма послідовності для ВВ «Введення значення метрики CBO web-застосунку на основі Spring-фреймворку

Для представлення динамічної моделі існують дві діаграми: діаграм послідовності та діаграма комунікацій (кооперацій). В даній роботі динамічна модель представлена в вигляді діаграми послідовності.

З допомогою діаграми послідовності описується взаємодія об'єктів між собою у часі з передачею повідомлень від одного об'єкта до іншого [18, 19].

3.4 Розробка робочого проекту ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Під час робочого проектування всі рішення, що приймалися на попередніх етапах, приймають форму, що доступна користувачеві. Вибираються засоби розробки ПЗ. Розробляється ПЗ для вирішення задачі [18, 19].

Під час робочого проектування також здійснюється тестування та випробування ПЗ.

Вибір засобів розробки ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

В якості мови для реалізації спроектованого програмного забезпечення обрано Python через низку обґрунтованих причин, що робить її популярною та зручною для розробки різноманітних програмних рішень. Ось основні аргументи для вибору Python:

- Простота та зрозумілість синтаксису

Python має простий та читабельний синтаксис, який дозволяє швидко розпочати програмування.

- Широка сфера застосування

Python використовується при веб-розробках, при аналізі даних, в штучному інтелекті та машинному навчанні, в наукових розрахунках.

- Багатий набір бібліотек та фреймворків

- Кросплатформеність

- Висока продуктивність при розробці

- Велика спільнота та підтримка
- Високий рівень абстракції
- Безкоштовність та відкритий код
- Сумісність з іншими мовами

Через ці переваги мова Python знаходиться серед самих популярних мов програмування на сьогоднішній день, і її вибір часто є обґрунтованим для багатьох проектів.

Реалізація та тестування ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Для перевірки відповідності ПЗ заданим вимогам та програмним документам було виконано тестування.

Тестування проводиться для виявлення помилок це процес виконання програми з метою виявлення помилок [20]. Результати досліджень показують, що найбільша увага при тестуванні програм приділяється проектуванню та створенню ефективних тестів. Це пов'язано з неможливістю «повного» тестування не тривіальної програми, тобто будь-який набір тестів для ПЗ не може гарантувати відсутності помилок.

Існує дві основні стратегії тестування: «чорної скриньки» (під управлінням вхідних і вихідних даних), «білої скриньки» (під управлінням логіки).

До методів «чорної скриньки» відносять: еквівалентного розбиття (базується на припущенні, що якщо виконується тест для одного із значень класу еквівалентності, то він аналогічно виконуватиметься і для всього класу); аналіз граничних значень (умови на границі); припущення про помилки (ґрунтується на інтуїції і досвіді типових помилок); створення функціональних діаграм (складаються діаграми, по яких вибираються класи еквівалентності, але вже не інтуїтивно, а залежно від логіки програми).

До методів «білої скриньки» відносять: покриття операторів; покриття рішень; покриття умов; покриття рішень/умов (кожна умова і рішення повинні

бути виконані і не виконані); комбінаторне покриття умов (потрібно перевірити всі комбінації умов).

В даній кваліфікаційній роботі тестування виконувалося методом еквівалентного розбиття.

Тестування функції «Введення значення метрики СВО web-застосунку» ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Класи еквівалентності функції «Введення значення метрики СВО web-застосунку» наведені в таблиці 3.3.

Таблиця 3.3 – Класи еквівалентності для функції «Введення значення метрики СВО web-застосунку»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
значення метрики СВО web-застосунку	1. дійсне додатне число $<=14$	2. дійсне додатне число >14 3. від'ємне число 4. не число 5. нічого не введено

Тести для класів еквівалентності функції «Введення значення метрики СВО web-застосунку» представлені в таблиці 3.4.

Таблиця 3.4 – Тести для класів еквівалентності функції «Введення значення метрики СВО web-застосунку»

№ п/п	Вхідні дані	Покритий клас	Результат
1	2	3	4
1	12,54	1	Можливість продовжити роботу

Продовження табл. 3.4

1	2	3	4
2	40	2	Повідомлення про помилкове значення
3	-23,63	3	Повідомлення про помилкове значення
4	текст	4	Повідомлення про помилкове значення
5	нічого не введено	5	Повідомлення не задане значення

Тестування функції «Введення значення метрики WMC web-застосунку» ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Класи еквівалентності для функції «Введення значення метрики WMC web-застосунку» представлені в таблиці 3.5.

Таблиця 3.5 – Класи еквівалентності для функції «Введення значення метрики WMC web-застосунку»

Вхідні умови	Правильні класи еквівалентності	Неправильні класи еквівалентності
значення метрики WMC web-застосунку	1. дійсне додатне число $<=30$	2. дійсне додатне число >30 3. від'ємне число 4. не число 5. нічого не введено

Тести для класів еквівалентності функції «Введення значення метрики WMC web-застосунку» представлені в таблиці 3.6.

Таблиця 3.6 – Тести для класів еквівалентності функції «Введення значення метрики WMC web-застосунку»

№ п/п	Вхідні дані	Покритий клас	Результат
1	34	1	Можливість продовжити роботу
2	57	2	Повідомлення про помилкове значення
3	-23,63	3	Повідомлення про помилкове значення
4	текст	4	Повідомлення про помилкове значення
5	нічого не введено	5	Повідомлення не задане значення

Випробування ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Випробування ПЗ проводиться на завершальному етапі та відноситься до найважливіших етапів ЖЦ. На цьому етапі необхідно перевірити та зафіксувати досягнуті показники якості ПЗ. Мета випробувань полягає в експериментальному визначенні фактичних характеристик розробленого ПЗ й визначення наскільки ПЗ відповідає технічному завданню [18, 19].

Програма та методика випробувань наведена в Додатку Б. В роботі випробовувалися наступні функції:

- Введення значення метрики CBO (Coupling between object classes) web-застосунку на основі Spring-фреймворку ПЗ для оцінювання якості.
- Введення значення метрики WMC (Weighted methods per class) web-застосунку на основі Spring-фреймворку ПЗ для оцінювання якості.
- Введення довірчої ймовірності.
- Оцінювання значення метрики RFC на основі значень метрик CBO та WMC з використанням побудованої нелінійної регресійної моделі.

– Визначення верхньої та нижньої меж довірчого інтервалу для значення RFC.

– Визначення верхньої та нижньої меж інтервалу прогнозування для значення RFC.

– Визначення якості web-застосунку на основі Spring-фреймворку згідно методики [8].

Під час випробувань завантажили розроблене ПЗ та ввели в форму всі необхідні дані.

Екранна форма з вхідними даними ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку представлена на рисунку 3.5.

Оцінювання якості Spring-фреймворк web-застосунків

Значення метрики CBO web-застосунку: 3,56

Значення метрики WMC web-застосунку: 7,25

Довірча ймовірність: 0.95

Оцінити

Результат оцінювання метрики RFC:

Якість:

Межі інтервалу довіри метрики RFC:

Межі інтервалу прогнозу метрики RFC:

Рисунок 3.5 – Екранна форма з вхідними даними ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Після натискання на кнопку «Оцінити» отримали оцінювання метрики RFC на основі значень метрик CBO та WMC з використанням побудованої моделі нелінійної регресії, верхню та нижню межі інтервалу прогнозу для значення RFC,

верхню та нижню межі інтервалу довіри для значення RFC, а також якість Spring-фреймворк web-застосунку згідно методики [8]. Екранна форма з результатами оцінювання представлена на рисунку 3.6.

