

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

«__» _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

на тему: Регресійна модель для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3, та розробка програми для її реалізації.

Виконав: студент 6151м групи

Котула В.С. _____

(підпис, ПІБ)

Керівник роботи:

Доцент, к.т.н. Пономаренко Т.В.

(посада, науковий ступень вчене звання)

(підпис, ПІБ)

Миколаїв – 2021 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)

«__» _____ 2021 р.

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

Студенту Котулі Віктору Степановичу _____

1. Тема роботи: Регресійна модель для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3, та розробка програми для її реалізації.

Керівник роботи Доцент каф. ПЗАС, к.т.н., доцент Пономаренко Т.В. _____

Затверджені наказом ректора № 1239уч від «13» _____ 10 _____ 2021 р.

2. Термін подання роботи: 06.12.2021 р. _____

3. Вихідні дані по роботі: _____

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

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

5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та	Підпис, дата
--------	-----------------------	--------------

	посада консультанта	завдання видав	завдання прийняв

7. Дата видачі завдання 13.10.2021 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	18.10.2021	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	19.10.2021	НС*
3.	Підготовка розділу (ів) з результатів власних досліджень	22.10.2021	НС*
4.	Підготовка розділу з проекту програмного забезпечення	16.11.2021	
5.	Підготовка організаційно-економічного розділу	19.11.2021	
6.	Підготовка розділу з охорони праці	22.11.2021	
7.	Підготовка розділу з охорони навколишнього середовища	24.11.2021	
8.	Підготовка розділу Висновки	25.11.2021	
9.	Оформлення списку використаних джерел та додатків	29.11.2021	
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.)	06.12.2021	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
11.	Підготовка презентації та доповіді	09.12.2021	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	10.12.2021	
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	20.12.2021	

* - за результатами наукового стажування (НС), яке було з 01.09.2021 до 10.10.2021 р.

Студент

(підпис)

(ПІБ)

Керівник роботи

(підпис)

(ПІБ)

РЕФЕРАТ

Котула Віктор Степанович

«Регресійна модель для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3, 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,

2021 р.

Обсяг роботи: 80 стор., 20 табл., 14 рис., 23 використаних джерела, 5 додатків.

Актуальність теми роботи. Оцінювання трудомісткості усіх процесів розробки програмного забезпечення, включаючи виправлення недоробок розробниками, особливо для нових інноваційних фреймворків таких як Yii3 є актуальною задачею.

Мета і завдання дослідження. Метою роботи є підвищення достовірності оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- 1) проаналізувати існуючі алгоритми та методи оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3, сформулювати постановку задачі;
- 2) удосконалити математичну модель оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3;
- 3) розробити програмне забезпечення для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

Об'єктом дослідження є процес оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

Предметом дослідження є регресійна модель оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

Методи дослідження. Для вирішення поставлених задач були застосовані методи теорії ймовірностей та математичної статистики, регресійного аналізу, побудови нелінійних регресійних моделей на основі нормалізуючих перетворень.

Наукова новизна одержаних результатів полягає в удосконаленні багатфакторної регресійної моделі для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3 на основі нормалізуючого перетворення на основі натурального логарифму. Це дозволяє підвищити достовірність оцінювання трудомісткості виправлення недоробок розробниками програмного забезпечення, що створюють його за допомогою Yii3.

Практичне значення отриманих результатів полягає в розробці програмного забезпечення для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

Апробація результатів досліджень. Результати досліджень викладені у роботі, були оприлюднені у Матеріалах II Всеукраїнської науково-практичної інтернет конференції ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: МОДЕЛІ, АЛГОРИТМИ, СИСТЕМИ (ITMAS – 2021).

Публікації. Опубліковано тези доповіді на II Всеукраїнській науково-практичній інтернет конференції ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: МОДЕЛІ, АЛГОРИТМИ, СИСТЕМИ (ITMAS – 2021).

Ключові слова: багатфакторна регресійна модель, нормалізуюче перетворення, трудомісткість виправлення недоробок, Yii3, веб застосунок.

ABSTRACT

Kotula Victor

«A regression model for estimating the effort of fixing the bugs of web applications created using the Yii3 framework and developing the software for its implementation»

Qualification work for obtaining a master's degree in specialty 121 – "Software Engineering".

Admiral Makarov National University of Shipbuilding. Mykolayiv,
2021

The work contains: 80 pages, 20 tables, 14 figures, 29 references, 5 appendices.

Relevance of the topic. Assessing the complexity of all software development processes, including debugging by developers, especially for new innovative frameworks such as Yii3 is an urgent task.

The purpose and objectives of the study. The aim of the work is to increase the reliability of estimating the complexity of correcting the shortcomings of web applications created using the Yii3 framework.

To achieve this goal it is necessary to solve the following tasks:

- 1) to analyze the existing algorithms and methods for estimating the complexity of correcting the shortcomings of web applications created using the Yii3 framework, to formulate the problem statement;
- 2) to improve the mathematical model for estimating the complexity of correcting the shortcomings of web applications created using the Yii3 framework;
- 3) develop software to assess the complexity of correcting the shortcomings of web applications created using the Yii3 framework.