Програма працює згідно програми та методики випробувань та згідно вимог, сформульованих у технічному завданні.

Field	Value
Значення метрики CBO web-застосунку:	3,56
Значення метрики WMC web-застосунку:	7,25
Довірча ймовірність:	0.95
Оцінити	
Результат оцінювання метрики RFC:	5.5712
Якість:	середня
Межі інтервалу довіри метрики RFC:	4.6841 - 6.6264
Межі інтервалу прогнозу метрики RFC:	2.9860 - 10.3949

Рисунок 3.6 – Екранна форма з результатами оцінювання ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ WEB-ЗАСТОСУНКІВ НА ОСНОВІ SPRING-ФРЕЙМВОРКУ

4.1 Вступ

Дана робота присвячена розробці ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку. Його можна використовувати для оцінювання якості web-застосунків на основі Spring-фреймворку.

Ефективність програмних систем визначається ступенем їх позитивного впливу на підприємство, що управляється, або процес в результаті використання засобів обчислювальної техніки, програмного забезпечення, зміни інформаційних потоків та організаційної структури управління.

Створення програмного забезпечення вимагає одноразових витрат на розробку, придбання необхідних технічних засобів і поточних витрат на функціонування продукту. Економія від функціонування ПЗ визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного забезпечення характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку оптовими цінами, тарифами і ставками заробітної плати.

4.2 Розрахунок витрат на створення й експлуатацію програми для оцінювання якості web-застосунків на основі Spring-фреймворку

Витрати на розробку системи складаються з витрат на:

- заробітну платню розробника;
- амортизацію ЕОМ, на якій виконується розробка;
- експлуатацію цієї ЕОМ;
- засоби розробки;

– матеріали і комплектуючі.

Виходячи з того, що основна заробітна платня розробників програмного забезпечення складає 20000 грн./міс., вартість сучасної ПЕОМ складає 40000 грн. і вартість кіловат-години електроенергії складає 4,32 грн., розрахуємо вартість розробки системи.

Розрахунок заробітної плати:

$$Z_{zn} = ZP_{роз} \cdot T_{роз} \quad (4.1)$$

де $ZP_{роз}$ – зарплатня розробника за місяць; $T_{роз}$ – тривалість розробки (дослідження, створення, налагодження і впровадження).

Для розробки даної системи необхідно 4 місяці (за експертною оцінкою часу на розробку аналогічних систем). Кількість розробників – 1 людина.

З цього випливає, що загальна сума витрат на заробітну платню складе:

$$Z_{zn} = ZP_{роз} \cdot T_{роз} = 20000 \cdot 4 = 80000 \text{ (грн.)}$$

Витрати на амортизацію ЕОМ, на якій виконується розробка, розраховується за формулою:

$$Z_{амор} = C_{варт} \cdot A \cdot T_{роз} \quad (4.2)$$

де $C_{варт} = 12000$ грн. – балансова вартість ЕОМ; $A = 60\%$ - амортизація за рік; термін служби ЕОМ – 5 років; $T_{роз} = 0,33$ року – час, необхідний для розробки системи.

$$Z_{амор} = 12000 \cdot 0,6 \cdot 0,33 = 2376 \text{ (грн.)}$$

Витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної енергії і розраховуються по формулі:

$$Z_{експ} = P_{ЕОМ} \cdot T_{розр} \cdot N_{р\delta} \cdot N_{рч} \cdot E_{ел} \quad (4.3)$$

де $P_{EOM} = 0,4$ квт./год. – потужність ЕОМ; $T_{розр} = 4$ місяці – тривалість розробки; $N_{рд} = 22$ дні – число робочих днів у місяці; $N_{рч} = 8$ годин – число годин у робочому дні; $E_{ел} = 4,32$ грн. – вартість 1 кВт/год електроенергії:
 $Z_{екст} = 0,4 \cdot 4 \cdot 22 \cdot 8 \cdot 4,32 = 152,06$ (грн.)

Витрати на матеріали і комплектуючі вироби враховують витрати на папір для друкувальних пристроїв, на картридж і тонер для принтера, а також на непередбачені витрати. Розрахунок приведений у таблиці 4.1 (ціни взяті відповідно до прайс-листів та договірних цін на даний вид матеріалів).

Таблиця 4.1 – Розрахунок витрат на матеріали і комплектуючі вироби

№ з/п	Найменування виробів	Вартість, грн.
1	Папір для принтера	200,00
2	Картридж та тонер для принтера	250,00
3	Література (доступ до Internet)	140,00
4	Непередбачені витрати	60,00
Разом		650,00

Загальний кошторис витрат на створення системи з урахуванням вищевказаного представлений у таблиці 4.2.

Таблиця 4.2 – Загальний кошторис витрат на створення системи

№ з/п	Найменування витрат	Вартість, грн.
1	Заробітна платня основна	80000,00
2	Заробітна плата додаткова (20% від п.1)	16000,00
3	Відрахування на соціальне страхування (38% від пп. 1 та 2)	36480,00
4	Витрати на амортизацію ЕОМ	2376,00
5	Витрати на експлуатацію ЕОМ	152,06
6	Витрати на матеріали і комплектуючі вироби	650,00
7	Адміністративні витрати (50% від основної заробітної платні)	40 000,00
Разом на створення системи		175658,06

До витрат на впровадження системи можна віднести витрати на придбання технічного забезпечення, вартість програмного забезпечення, вартість навчання кадрів, витрати на монтаж и настроювання мережі.

Оскільки параметри технічних засобів, які вже є, відповідають вимогам, то їх вартість при розрахунку витрат враховувати не будемо.

Виходячи з вимог до програмного забезпечення, а також проаналізувавши цінову політику, можемо прийняти наступне (таблиця 4.3).

Таблиця 4.3 – Перелік програмного забезпечення, необхідного для впровадження системи

№	Найменування ПО	Кількість	Вартість, грн.
1	MS Windows 10 Professional + Office 2019 Pro Plus	1	500,00
Разом		500,00	

Витрати на програмне забезпечення склали 500 грн.

Згідно з досвідом створення аналогічних програм, приймаємо, що вартість підготовки кадрів дорівнює 1000,00 грн. Витрати на супроводження програмного забезпечення дорівнюватимуть 1% від вартості програми и становлять 1756,58 грн.

Інвестиційні витрати на впровадження системи з урахуванням вартості навчання кадрів і витрати на супровід представлені в таблиці 4.4.

Таблиця 4.4 – Інвестиційні витрати на впровадження системи

№	Найменування інвестицій	Вартість, грн.
1	Вартість устаткування	0,00
2	Вартість програмного забезпечення	500,00
3	Вартість підготовки кадрів	1000,00
4	Супроводження системи	1756,58
Разом на впровадження системи		3256,59

Разом загальна сума витрат на створення і впровадження системи складає 178914,65 грн.