The object of the research is the process of estimating the complexity of correcting the shortcomings of web applications created using the Yii3 framework.

The subject of the study is a regression model for estimating the complexity of correcting the shortcomings of web applications created using the Yii3 framework.

Research methods. Methods of probability theory and mathematical statistics, regression analysis, construction of nonlinear regression models based on normalizing transformations were used to solve the problems.

The scientific novelty of the obtained results is to improve the multifactor regression model for estimating the complexity of correcting the shortcomings of web applications created using the Yii3 framework through the use of normalization conversion using the decimal logarithm, which increased the reliability of estimating existing model.

The practical significance of the results is to develop software to assess the complexity of correcting the shortcomings of web applications created using the Yii3 framework.

Approbation of research results. The research results are presented in the paper, were published in the Proceedings of the II All-Ukrainian scientific-practical Internet conference INFORMATION TECHNOLOGIES: MODELS, ALGORITHMS, SYSTEMS (ITMAS – 2021).

Publications. Abstracts of the report were published at the II All-Ukrainian scientific-practical Internet conference INFORMATION TECHNOLOGIES: MODELS, ALGORITHMS, SYSTEMS (ITMAS – 2021).

Keywords: REGRESSION MODEL, NORMALIZING TRANSFORMATION, BUGFIX EFFORT, YII3 FRAMEWORK, WEB APPLICATION.

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК	12
1.1 Аналіз особливостей фреймворка YІІЗ	12
1.2 Аналіз існуючих методів та моделей оцінювання трудомісткості	14
1.3 Існуюча модель трудомісткості виправлення недоробок	15
2 МАТЕМАТИЧНА МОДЕЛЬ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКА YІІЗ	21
3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКА YІІЗ	36
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕАЛІЗАЦІЇ МАТЕМАТИЧНОЇ МОДЕЛІ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКА YІІЗ	45
5 ОХОРОНА ПРАЦІ	51
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	57
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ	67
ДОДАТОК Б – ТЕКСТ ПРОГРАМИ	74

ДОДАТОК В – ПРОГРАМА І МЕТОДИКА ВИПРОБОВУВАНЬ	76
ДОДАТОК Г – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	78
ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА	79

ВСТУП

Актуальність теми. Працюючи над якістю програмного забезпечення команда проекту, де функціонують з однієї сторони спеціалісти QA відділу , а з іншої розробники, повинна мати інформацію про час який необхідний розробникам на усунення знайдених тестувальниками недоробок. За результатами цього процесу формується bugfix release. У роботі [1] автори побудували тринадцятофакторну модель оцінювання необхідного часу на виправлення недоробок розробниками програмного забезпечення для Mosilla на величезному наборі даних, але відзначили, що якість моделі недосконала і її ще можна покращити. Застосування результатів таких масштабних розробок для інноваційних свіжих фреймворків є актуальною задачею.

Для оцінки авторами [2] було використано цю модель, побудовану на наборі даних, який містить 7027 записів. Кожен запис складається з набору показників, які пов'язані з єдиною унікальною транзакцією виправлення недоробок ревізій вихідного файлу. Результати оцінювання якості, отримані для моделі [3] прогнозування трудомісткості виправлення недоробок були наступними R2 0.53, MAE 11.88, RMSE 23.5, MMRE 0.81, RMRE8 4.7

Мета і завдання дослідження. Метою роботи є підвищення достовірності оцінювання необхідного часу на виправлення недоробок розробниками програмного забезпечення, що розробляється за допомогою фреймворку Yii3 шляхом удосконалення математичної моделі для оцінювання трудомісткості виправлення недоробок веб застосунків на виправлення недоробок розробниками програмного забезпечення за допомогою фреймворку Yii3 та розробка програмного забезпечення для її реалізації.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1) проаналізувати існуючі алгоритми та методи оцінювання трудомісткості виправлення недоробок розробниками програмного забезпечення (bugfix), сформулювати постановку задачі;

2) удосконалити математичну модель оцінювання трудомісткості виправлення недоробок розробниками програмного забезпечення (bugfix) розробниками програмного забезпечення;

3) розробити програмне забезпечення для оцінювання трудомісткості виправлення недоробок розробниками програмного забезпечення (bugfix).

Об'єктом дослідження є процес оцінювання трудомісткості виправлення недоробок розробниками програмного забезпечення (bugfix) розробниками програмного забезпечення.

Предметом дослідження є регресійна модель трудомісткості виправлення недоробок розробниками програмного забезпечення (bugfix) розробниками програмного забезпечення.

Методи дослідження. Для вирішення поставлених задач були застосовані методи теорії ймовірностей та математичної статистики, регресійного аналізу, побудови нелінійних регресійних моделей на основі нормалізуючих перетворень за допомогою натурального логарифму.

Наукова новизна одержаних результатів полягає в удосконаленні оренні багатofакторної регресійної моделі для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3 на основі існуючої відповідної моделі для аналогічного програмного забезпечення, що дозволило підвищити достовірність оцінювання необхідного часу на виправлення недоробок розробниками програмного забезпечення, що створюють його за допомогою Yii3.