4.3 Розрахунок економічної ефективності розробки і впровадження ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

Основним показником економічної ефективності функціонування програмного забезпечення є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Економічна ефективність впровадження підсистеми оцінювання трудомісткості розбирання списку товарів в залежності від кількості ключів очікується за рахунок вивільнення робочого часу працівників та підвищення продуктивності праці. Крім того, не піддається прямій грошовій оцінці зменшення кількості помилкових та необережних рішень, підвищення оперативності керування, поліпшення організації роботи та своєчасне отримання необхідної інформації тощо.

Використання даної програми дозволяє вивільнити 0,5 робочого часу. Оскільки з системою працює один робітник, то система умовно вивільнить 0,5 робітника.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження системи вивільнить 0,5 робітника.

Зарплата 0,5 робітника у рік складає $\Delta C = 20000 \cdot 12 \cdot 0,5 = 120000$ (грн.). Річний економічний ефект розраховується за формулою 4.4:

$$E_{рік} = \Delta C - C_{супр} - E_n \cdot k, \quad (4.4)$$

де $C_{супр}$ – вартість супроводження системи; E_n – нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь – для обчислювальної техніки приймається = 0,5; k – додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи.

$$E_{\text{рік}} = 120000 - 1756,58 - 0,5 \cdot 178914,65 = 28786,01 \text{ (грн.)}$$

Строк окупності системи розраховується за формулою 4.5:

$$I = \frac{k}{\Delta C - C_{\text{супр}}}. \quad (4.5)$$

Отже

$$I = 1,51 \text{ (року).}$$

Тобто, система окупиться через 1,51 року, що приблизно дорівнює 18 місяцям.

4.4 Висновки за розділом 4

У цьому розділі були здійснені розрахунки витрат на створення й експлуатацію ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку, а також розрахунки економічної ефективності та строку окупності даного ПЗ. Виходячи з розрахунків, впровадження такого ПЗ окупить себе за 18 місяців. Отже, можна зробити висновок, що дана розробка програмного забезпечення економічно вигідна та обґрунтована.

5 ОХОРОНА ПРАЦІ

5.1 Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

5.2 Аналіз небезпечних і шкідливих факторів, характерних при роботі з персональним комп'ютером

При роботі на персональному комп'ютері співробітники фірми стикаються з дією таких небезпечних фізичних і психологічних чинників як: підвищена температура зовнішнього середовища, недостатня освітленість робочої зони, відсутність або нестача природного світла, електричний струм, статична електрика, розумове перенапруження, перенапруження очей, монотонність праці.

Мікрокліматичні параметри впливають на самопочуття і здоров'я людини, а також на надійність роботи засобів обчислювальної техніки. Для комфортної роботи приймається температура від 20 С до 25 С в теплу і холодну пору року відповідно, при відносній вологості 55%.

Також існують інші небезпечні шкідливі фактори, такі як:

– Небезпека ураження електричним струмом.

Все обладнання ЕОМ, як будь-яка електроустановка, представляють для людини велику потенційну небезпеку, оскільки в процесі експлуатації або проведення профілактичних робіт людина може торкнутися струмопровідних частини.

Експериментальні дослідження показали, що людина відчуває подразнюючу дію змінного струму промислової частоти силою 0,6-1,5 мА і постійного струму 5-7 мА. Ці струми не становлять серйозної небезпеки для діяльності організму людини, оскільки при такій силі струму можливо самостійне звільнення осіб від контакту з струмопровідними частинами. Для змінного струму промислової частоти сила не відпускає струму знаходиться в межах 6-20 мА. Постійний струм призводить до сильних больових відчуттів, сила такого струму 15-80 мА і більше.

– Недостатня освітленість робочого місця.

Правильно спроектоване і виконане висвітлення в приміщенні, де працюють оператори, забезпечує високу працездатність, надає позитивний психологічний вплив на тих, хто працює, сприяє підвищенню продуктивності роботи.

Рекомендована освітленість для роботи з екраном дисплея становить 400-750 лк. Рекомендовані яскравості в полі зору операторів повинні лежати в межах 1:5-1:10.

– Можливість виникнення пожежі.

Приміщення, в якому розміщені ПК за категоріями пожежної небезпеки відноситься до категорії "В". Зазвичай в ньому знаходиться велика кількість можливих джерел спалаху: кабельні лінії, що використовуються для живлення ПК від мережі змінного струму напругою 220 В, монітор, устаткування, меблі з горючих матеріалів, папір.

5.3 Розробка заходів по зменшенню дії небезпечних і шкідливих факторів

При розробці заходів, які забезпечують охорону праці та БЖ, враховують всі шкідливі і небезпечні фактори, зменшуючи кожен з них до допустимих значень. Тим не менше, тільки дотримання правил і норм технічної безпеки та

охорони праці забезпечить гарну робочу обстановку і запобіжить нещасні випадки на робочому місці.

При придбанні моніторів, слід вибирати монітори, що узгоджені з рекомендаціями щодо зменшення впливу на людину.

– Вимоги до мікроклімату в робочому приміщенні.

Для підтримки необхідних мікрокліматичних параметрів необхідно встановлювати обладнання для кондиціювання. Для зменшення кількості пилу встановлюють періодичність вологого прибирання в приміщенні.

– Вимоги до електробезпеки.

Велике значення для запобігання одержання травм від струму має правильна організація обслуговування діючих електроустановок, проведення ремонтних та профілактичних робіт, що здійснюється за допомогою наступних заходів: допуск до роботи, нагляд під час роботи, відключення під час ремонту, вивішування попереджувальних плакатів і знаків безпеки, перевірка відсутності напруги, накладення заземлення.

Для забезпечення електробезпеки обслуговуючого персоналу передбачені заземлювальні пристрої, до яких підключені всі металеві частини робочого обладнання.

Для зниження величин виникаючих статичних зарядів в обчислювальних центрах застосовують покриття технологічне з одношарового антистатичного лінолеуму марки ASN. Можна застосовувати загальне і місцеве зволоження повітря. Одним з нових методів зменшення статичної напруги в приміщенні є нейтралізація електрики іонізованим газом.

– Вимоги до освітленості.

В дисплейних залах, звичайно, застосовують одностороннє природне бічне освітлення. З метою уникнення прямого сонячного світла використовують приміщення з вікнами з північною, північно-східною або північно-західною орієнтацією. Монітори розташовують подалі від вікон і так, щоб вікна були збоку. Якщо екран монітора розташований до вікна, необхідні спеціальні пристрої.

Для штучного освітлення приміщень слід використовувати люмінесцентні лампи, оскільки в них висока світлова віддача (до 75 лм / Вт і більше), тривалий термін служби (до 10000 годин), мала яскравість поверхні, яка світиться, близький до природного спектральний склад випромінюваного світла, який забезпечує якісне перенесення кольорів. Найбільш прийнятними для дисплейних приміщень є люмінесцентні лампи ЛБ (білого світла) і ЛТБ (білого-тепло-білого світла) потужністю 40, 80 Вт.

Для запобігання засвічення екранів дисплеїв прямими світловими потоками світильники загального освітлення розташовують збоку від робочого місця, паралельно лінії зору оператора і стіні з вікнами. Таке розміщення світильників дозволяє проводити їх послідовне включення залежно від величини природної освітленості і виключає роздратування очей смугами світла і тіні, що чергуються, які виникають при поперечному розташуванні світильників.

– Вимоги до пожежної безпеки.

Для запобігання виникнення пожежі необхідно передбачити заходи пожежної профілактики: дотримання протипожежних вимог під час проектування і експлуатації систем вентиляції згідно зі ДБН 1.01.02-84; дотримання умов пожежної безпеки електроустановок згідно вимог; наявність засобів сповіщення:

- пожежні сповіщувачі;
- установки пожежогасінні;
- інструкції по заходам протипожежної безпеки, план евакуації людей і технічних засобів.

Для поліпшення умов пожежної безпеки в приміщенні має бути встановлена підлога з негорючих матеріалів; папір має зберігатися в металевій шафі; в наявності мають бути два вогнегасники, а також два димових датчика; має систематично проводитися прибирання та вентилявання приміщення.

1) Загальні вимоги з техніки безпеки

Кожен користувач ПК зобов'язаний:

- дотримуватися правило внутрішнього трудового розпорядку;

- палити лише у відведеному для цієї мети місці.

До основних небезпек, які можуть призвести до травм при роботі користувачем ПК, відносять небезпеку ураження електричним струмом та іонізуючим випромінюванням. Необхідно дотримуватися правил пожежної безпеки на робочому місці. У разі загоряння відключити електроживлення і зателефонувати в пожежну частину, пояснивши, що і де горить.

2) Вимоги до безпеки перед початком роботи.

Починаючи працювати на ПК, необхідно перевірити, чи добре прибране робоче місце, ознайомитися з можливими неполадками, та дізнатися про прийняті заходи по їх усуненню. Також перед початком роботи необхідно:

- отримати завдання у керівника роботи, а також інструктаж з техніки безпеки;
- підготувати необхідні інструменти, прилади. Перевірити їх працеспроможність, якщо необхідно, отримати захисні пристосування, запчастини, радіодеталі, матеріали;
- перевірити надійність кабелів в місцях їх підключення до джерел живлення;
- перевірити відсутність замикання між земляними ланками та ланками живлення напруги;
- перевірити наявність, справність і відповідність по струму запобіжників в блоках ремонтваної ПК або пристроїв.

3) Вимоги до безпеки під час завершення роботи:

- вимкніть електроживлення від ПК і обладнання, яке забезпечує роботу ПК;
- приведіть у порядок робоче місце;
- повідомте керівника робіт про закінчення роботи та її результати; зробіть запис у журналі про роботу ПК.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

В останні роки все частіше поставала проблема охорони навколишнього середовища. Поступово вона перетворилася на глобальну та з кожним днем набуває ще більшої актуальності. Причиною такого положення являються різноманітні антропогенні фактори. Це і демографічний вибух, і зростаюча швидкість урбанізації, і збільшення темпів та об'ємів виробництва та багато іншого. Великий вплив на навколишнє середовище здійснює людський фактор: викиди відходів, забруднення водоймищ та лісів, збільшення навантаження на орні землі тощо – усе це результати діяльності людини. Вже сьогодні у світі знищено більше 20 мільйонів гектарів орних земель. На атомних електростанціях накопичено тисячі тонн відпрацьованого ядерного палива, десятки тисяч кубометрів твердих і десятки мільйонів літрів рідких радіоактивних відходів. Понад 70 млн. м³ радіоактивних відходів зосереджено у відвалах та сховищах уранової, гірничодобувної та переробної промисловості України. Особливу небезпеку для здоров'я людини викликає забруднення води побутовими та промисловими стоками. Окрім забруднення отруйними речовинами та важкими металами така вода містить збудники різноманітних захворювань та зовсім непридатна для постачання населенню.

Внаслідок цього були розроблені міри для збереження навколишнього середовища, які включають в себе комплекс заходів з охорони атмосфери, водних ресурсів, флори, фауни та геологічного середовища. Таким чином, охорона навколишнього середовища – це комплексні заходи з раціонального використання природних ресурсів, їх збереження та примноження, а також забезпечення екологічної безпеки. Основної їх метою є зведення до мінімуму шкідливого впливу людини на оточуючий світ.

6.2 Вирішення проблем охорони навколишнього середовища шляхом застосування геоінформаційних систем

В наш час у всьому світі приділяється підвищена увага до проблем навколишнього середовища. Стрімкий розвиток господарської діяльності створив усі передумови екологічної кризи. У зв'язку з цим набуває важливості напрямок, пов'язаний з кількісною оцінкою антропогенного впливу на навколишнє середовище, створенням систем комплексної оцінки екологічної обстановки, а також моделюванням та прогнозуванням розвитку ситуації.

Однією з найскладніших та, зазвичай, найдорожчих справ на початку будь-якої серйозної роботи чи дослідження є збір якомога більшої кількості інформації про досліджуваний об'єкт, процес чи явище. Оцінка складних природних об'єктів неможлива без багатостороннього аналізу впливу різноманітних факторів. При такому комплексному підході доводиться спиратися на узагальнюючі характеристики навколишнього середовища, внаслідок чого об'єми навіть мінімально достатньої інформації повинні бути досить великими. У протилежному випадку буде складно досягти достатньої обґрунтованості прийнятих рішень. Однак, простого накопичення даних теж недостатньо. Ці дані повинні бути легкодоступними та систематизованими у відповідності до потреб. Дуже корисною виявляється можливість пов'язати різноманітні дані, порівняти, проаналізувати та візуалізувати їх у зручному вигляді. Хоча групування, співставлення, аналіз та інтерпретація даних залежать у першу чергу від кваліфікації дослідника, на етапі обробки даних важливе місце посідає технічна оснащеність, оскільки створення систем з такими задачами в наш час неможливе без використання сучасних комп'ютерних інструментів. В якості таких інструментів все частіше використовується потужна технологія географічних інформаційних систем (ГІС).

ГІС використовуються для створення карт основних параметрів навколишнього середовища. При отриманні нових даних ці карти використовуються для оцінки масштабів та темпів деградації флори та фауни. З

допомогою них можна проводити моніторинг широкомасштабних антропогенних впливів. Дані про антропогенне навантаження доцільно співставити з областями, що представляють особливий інтерес з точки зору охорони навколишнього середовища, наприклад, парками, заповідниками, заказниками.

З допомогою ГІС зручно моделювати вплив та розповсюдження забруднення від точкових та просторових джерел на місцевості, в атмосфері та по гідрологічній мережі. Результати окремих розрахунків можна накласти на природні карти чи план житлових масивів у даній місцевості, в результаті чого оперативно оцінити найближчі та більш віддалені наслідки надзвичайних ситуацій та вплив постійно діючих джерел забруднення.

ГІС широко використовуються при складанні та веденні різноманітних, у тому числі земельних, кадастрів. З їх допомогою зручно створювати бази даних та карти земельної власності, об'єднувати їх з базами даних по будь-яким соціально-економічним показникам, створювати комплексні карти, візуалізувати інформацію у вигляді різного роду діаграм, графіків тощо.