Практичне значення отриманих результатів полягає в розробці ПЗ для оцінювання трудомісткості виправлення недоробок розробниками програмного забезпечення, що створюється за допомогою фреймворку Yii3.

Апробація результатів досліджень. Результати досліджень викладені у роботі, були оприлюднені у Матеріалах II Всеукраїнської науково-практичної інтернет конференції ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: МОДЕЛІ, АЛГОРИТМИ, СИСТЕМИ (ITMAS – 2021).

Особистий внесок здобувача. Усі основні результати кваліфікаційної роботи отримані здобувачем самостійно. Кваліфікаційна робота є самостійно виконаною працею. У тезах доповіді [4], написаних у співавторстві, здобувачу належить створення програми для побудови моделі.

Публікації. Опубліковано тези доповіді на II Всеукраїнській науково-практичній інтернет конференції ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: МОДЕЛІ, АЛГОРИТМИ, СИСТЕМИ (ITMAS – 2021).

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК ВЕБ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ Yii3

1.1 Аналіз особливостей фреймворка Yii3.

На сьогодні, якщо пошукати на github.com можна знайти більше 200 проектів за ключовим словом Yii3, результати пошуку наведено на рисунку 1. Частина результатів пошуку це модулі цього нового фреймворку, а інша частина – це пілотні проекти створені з його використанням, що націлені на раннє виявлення недоліків.

Розробники веб застосунків, що створюються з використанням нових фреймворків завжди дуже залежать від наявності супровідної документації, а у молодих фреймворків вона завжди неповна, тому розробка додаткового інструменту, що дозволить оцінювати трудомісткість виправлення недоробок веб застосунків з використанням нових фреймворків, зокрема Yii3 є актуальною.

Yii3, Yii2 та Yii1 дуже відрізняються за структурою, принципами архітектури та складністю розробки. Якщо аналізувати розмір проектів зі схожим функціоналом в цих фреймворках є дуже значне спрощення для розробників з кожною новою версією і дуже велика різниця у обсягах коду – з кожним новим нащадком писати вручну треба все менше, але у останній версії є значне ускладнення в зв'язку з тим що він зовсім новесенький і постійно доробляється. Розробка ведеться командою ентузіастів які в свій вільний час роблять це за дуже незначну винагороду, порівняно з основним місцем роботи, отож документованість і швидкість усунення недоробок зростає. Існує цілий ряд розробників, які вкладають свій вільний час в

розробку нових фреймворків, частина агенств та інших софтверних компаній виділяє ресурси для розвитку таких проектів. Незважаючи на те, що платформа ще знаходиться у бетаверсії, є активні користувачі і сміливі розробники, що реалізують проекти за її допомогою. На рисунку 1.1 наведено результати пошуку по ключовим словам Yii3 на github.com де видно що таких проектів вже більше 200.

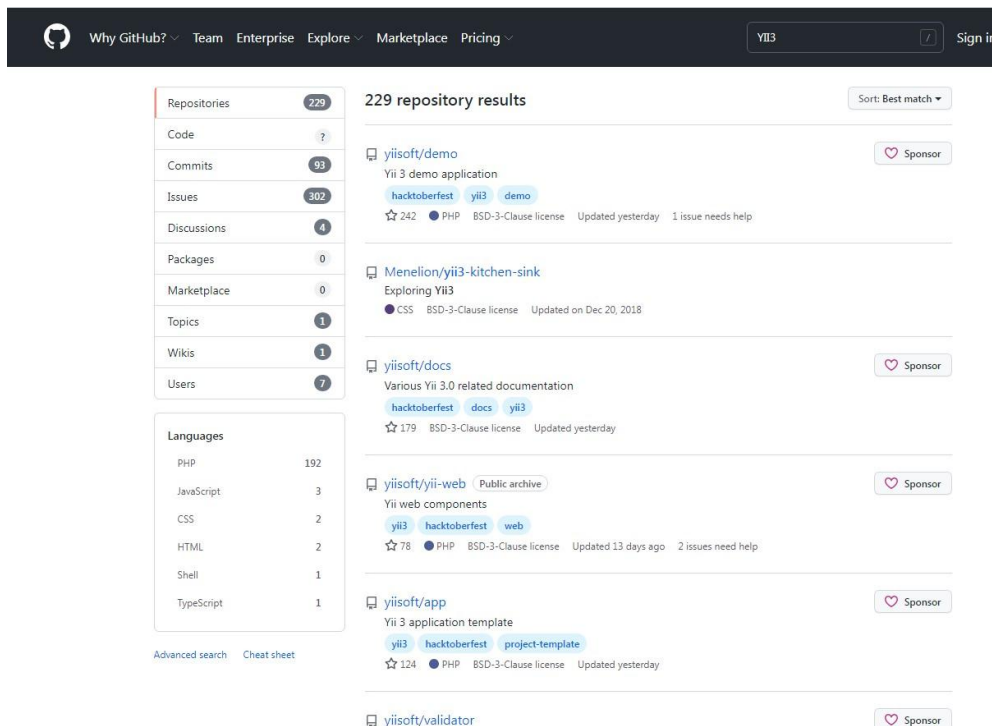


Рисунок 1.1 – Результати пошуку проектів Yii3 на Github.com

Після знайдення проектів, що підходять для наших цілей їх було клоновано за допомогою програми github desktop (file-clone repository). Це зроблено для того щоб інформація про зміни в репозиторіях файлів між версіями була доступна для аналізу. Інтерфейс програми GitHub Desktop наведено на рисунку 1.2.