Ще одна розповсюджена сфера застосування ГІС – збір та управління даними по природоохоронним територіям, таким як заповідники, заказники, національні парки тощо. У межах таких районів можна проводити повноцінний просторовий моніторинг рослинності, популяцій рідкісних видів тварин, визначати вплив антропогенних втручань, таких як туризм чи прокладання доріг, та реалізувати відповідні природоохоронні заходи. Можливе і вирішення многокористувацьких задач, таких як регулювання випасу скота та прогнозування продуктивності земельних угідь. Такі задачі ГІС вирішують на науковій основі, тобто, вибираються рішення, що забезпечують мінімальний рівень впливу на дику природу, збереження рівня чистоти повітря, води та ґрунту на необхідному рівні, особливо у часто відвідуваних туристами районах.

ГІС являються ефективним засобом для вивчення середовища існування в цілому чи окремих видів рослинного та тваринного світу у просторових та часових аспектах. Якщо задані конкретні параметри оточуючого середовища, що

необхідні для існування, наприклад, якого-небудь виду тварин, включаючи наявність пасовищ та місць для розмноження, відповідні типи та запаси кормових ресурсів, джерела води, вимоги до чистоти природного середовища, то ГІС допоможе швидко знайти райони, у межах яких умови існування чи відновлення чисельності виду будуть близькими до оптимальних. На стадії адаптації переселеного виду до нової місцевості ГІС ефективні для моніторингу найближчих та віддалених наслідків вжитих заходів, оцінки їх успішності, виявлення проблем та пошуку шляхів до їх вирішення.

З розширенням та поглибленням природоохоронних заходів однією з основних сфер застосування ГІС є стеження за наслідками вживаних заходів на локальному та регіональному рівнях. Джерелом оновлюваної інформації можуть бути результати наземних зйомок та замірів, чи дистанційних спостережень з повітряного транспорту чи з космосу. Використання ГІС ефективно для моніторингу умов життєдіяльності місцевих та привнесених видів, виявлення причинно-наслідкових послідовностей та взаємозв'язків, оцінки сприятливих та несприятливих наслідків вживаних природоохоронних заходів на екосистему в цілому та окремі її компоненти, прийняття оперативних рішень по їх корегуванню в залежності від зміни зовнішніх умов

ВИСНОВКИ

Мета, яка ставилася в кваліфікаційній роботі і полягала в підвищенні достовірності оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків на основі Spring-фреймворку для визначення їх якості, повністю досягнута.

Для досягнення мети були вирішені наступні завдання:

– Виконано аналіз методів та моделей регресії для оцінювання метрики RFC на основі метрик CBO та WMC web-застосунків, розроблених з використанням Spring-фреймворку, для визначення їх якості.

– Виконано збір даних, що були використані для розробки моделі нелінійної регресії оцінювання метрики RFC на основі метрик CBO та WMC Spring-фреймворк web-застосунків.

– Перевірено зібрані багатовимірні дані на наявність викидів.

– Удосконалено модель нелінійної регресії для оцінювання метрики RFC на основі метрик CBO та WMC Spring-фреймворк web-застосунків, для визначення їх якості.

– Розроблено ПЗ для оцінювання якості web-застосунків, створених з використанням Spring-фреймворку.

– Розроблено розділи з економіки, охорони праці, охорони навколишнього середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chen C. Exploring the Dependency Relationships between Software Qualities / C. Chen, M. Shoga, B. Boehm // Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 2019, pp. 105-108.
2. Говорущенко Т. О. Аналіз галузі оцінювання якості програмного забезпечення / Т.О. Говорущенко // Вісник Національного університету "Львівська політехніка". Комп'ютерні системи та мережі. – Л.: Вид-во Нац. ун-ту "Львів. політехніка", 2013. – № 773. – С. 41-48.
3. Chidamber S.R. Towards a Metrics Suite for Object Oriented Design / S.R. Chidamber, C.F. Kemerer // Proc. of the 6th ACM Conference on Object Oriented Programming, Systems, Languages and Applications (OOPSLA), 1991, Phoenix, AZ, pp. 197-211.
4. Basili V.R. A Validation of Object-Oriented Design Metrics as Quality Indicators / V.R. Basili, L.C. Briand, W.L. Melo // IEEE Transactions on Software Engineering, vol. 22, no. 10, October 1996, pp. 751-761.
5. Chidamber S.R. A Metrics Suite for Object-Oriented Design / S.R. Chidamber, C.F. Kemerer // IEEE Transactions on Software Engineering, vol. 20, no. 6, June 1994, pp. 476-493.
6. Поморова О.В. Сучасні проблеми оцінювання якості програмного забезпечення / О.В. Поморова, Т.О. Говорущенко // Радіоелектронні і комп'ютерні системи. – Харків: НАУ "ХАІ", 2013. – № 5. – С.319–327.
7. Laing V. Principal Components of Orthogonal Object-Oriented Metrics / V. Laing, C. Coleman // White Paper Analyzing Results of NASA Object-Oriented Data (Task 323-08-14). – Greenbelt, Maryland (US): NASA Goddard Space Flight Center, the Software Assurance Technology Center (SATC), 2001.

8. Prykhodko S. A Technique for Detecting Software Quality Based on the Confidence and Prediction Intervals of Nonlinear Regression for RFC Metric / S. Prykhodko, N. Prykhodko // Proceedings of the 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022. – pp. 499-502.
9. Prykhodko S. A modified technique for constructing nonlinear regression models based on the multivariate normalizing transformations / S. Prykhodko, N. Prykhodko // Selected Papers of the VIII International Scientific Conference “Information Technology and Implementation” (IT&I-2021). Workshop Proceedings (IT&I-WS-2021), Kyiv, Ukraine, December 1-3, 2021. CEUR Workshop Proceedings, 3179, 156-166.
10. Приходько С.Б. Нелінійна регресійна модель для оцінювання метрики RFC застосунків з відкритим кодом на Java / С.Б. Приходько, Р.О. Позняков // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2022): Матеріали III-ої Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2022 р.). – Миколаїв: НУК імені адмірала Макарова, 2022. – С. 8-9.
11. Приходько С.Б. НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ ОЦІНЮВАННЯ МЕТРИКИ RFC ЗАСТОСУНКІВ З ВІДКРИТИМ КОДОМ НА KOTLIN / С.Б.Приходько, А.В. Кольцов, Є.В. Грабовський // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2023): Матеріали IV-ої Всеукраїнської науково-практичної інтернет конференції (30-31 жовтня 2023 р.). – Миколаїв: НУК імені адмірала Макарова, 2023. – С. 20-21.
12. Приходько С.Б. Нелінійна регресійна модель для оцінювання метрики RFC Застосунків для онлайн-покупок з відкритим кодом на Java / С.Б. Приходько, Г.О. Скакунов // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2024): Матеріали V-ої Всеукраїнської науково-практичної інтернет конференції (30-31 жовтня 2024 р.). – Миколаїв: НУК імені адмірала Макарова, 2024. – С. 59-60.
13. Каіров В.О. Математичне моделювання в задачі оцінювання метрики RFC web-застосунків на основі Spring-фреймворку / В.О. Каіров, І.І. Надточій, С.А. Вірчак // Матеріали VII Всеукраїнської науково-практичної інтернет-

конференції студентів, аспірантів та молодих вчених за тематикою «Сучасні комп'ютерні системи та мережі в управлінні»: збірка наукових праць (29 листопада 2024 року, м. Херсон, м. Хмельницький) / за ред. А.А. Григорової. – Херсон: Видавництво ФОП Вишемирський В. С., 2024. – С.175

14. Härdle W.K. Regression Models / W. K. Härdle, L. Simar // In: Applied Multivariate Statistical Analysis. – Springer, Berlin, Heidelberg, 2015.

15. 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.

16. Латанська Л.О. Нелінійна регресійна модель для оцінювання розміру веб-застосунків, що створюються з використанням PHP фреймворку Symfony / Л.О. Латанська, Л.М. Макарова, А.В. Кольцов, Д.Х. Давлатова // «Вісник Хмельницького національного університету. Серія: Технічні науки». – Хмельницький: ХНУ, 2022. – № 6 (315). – С. 119-124.

17. Mardia K.V. Measures of multivariate skewness and kurtosis with applications. *Biometrika*. 1970. Vol. 57, Issue 3. P. 519–530.

18. Моделювання програмного забезпечення: навч.-метод. посіб. / С.Ю. Манаков, О.Г. Трофименко, Ю.Г. Лобода, А.І. Дика. – Одеса: Фенікс, 2023. – 145 с.

19. Проектування інформаційних систем: Комп'ютерний практикум: навч. посібник / Укладачі: Л. М. Добровська, О.В. Аверьянова. Київ: КПІ ім. Ігоря Сікорського, 2021. – 202 с. URL: https://ela.kpi.ua/bitstream/123456789/41778/1/Proiektuvannia-informatsiinykhsystem_KompPrakt.pdf (дата звернення: 18.10.2024).

20. Авраменко А.С. Тестування програмного забезпечення. Навчальний посібник / А.С. Авраменко, В.С. Авраменко, Г.В. Косенюк. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с.

ДОДАТОК А

Технічне завдання на розробку програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку

Вступ

Назва – ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

1 Підстави для розробки

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

2 Призначення розробки

2.1 Функціональне призначення

Функціональним призначенням програми є надання користувачу можливості визначити якість web-застосунків на основі Spring-фреймворку.

2.2 Експлуатаційне призначення

Програма призначена для експлуатації на персональних комп'ютерах користувачів для оцінювання якості web-застосунків на основі Spring-фреймворку.

3 Вимоги до програми або програмного продукту

3.1 Вимоги для функціональних характеристик

3.1.1 Вимоги до переліку виконуваних функцій

Розроблена програма повинна забезпечувати виконання наступних функцій:

- Введення у форму значення метрики CBO (Coupling between object classes) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).
- Введення у форму значення метрики WMC (Weighted methods per class) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).
- Введення у форму довірчої ймовірності.
- Оцінювання значення метрики RFC на основі введених значень метрик CBO та WMC з використанням побудованої нелінійної регресійної моделі.
- Визначення верхньої та нижньої меж довірчого інтервалу для значення RFC.
- Визначення верхньої та нижньої меж інтервалу прогнозування для значення RFC.
- Визначення якості web-застосунку на основі Spring-фреймворку.

3.1.2 Вимоги для організації вхідних та вихідних даних

Внесення вхідних даних виконується шляхом їх введення у спеціальних формах. Вхідними даними є значення метрик CBO та WMC, довірна ймовірність.

Результуючі дані (оцінене значення метрики RFC, верхня та нижня межі довірчого інтервалу, верхня та нижня межі інтервалу прогнозування) повинні виводитися в відповідних місцях на створеній програмою формі.

3.2 Вимоги до надійності

Надійне (стійке) функціонування програми має бути забезпечено надійним (стійким) функціонуванням операційної системи.

Відмова програми можлива внаслідок некоректних дій користувача при взаємодії з операційною системою. Щоб уникнути виникнення відмов програми за вказаною вище причиною необхідно забезпечити стабільне функціонування операційної системи.

3.3 Вимоги до складу і параметрів технічних засобів

Для роботи ПЗ необхідні технічні засоби з наступними мінімальними параметрами:

- Оперативна пам'ять – 1 Гб.
- Дисковий простір – 500 Мб.

3.4 Умови експлуатації

Кліматичні умови експлуатації програмного продукту відповідають умовам експлуатації апаратної частини персонального комп'ютера.

3.5 Вимоги до інформаційної та програмної сумісності

Для роботи ПЗ необхідна операційна система Windows 7 та вище..

3.6 Вимоги до захисту інформації та програм

Вимоги до захисту інформації та програм відсутні.

3.7 Вимоги до маркування та упаковки

Вимоги до маркування відсутні. Вимоги до упаковки відсутні.

3.8 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання програми відсутні.

4 Вимоги до програмної документації

Склад програмної документації:

- технічне завдання;
- текст програми;
- опис програми;
- програма та методика випробувань;
- інструкція користувача.

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 – 18 місяців.

6 Стадії та етапи розробки програмного забезпечення

Стадії та етапи виконання розробки програмного забезпечення можна представити наступним чином (табл. А.1):

Таблиця А.1 – Стадії та етапи розробки

Стадії розробки	Етапи робіт	Контрольні точки
Технічне завдання	Обумовлення необхідності розробки	З 07.09.2024 до 10.09.2024
	Розробка та затвердження технічного завдання	З 11.09.2024 до 15.09.2024
Ескізний проект	Розробка ескізного проекту	З 16.09.2024 до 10.10.2024
	Затвердження ескізного проекту	З 11.10.2024 до 14.10.2024
Технічний проект	Розробка технічного проекту	З 15.10.2024 до 05.11.2024
	Затвердження технічного проекту	З 06.11.2024 до 10.11.2024
Робочий проект	Розробка ПЗ	З 11.11.2024 до 21.11.2024
	Розробка програмної документації	З 22.11.2024 до 01.12.2024
	Випробування програми	З 02.12.2024 до 05.12.2024

7 Порядок контролю та приймання

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

Контроль здійснюється замовником на кожній стадії розробки ПЗ. Приймання здійснюється замовником за програмою та методикою випробувань. Випробування проводиться в зазначений термін.

Хід проведення приймально-здавальних випробувань документується за допомогою протоколу проведення випробувань. На підставі протоколу проведення випробувань виконавач сумісно з замовником підписують акт приймання-здачі програми в експлуатацію.

ДОДАТОК Б

Програма та методика випробувань програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку

1 Об'єкт випробувань

Об'єктом випробувань є програмне забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку.

2 Мета випробувань

Мета випробувань – перевірка працездатності програмного застосунку, перевірка відповідності фактичних характеристик програмного застосунку тим вимогам, які було сформульовано 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:

- Введення у форму значення метрики CBO (Coupling between object classes) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).

- Введення у форму значення метрики WMC (Weighted methods per class) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).

- Введення у форму довірчої ймовірності.