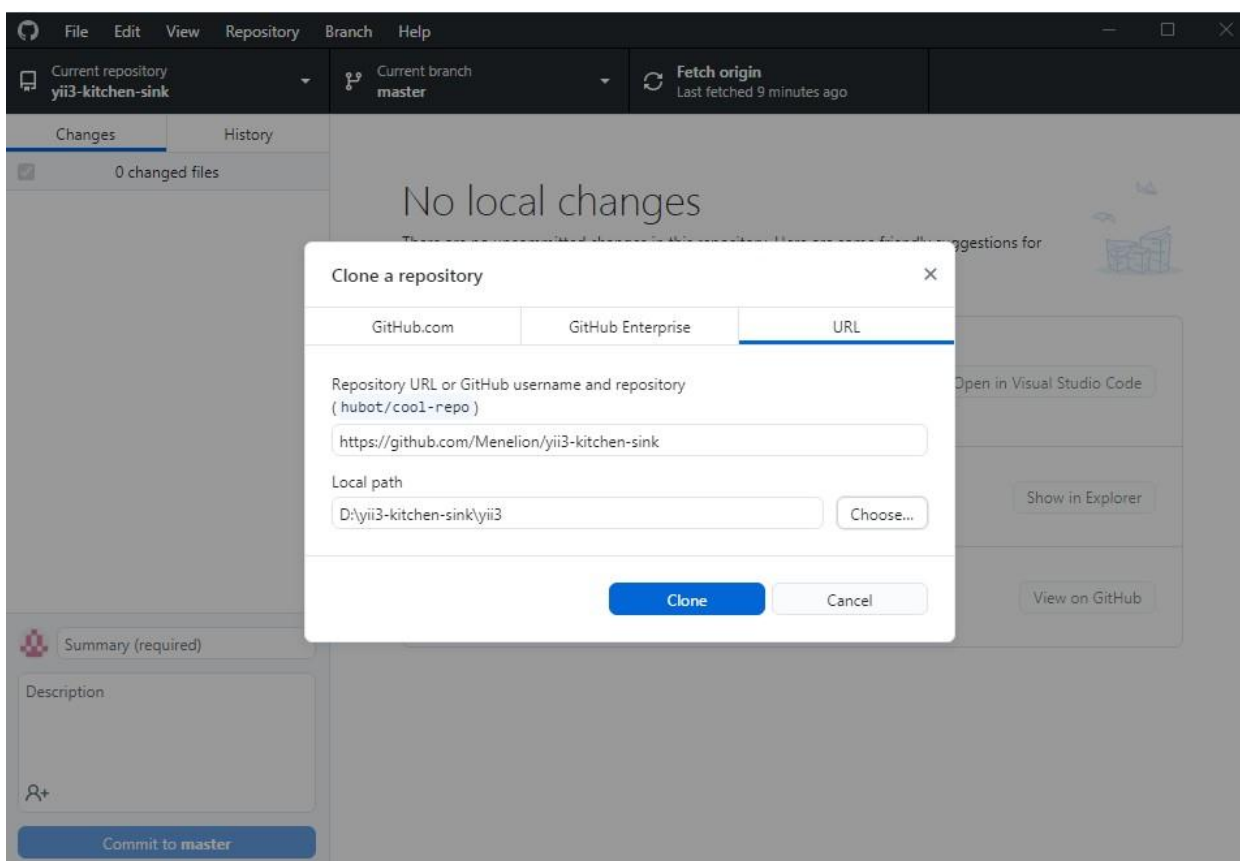


Рисунок 1.2 – Процес клонування проекту на локальний комп'ютер через програму GitHub Desktop

1.2 Аналіз існуючих моделей оцінювання трудомісткості виправлення недоробок

На ранніх етапах розробки проекту для оцінювання необхідного обсягу тестування можна задати високорівневі оцінки часу, необхідного для виконання дій з планування тесту, часу і трудомісткості виконання всіх тестів. Ці оцінки ґрунтуються на тому, що відомо про вимоги проекту.

Одиницею вимірювання трудомісткості тестування можуть бути:

- РН: людино-година;
- РД: людино-день;
- РМ: людино-місяць;
- РҮ: людино-рік.

Виділяють три типи програмних проектів залежно від розміру за моделлю COSOMO II для оцінки продуктивності та трудомісткості. Це вбудований, напівокремий та органічний типи проектів. Відрізняються вони за розміром, часовими обмеженнями, кількістю людей у команді та інноваційністю. Особливості градації та стандартні коефіцієнти для таких проектів наведено у таблиці 1.1.

Таблиця 1.1 Типи проекту

Тип проекту	Розмір у тисячах рядків коду	Характеристики	Стандартні коефіцієнти
Органічний	2-50 KLOC	малий, стабільний, мало інновацій	• a: 2.4 • b: 1.05 • c: 2.5 • d: 0.38
Напівокремий	50-300 KLOC	середній розмір, середні можливості, середні часові обмеження	• a: 3.0 • b: 1.12 • c: 2.5 • d: 0.35
Вбудований	> 300 KLOC	велика проектна команда, складний, інноваційний, суворі обмеження	• a: 3.6 • b: 1.2 • c: 2.5 • d: 0.32

Cosomo для середніх проектів визначає наступні вартісні метрики програмних продуктів за стандартними моделями.

Атрибути продуктів:

- **RELY** Required Software Reliability;
- **DATA** Data Base Size;
- **CPLX** Product Complexity.

Атрибути технічного забезпечення проектів програмного забезпечення (Computer Attributes):

- **TIME** Execution Time Constraint;

- **STOR** Main Storage Constraint;
- **VIRT** Virtual Machine Volatility;
 - **TURN** Computer Turnaround Time.

Атрибути фахівців:

- **ACAP** Analyst Capability;
- **AEXP** Application Experience;
- **PCAP** Programming Capability;
- **VEXP** Virtual Machine Experience;
- **LEXP** Programming Language Experience.

Атрибути проектів:

- **MODP** Modern Programming Practices;
- **TOOL** Use of Software Tools;
- **SCED** Required Development Schedule.

На сьогоднішній день також існують наступні методи та моделі для оцінювання трудомісткості тестування програмного забезпечення:

- **COQUALMO** (Constructive Quality Model) – модель для оцінювання трудовитрат на тестування, яка є похідною від моделі **COCOMO**. Вона призначена для оцінювання трудових витрат на розробку програмного продукту в цілому (з урахуванням усіх етапів);
- методи, засновані на інформації про кількість рядків коду **SLOC** (Source Lines of Code);
- методи на основі функціональних точок та варіантів використання;
- методи експертних оцінок, що використовують попередній досвід розробки аналогічних проектів;
- моделі оцінювання трудовитрат на тестування як відсотків від трудовитрат на кодування ПЗ [1-3].

В реальних умовах практично неможливо отримати такі дані про програмні проекти, які будуть розподілені за нормальним законом, і зв'язки між цими даними не завжди є лінійними. Це доводять результати побудови інтервалів передбачення, наведені в табл. 1.1. Інтервал, який було

побудовано за допомогою нелінійної регресії та нормалізації даних, не має від'ємних значень нижньої границі для жодного з програмних проектів і для 8 з 14 проектів є меншим на 43.65% за інтервал передбачення лінійної регресії.

Саме тому розроблювана математична модель також повинна враховувати нелінійність зв'язків між емпіричними даними та використовувати їх нормалізацію.

В залежності від способу отримання даних для визначення трудомісткості, існує також модель Путнема (SLIM (Software Life-cycle Model)), метод функціональних точок, нейронні мережі, методи імітаційного моделювання, динамічні методи, композитні методи. Такі методи є комбінацією двох і більше методів для формування найбільш підходящої системи оцінки.

Як впливає з аналізу сучасних підходів до розрахунку трудомісткості тестування, на даний момент існує велика кількість методів та моделей для її оцінювання.

Використання командою проекту цих методів та моделей є одним з головних аргументів при техніко-економічному обґрунтуванні вартості розробки. Але частина з цих підходів вже застаріла та їх використання дає неадекватний результат, який приводить до несприятливих наслідків.

Окрім того, прогнози щодо точності оцінювання трудомісткості тестування програмного забезпечення існуючими методами та моделями, зазвичай працюють лише для певного типу проектів, аналогічних тим, на яких ті чи інші моделі чи методи були випробувані.

Тобто, не існує універсальних методів чи моделей, які є оптимальними для всіх проектів і середовищ розробки. Також в більшості підходів враховують лише лінійні зв'язки між факторами.

Для розроблених чи вдосконалених методів та моделей обов'язковою є перевірка їх якості [9, 13].

Для перевірки якості розробленої математичної моделі використаємо наступні показники: коефіцієнт детермінації, середню величину відносної похибки, рівень прогнозування [13, 14, 15].

Коефіцієнт детермінації оцінює долю дисперсії залежної змінної, яка пояснюється з допомогою незалежної змінної в лінійній регресійній моделі. обчислюється за формулою:

$$R^2 = 1 - (\sum_1^n (y_i - \hat{y}_i)^2 / \sum_1^n (y_i - \bar{y})^2),$$

де y_i – емпіричне значення; $\hat{y}_i$ – розраховане значення; $\bar{y}$ – середнє значення; n – кількість емпіричних даних.

Коефіцієнт детермінації приймає значення від 0 (відсутній лінійний зв'язок між показниками) до 1 (відсутній кореляційний зв'язок між показниками). Моделі з коефіцієнтом детермінації $>0,5$ вважаються допустимими, а з $>0,8$ – достатньо ефективними.

Середня величина відносної похибки MMRE визначається за формулою: .

$$MMRE = \frac{1}{n} \cdot \sum_1^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| .$$

Рівень прогнозування обчислюється за формулою:

$$PRED(l) = \frac{k}{n}, \quad (1.3)$$

де k – кількість значень $\left| \frac{y_i - \hat{y}_i}{y_i} \right| \leq l \ (i = \overline{1, n})$.