- Оцінювання значення метрики RFC на основі введених значень метрик CBO та WMC з використанням побудованої нелінійної регресійної моделі.

– Визначення верхньої та нижньої меж довірчого інтервалу для значення RFC.

– Визначення верхньої та нижньої меж інтервалу прогнозування для значення RFC.

– Визначення якості web-застосунку на основі Spring-фреймворку.

4 Вимоги до програмної документації

Програмна документація включає в себе наступні документи:

- технічне завдання;
- програму та методику випробувань;
- інструкцію користувача;
- опис програмного забезпечення.
- текст програми

5 Засоби і порядок випробувань

Технічні засоби

Для роботи ПЗ необхідні технічні засоби з наступними мінімальними параметрами:

- Оперативна пам'ять – 1 Гб.
- Дисковий простір – 500 Мб.

Програмні засоби

Для роботи ПЗ необхідна операційна система Windows 7 та вище.

Порядок проведення випробувань

Випробування програмного забезпечення повинні виконуватися в наступному порядку:

- виконуються інструкції до кожної функції;
- робиться аналіз отриманих результатів і встановлюється відповідність програмного продукту вимогам і системним документам.

При виявленні помилок у роботі системи складається їх перелік і обговорюється термін їх виправлення розробником. Після цього замовник

проводить повторне тестування в повному обсязі (можливе використання нових чи додаткових тестів).

5 Методи випробувань

Випробування будемо проводити за стратегією «чорного ящика». Всі функції програми будуть проходити випробування. Перевірка працездатності програми виконується згідно з пунктом «Вимоги до функціональних характеристик» технічного завдання.

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

Порядок випробувань:

- Перевірка кожної форми графічного інтерфейсу на можливість введення даних.
- Перевірка кожної форми графічного інтерфейсу на введення невалідних даних.
- Перевірка можливості виконання оцінювання метрики RFC на основі введених значень метрик CBO та WMC з використанням побудованої нелінійної регресійної моделі.
- Перевірка можливості визначення верхньої та нижньої меж довірчого інтервалу для значення RFC.
- Перевірка можливості визначення верхньої та нижньої меж інтервалу прогнозування для значення RFC.
- Перевірка можливості визначення якості.

ДОДАТОК В

Інструкція користувача програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку

1 Призначення програми

Функціональним призначенням програми є надання користувачу можливості оцінити якість web-застосунків на основі Spring-фреймворку для визначення їх якості.

Програма призначена для експлуатації на персональних комп'ютерах користувачів для оцінювання метрики RFC через метрики СВО та RFC для web-застосунків на основі Spring-фреймворку для визначення їх якості.

Програмне забезпечення має можливість виконання наступних функцій:

- Введення у форму значення метрики СВО (Coupling between object classes) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).
- Введення у форму значення метрики WMC (Weighted methods per class) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).
- Введення у форму довірчої ймовірності.
- Оцінювання значення метрики RFC на основі введених значень метрик СВО та WMC з використанням побудованої нелінійної регресійної моделі.
- Визначення верхньої та нижньої меж довірчого інтервалу для значення RFC.
- Визначення верхньої та нижньої меж інтервалу прогнозування для значення RFC.
- Визначення якості web-застосунку на основі Spring-фреймворку.

2 Умови виконання програми

Для роботи ПЗ необхідні технічні засоби з наступними мінімальними параметрами:

- Оперативна пам'ять – 1 Гб.
- Дисковий простір – 500 Мб.

Програма повинна працювати під управлінням ОС Windows версії не нижче 7.

3 Виконання програми

При роботі з програмним забезпеченням необхідно виконати наступні дії:

– Запустити програмне забезпечення. Після цього відобразиться форма, яка представлена на рисунку В1.

Оцінювання якості Spring-фреймворк web-застосунків

Значення метрики CBO web-застосунку: 3,56

Значення метрики WMC web-застосунку: 7,25

Довірча ймовірність: 0.95

Оцінити

Результат оцінювання метрики RFC:

Якість:

Межі інтервалу довіри метрики RFC:

Межі інтервалу прогнозу метрики RFC:

Рисунок В.1 – Форма для введення даних для оцінювання

– Ввести у форму, яка з’явилася, значення метрики CBO (Coupling between object classes) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).

– Ввести у форму, яка з’явилася, значення метрики WMC (Weighted methods per class) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).

– Ввести у форму, яка з’явилася, довірчу ймовірність.

– Після введення вхідних даних натиснути кнопку «Оцінити» для отримання: оцінки значення метрики RFC, значення верхньої та нижньої меж довірчого інтервалу для RFC, значення верхньої та нижньої меж інтервалу прогнозування для RFC, якості програмного забезпечення.

Отримані розраховані значення будуть представлені в нижній частині монітору. Форма після розрахунку представлена на рисунку В2.

Значення метрики CBO web-застосунку:	3,56
Значення метрики WMC web-застосунку:	7,25
Довірча ймовірність:	0.95
<input type="button" value="Оцінити"/>	
Результат оцінювання метрики RFC:	5.5712
Якість:	середня
Межі інтервалу довіри метрики RFC:	4.6841 - 6.6264
Межі інтервалу прогнозу метрики RFC:	2.9860 - 10.3949

Рисунок В.2 – Форма після оцінювання

ДОДАТОК Г

Текст програми

```

# General libraries
import sys

# Libraries for calculations
import math
import numpy as np
from scipy import stats

# User interface libraries
from PyQt6.QtWidgets import (
    QApplication, QWidget, QLineEdit, QFormLayout, QVBoxLayout, QMainWindow,
    QGroupBox, QPushButton
)
from PyQt6.QtGui import QDoubleValidator, QPalette, QColor
from PyQt6.QtCore import Qt

class RFCEvaluator:
    N = 40
    k = 2
    V = N - k - 1

    B = [
        0.107413151,
        0.577545438,
        0.372006697
    ]

    SZY = 0.128409561

    # Inverse covariance matrix
    ICM = np.array([
        [1.908072374, 0.335364555],
        [0.335364555, 0.505590197]
    ])

    ZXMeans = [0.737321056, 0.770834467]

    def predictKLOC(self, X):
        return (10 ** self.B[0]) * (X[0] ** self.B[1]) * (X[1] ** self.B[2])

    def getSEZ(self, X):
        zx = np.log10(X) - self.ZXMeans
        return np.matmul(np.matmul(zx.T, self.ICM), zx)

```

```

def calcPredInterval(self, value, X, alpha):
    Y = math.log10(value)
    sez = self.getSEZ(X)
    predHalfIntervalValue = (stats.t.ppf(1 - alpha / 2, self.V) * self.SZY *
math.sqrt((1 + 1.0 / self.N + sez)))
    return 10 ** (Y - predHalfIntervalValue), 10 ** (Y + predHalfIntervalValue)

def calcConfInterval(self, value, X, alpha):
    Y = math.log10(value)
    sez = self.getSEZ(X)
    confHalfIntervalValue = (stats.t.ppf(1 - alpha / 2, self.V) * self.SZY *
math.sqrt((1.0 / self.N + sez)))
    return 10 ** (Y - confHalfIntervalValue), 10 ** (Y + confHalfIntervalValue)

class AppMainWindow(QMainWindow):
    def __init__(self):
        super().__init__()

        self.RFCEvaluator = RFCEvaluator()

        # Input for count of classes
        self.UICBOInput = QLineEdit("", self)
        self.UICBOInput.setValidator(QDoubleValidator(0.0, 1000.0, 2, parent=self))

        # Input for count of functions
        self.UIWMCInput = QLineEdit("", self)
        self.UIWMCInput.setValidator(QDoubleValidator(0.0, 1000.0, 2, parent=self))

        # Input for value of confidence
        self.UICConfidenceInput = QLineEdit("0.95", self)
        self.UICConfidenceInput.setValidator(QDoubleValidator(0.0, 1.0, 2,
parent=self))

        # Output for result predicted KLOC
        self.UIQualityOutput = QLineEdit(self)
        self.UIQualityOutput.setReadOnly(True)

        self.UIQualityInput = QLineEdit(self)
        self.UIQualityInput.setReadOnly(True)

        # Output for result predicted confidence interval
        self.UICConfidenceIntervalInput = QLineEdit(self)
        self.UICConfidenceIntervalInput.setReadOnly(True)

        # Output for result predicted prediction interval
        self.UIPredictionIntervalInput = QLineEdit(self)
        self.UIPredictionIntervalInput.setReadOnly(True)

        self.UIEvaluateButton = QPushButton('Оцінити')

```

```

self.UIEvaluateButton.clicked.connect(self.execCalculations)

UIDataInputGroup = QGroupBox("", self)
UIDataOutputGroup = QGroupBox("", self)

UIMainWidget = QWidget()
self.setCentralWidget(UIMainWidget)

UIMainLayout = QVBoxLayout(UIMainWidget)
UIMainLayout.addWidget(UIDataInputGroup)
UIMainLayout.addWidget(UIDataOutputGroup)

# Add UI items to data input group layout
UIDataInputGroupLayout = QFormLayout(UIDataInputGroup)
UIDataInputGroupLayout.addRow("Значення метрики СВО web-застосунку:",
self.UICBOInput)
UIDataInputGroupLayout.addRow("Значення метрики WMC web-застосунку:",
self.UIWMCInput)
UIDataInputGroupLayout.addRow("Довірча ймовірність:",
self.UIConfidenceInput)
UIDataInputGroupLayout.addRow(self.UIEvaluateButton)

# Add UI items to data output group layout
UIDataOutputGroupLayout = QFormLayout(UIDataOutputGroup)
UIDataOutputGroupLayout.addRow("Результат оцінювання метрики RFC:",
self.UIQualityOutput)
UIDataOutputGroupLayout.addRow("Якість:", self.UIQualityInput)
UIDataOutputGroupLayout.addRow("Межі інтервалу довіри метрики RFC:",
self.UIConfidenceIntervalInput)
UIDataOutputGroupLayout.addRow("Межі інтервалу прогнозу метрики RFC:",
self.UIPredictionIntervalInput)

def getAlpha(self):
    return 1 - float(self.UIConfidenceInput.text())

def execCalculations(self):
    X = [
        float(self.UICBOInput.text().replace(",", ".")),
        float(self.UIWMCInput.text().replace(",", "."))
    ]

    alpha = self.getAlpha()
    value = self.RFCEvaluator.predictKLOC(X)
    predIntervalValues = self.RFCEvaluator.calcPredInterval(value, X, alpha)
    confIntervalValues = self.RFCEvaluator.calcConfInterval(value, X, alpha)

    if value >= confIntervalValues[0] and value <= confIntervalValues[1]:
        quality = 'середня'
    elif value >= predIntervalValues[0] and value <= confIntervalValues[0] or
value >= confIntervalValues[1] and value <= predIntervalValues[1]:

```

```

        quality = 'висока'
    elif value >= predIntervalValues[1]:
        quality = 'низька'

    self.UIQualityOutput.setText(f'{value:.4f}')
    self.UIQualityInput.setText(quality)
    self.UIConfidenceIntervalInput.setText(f'{confIntervalValues[0]:.4f} -
{confIntervalValues[1]:.4f}')
    self.UIPredictionIntervalInput.setText(f'{predIntervalValues[0]:.4f} -
{predIntervalValues[1]:.4f}')

if __name__ == '__main__':
    app = QApplication(sys.argv)

    # Set Fusion style
    app.setStyle("Fusion")

    # Customize palette
    palette = QPalette()
    palette.setColor(QPalette.ColorRole.Window, QColor(239, 239, 244))
    palette.setColor(QPalette.ColorRole.WindowText, Qt.GlobalColor.black)
    palette.setColor(QPalette.ColorRole.Base, QColor(255, 255, 255))
    palette.setColor(QPalette.ColorRole.AlternateBase, QColor(242, 242, 242))
    palette.setColor(QPalette.ColorRole.Text, Qt.GlobalColor.black)
    palette.setColor(QPalette.ColorRole.Button, QColor(239, 239, 244))
    palette.setColor(QPalette.ColorRole.ButtonText, Qt.GlobalColor.black)
    app.setPalette(palette)

    window = AppMainWindow()
    window.setWindowTitle("Оцінювання якості Spring-фреймворк web-застосунків")
    window.resize(600, 400)
    window.show()
    sys.exit(app.exec())

```

ДОДАТОК Д

Опис програмного забезпечення для оцінювання якості web-застосунків на основі Spring-фреймворку

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку

1.2 Мови програмування

ПЗ написане на мові програмування Python та використанням бібліотек NumPy та SciPy.

2 Функціональне призначення

2.1 Класи вирішуваних завдань і призначення програми

ПЗ для оцінювання якості web-застосунків на основі Spring-фреймворку повинно забезпечувати виконання наступних функцій:

- Введення у форму значення метрики CBO (Coupling between object classes) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).

- Введення у форму значення метрики WMC (Weighted methods per class) web-застосунку на основі Spring-фреймворку для оцінювання значення його метрики RFC (Response for a class).

- Введення у форму довірчої ймовірності.

- Оцінювання значення метрики RFC на основі введених значень метрик CBO та WMC з використанням побудованої нелінійної регресійної моделі.

- Визначення верхньої та нижньої меж довірчого інтервалу для значення RFC.

– Визначення верхньої та нижньої меж інтервалу прогнозування для значення RFC.

– Оцінювання якості web-застосунку на основі Spring-фреймворку.

2.2 Функціональні обмеження

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

3 Використовувані технічні засоби

Для роботи ПЗ необхідні технічні засоби з наступними мінімальними параметрами:

– Оперативна пам'ять – 1 Гб.

– Дисковий простір – 500 Мб.

Програма повинна працювати під управлінням ОС Windows версії не нижче 7.

4 – Виклик та завантаження

Для запуску та подальшої роботи програмного забезпечення необхідна операційна система Windows 7 та вище.

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

5 Вхідні та вихідні дані

Вхідними даними для програми є значення метрик CBO та WMC, довірна ймовірність.

Вихідними даними є оцінене значення метрики RFC, верхня та нижня межі довірчого інтервалу, верхня та нижня межі інтервалу прогнозування, оцінене значення якості.