Із вищевикладеного випливає, що на сьогодні існує велика кількість методів та моделей оцінювання трудомісткості тестування, як одного з складових розробки але частина з цих підходів має низьку точність оцінювання, не враховує нелінійність зв'язків між факторами, призначена

для даних, що підпорядковуються нормальному закону розподілу. Також відомо, що не існує універсальних методів чи моделей, які є оптимальними для всіх типів програмного забезпечення і середовищ розробки. Тому існує необхідність удосконалювати існуючі та розробляти нові методи та моделі оцінювання трудомісткості тестування програмного забезпечення розробленого за допомогою фреймворка Yii3. Це дозволить підвищити точність оцінювання трудомісткості тестування з урахуванням нелінійних зв'язків між даними, що не підпорядковуються нормальному закону розподілу.

1.3 Існуюча модель трудомісткості виправлення недоробок.

У 2007 році група розробників на замовлення Пакистану провела ряд досліджень і побудувала регресійну модель виправлення недоробок. Для розробки надійної моделі оцінки авторами [1] були використані регресія, нейронну мережа, машину опорних векторів, правила класифікації та методи дерева рішень. Крім того, було порівняно результати різних методів з використанням кількох критеріїв оцінки. Вони прийшли до висновку, що у випадку машини опорних векторів відносна абсолютна похибка (MMRE) є найнижчою, тоді як у випадку застосувань правил класифікації кореляція між фактичними та прогнозованими значеннями зусилля є найвищою.

Модель оцінювання трудомісткості виправлення недоробок для системи програмного забезпечення з відкритим кодом відіграє важливу роль у забезпеченні якості програмного забезпечення та управлінні проектами програмного забезпечення. Більшість моделей оцінки трудомісткості пов'язані з комерційними або закритими програмними системами, тоді як важко розробити модель оцінки трудомісткості для програмних систем з відкритим кодом [1].

Причинами можуть бути велика кількість вихідних файлів, а також велика кількість розробників програмного забезпечення та учасників. В

умовах всесвітньої пандемії розробники та учасники випадковим чином поширені по всьому світу. Отримати фактичний час, витрачений кожним, складно. Задля подолання цих проблем, застосовано евристичний підхід, який оцінює фактичний час виправлення недоробок. Загальний час, витрачений на виправлення помилки розглядається як зусилля з виправлення недоробок. Для проведення експериментів було вибрано відкритий код проекту та витягнуто дані про виправлення недоробок із відповідного сервера та завантажено редакції вихідних файлів із репозиторію контролю версій.

2 МАТЕМАТИЧНА МОДЕЛЬ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК у ВЕБ ЗАСТОСУНКАХ , ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКА YII3

2.1 Регресійна модель для оцінювання трудомісткості виправлення недоробок, що створюються за допомогою фреймворка YII3.

Для отримання значень метрик було проведено клонування репозиторіїв з github. Було використано підхід, описаний у [1], щоб ідентифікувати лише зміни, які стосуються виправлення недоробок в версії.

Розглядаються лише відмінності між двома послідовними версіями, тобто випуском, у якому виправлено помилку, та її безпосереднім попередником, кожного вибраного вихідного файлу. Ці відмінності обробляються для визначення типу змін програмного файлу.

Було збережено всі дані про виправлення недоробок у наборі даних.

У таблиці 1.1 наведено набір показників програмного файлу, який був використаний для розробки моделі оцінки трудомісткості виправлення недоробок, що створюються за допомогою фреймворка YII3. Список показників не є остаточним; деякі інші показники також можуть бути додані до списку, як-от серйозність недоробок, голосування за помилки тощо.

Щоб отримати набір показників, у [1] було оброблено набір даних про зміну статусу недоробок та відповідні вихідні файли. Підраховано загальну кількість ревізій виправлень недоробок для всіх вибраних вихідних файлів. Дані програми потім оброблялися, щоб з'ясувати загальну кількість змінених рядків коду (LOC); експертизу розробника DEXP: вона отримана шляхом додавання значень рангу всіх тих розробників, які беруть участь у виправленні помилки; TPFREV загальна кількість попередніх виправлень вихідних файлів; TLOC загальна кількість рядків вихідного коду; TCLOC всього змінених рядків коду; TDELTA загальна кількість змін у вихідних

файлах; TFINC загальна кількість включених вихідних файлів/пакетів; TFUNC загальна кількість функцій; TELINE загальна кількість виконуваних рядків; TCYCLO загальні показники цикломатичної складності; TPCOUNT загальна кількість параметрів; TRPOINT загальна кількість точок повернення; TCOPE загальна кількість змінених операторів. За результатами авторами [1] було побудовано наступне рівняння:

$$\begin{aligned} \text{EFFORT} = & -12.9 - 13.8 \times \text{DCOUNT} - 1.09 \times \text{DEXP} + 5.8 \times \text{SFCOUNT} + 0.01 \times \text{TPREV} \\ & + 0.02 \times \text{TPFREVE} + 0.001 \times \text{TLOC} + 0.021 \times \text{TCLOC} - 0.035 \times \text{TDELTA} - \\ & 0.034 \times \text{TFUNC} + 0.004 \times \text{TELINE} - 0.025 \times \text{TCYCLO} + 0.02 \times \text{TPCOUNT}. \end{aligned}$$

Регресійні коефіцієнти (або В-коефіцієнти) являють собою незалежні вклади кожної незалежної змінної в результати.

Отож значення регресійних коефіцієнтів говорять про найбільшу значимість DCOUNT, DEXP, TCLOC, TDELTA, TELINE, TFUNC, тому для побудови рівняння для Y_{ii3} обрані 6 факторів (табл.3.2). Для побудови нелінійного регресійного рівняння використовуємо вибірку шестивимірних даних з наступних метрик: загальної кількості змінених рядків коду (LOC); експертизу розробника DEXP: вона отримана шляхом додавання значень рангу всіх тих розробників, які беруть участь у виправленні помилки; TPFREVE загальна кількість попередніх виправлень вихідних файлів; TLOC загальна кількість рядків вихідного коду; TCLOC всього змінених рядків коду; TDELTA загальна кількість змін у вихідних файлах; TFINC загальна кількість включених вихідних файлів/пакетів; TFUNC загальна кількість функцій; TELINE загальна кількість виконуваних рядків; TCYCLO загальні показники цикломатичної складності; TPCOUNT загальна кількість параметрів; TRPOINT загальна кількість точок повернення; TCOPE загальна кількість змінених операторів.

Таблиця 2.1 – Вибірка 32 програм із 6 факторами оцінки трудомісткості виправлення недоробок, що створюються за допомогою фреймворка YІІЗ

№ про екту	Effort Y	DCOUN T x ₁	DEXP x ₂	TCLOC x ₃	TDELTA x ₄	TELINE _{x₅}	TFUNC x ₆
1	6,182	5	7,2	7,2	0,8	0,6	1
2	6,182	8	2,75	3,5	1,25	0,13	1
3	6,314	9	3	5,56	1,89	0,56	1
4	6,358	9	3	5,78	2,11	0,56	1
5	6,482	9	3,11	5,78	2,11	0,56	1
6	6,583	10	3,3	5,5	1,1	0,9	1,1
7	6,605	8	4,38	4,5	1,38	0,25	1
8	6,749	11	3,55	7,27	0,82	0,73	1
9	6,863	19	4,95	3,42	0,89	0,37	1,11
10	6,879	10	4,3	5,7	2,1	0,4	1
11	6,948	22	1,91	3	0,41	0,23	1,05
12	6,972	7	8	7,14	0,71	0,43	1
13	7,007	25	2,68	2,6	0,44	0,32	1,04
14	7,015	10	4,3	6,9	0,6	0,4	1
15	7,107	9	4,44	3,33	0,67	0,44	1
16	7,124	12	6,33	4,67	1,08	0,33	1
17	7,153	11	5,27	7,18	1	0,64	1,09
18	7,169	23	3,65	2,87	0,43	0,35	1,04
18	7,219	27	2,04	3,07	0,59	0,19	1,04
20	7,269	35	2,69	2,26	0,26	0,17	1,06
21	7,292	12	7,33	6,67	0,5	0,42	1
22	7,301	19	5,74	5,47	0,79	0,26	1
23	7,375	21	2,1	3,67	0,67	0,24	1,05
24	7,383	20	4,2	5,7	2,7	0,25	1,1
25	7,428	24	9,17	3,38	0,46	0,33	1
26	7,441	27	3	5,56	1,89	0,56	1
27	7,451	27	3	5,78	2,11	0,56	1
28	7,484	27	3	5,78	2,11	0,56	1
29	7,707	32	3,44	3,25	0,34	0,03	1
30	7,902	30	3,2	3,37	0,07	0	1
31	7,98	66	2,3	3,45	0,79	0,82	1,12
32	8,725	117	6,33	3,34	0,66	0,37	1,02

Рівняння множинної регресії може бути представлено у вигляді $Y = X * a \pm \varepsilon$, де $X = X(x_1, x_2, \dots, x_m)$ – вектор незалежних (пояснюють) змінних; b – вектор параметрів

Визначимо вектор оцінок коефіцієнтів регресії. Відповідно до методу найменших квадратів, вектор s виходить з виразу: $s = (X^T X)^{-1} X^T Y$.

Таблиця 2.2 – Добуток матриці $X^T X$

32	701	133.66	152.65	33.73	12.96	32.82
701	29243	2885.91	2895.98	641.29	275.05	724.59
133.66	2885.91	665.588	668.3906	133.4397	54.5041	136.5281
152.65	2895.98	668.3906	805.4299	176.1721	67.6097	156.0839
33.73	641.29	133.4397	176.1721	50.5905	15.2722	34.5339
12.96	275.05	54.5041	67.6097	15.2722	6.6438	13.3472
32.82	724.59	136.5281	156.0839	34.5339	13.3472	33.7084

Таблиця 2.3 – $X^T X$ має наступні відповідності:

$\sum n$	$\sum y$	$\sum x_1$	$\sum x_2$	$\sum x_3$	$\sum x_4$	$\sum x_5$	$\sum x_6$
$\sum y$	$\sum y^2$	$\sum x_1 y$	$\sum x_2 y$	$\sum x_3 y$	$\sum x_4 y$	$\sum x_5 y$	$\sum x_6 y$
$\sum x_1$	$\sum y x_1$	$\sum x_1^2$	$\sum x_2 x_1$	$\sum x_3 x_1$	$\sum x_4 x_1$	$\sum x_5 x_1$	$\sum x_6 x_1$
$\sum x_2$	$\sum y x_2$	$\sum x_1 x_2$	$\sum x_2^2$	$\sum x_3 x_2$	$\sum x_4 x_2$	$\sum x_5 x_2$	$\sum x_6 x_2$
$\sum x_3$	$\sum y x_3$	$\sum x_1 x_3$	$\sum x_2 x_3$	$\sum x_3^2$	$\sum x_4 x_3$	$\sum x_5 x_3$	$\sum x_6 x_3$
$\sum x_4$	$\sum y x_4$	$\sum x_1 x_4$	$\sum x_2 x_4$	$\sum x_3 x_4$	$\sum x_4^2$	$\sum x_5 x_4$	$\sum x_6 x_4$
$\sum x_5$	$\sum y x_5$	$\sum x_1 x_5$	$\sum x_2 x_5$	$\sum x_3 x_5$	$\sum x_4 x_5$	$\sum x_5^2$	$\sum x_6 x_5$
$\sum x_6$	$\sum y x_6$	$\sum x_1 x_6$	$\sum x_2 x_6$	$\sum x_3 x_6$	$\sum x_4 x_6$	$\sum x_5 x_6$	$\sum x_6^2$

Таблиця 2.4 – Добуток матриць $X^T Y$

227.64829376
5276.50588508
953.10675686
1076.71982284
236.5917068
91.41063383
233.58381494

Таблиця 2.5 – Обернена матриця $(X^T X)^{-1}$

32.0343	-0.00157	-0.1032	-0.386	-0.1463	2.6769	-29.8611
-0.00157	9.8E-5	-0.000212	0.000846	-5.4E-5	-0.00267	-0.00252
-0.1032	-0.000212	0.01351	-0.01032	0.01411	0.01855	0.07629
-0.386	0.000846	-0.01032	0.03864	-0.02293	-0.1375	0.2985
-0.1463	-5.4E-5	0.01411	-0.02293	0.09987	-0.02838	0.1015
2.6769	-0.00267	0.01855	-0.1375	-0.02838	1.3972	-2.5117
-29.8611	-0.00252	0.07629	0.2985	0.1015	-2.5117	28.3574

Вектор оцінок коефіцієнтів регресії дорівнює:
 $Y(X) = (X^T X)^{-1} (X^T Y) = (5.3318; 0.022; 0.01; 0.08061; -0.07548; -0.7223; 1.2149)$

Таблиця 2.6 – Суми та середні суми вибірок

у та x_1	701	21.906	227.648	7.114	5276.506
у та x_2	133.66	4.177	227.648	7.114	953.107
у та x_3	152.65	4.77	227.648	7.114	1076.72
у та x_4	33.73	1.054	227.648	7.114	236.592
у та x_5	12.96	0.405	227.648	7.114	91.411
у та x_6	32.82	1.026	227.648	7.114	233.584
x_1 та x_2	133.66	4.177	701	21.906	2885.91
x_1 та x_3	152.65	4.77	701	21.906	2895.98
x_1 та x_4	33.73	1.054	701	21.906	641.29
x_1 та x_5	12.96	0.405	701	21.906	275.05
x_1 та x_6	32.82	1.026	701	21.906	724.59
x_2 та x_3	152.65	4.77	133.66	4.177	668.391
x_2 та x_4	33.73	1.054	133.66	4.177	133.44
x_2 та x_5	12.96	0.405	133.66	4.177	54.504
x_2 та x_6	32.82	1.026	133.66	4.177	136.528
x_3 та x_4	33.73	1.054	152.65	4.77	176.172
x_3 та x_5	12.96	0.405	152.65	4.77	67.61
x_3 та x_6	32.82	1.026	152.65	4.77	156.084
x_4 та x_5	12.96	0.405	33.73	1.054	15.272
x_4 та x_6	32.82	1.026	33.73	1.054	34.534
x_5 та x_6	32.82	1.026	12.96	0.405	13.347

Таблиця 2.7 – Дисперсії та середньоквадратичні відхилення

у та x_1	433.96	0.288	20.832
у та x_2	3.353	0.288	1.831
у та x_3	2.414	0.288	1.554
у та x_4	0.47	0.288	0.685
у та x_5	0.0436	0.288	0.209
у та x_6	0.00148	0.288	0.0385
x_1 та x_2	3.353	433.96	1.831
x_1 та x_3	2.414	433.96	1.554
x_1 та x_4	0.47	433.96	0.685
x_1 та x_5	0.0436	433.96	0.209
x_1 та x_6	0.00148	433.96	0.0385
x_2 та x_3	2.414	3.353	1.554
x_2 та x_4	0.47	3.353	0.685
x_2 та x_5	0.0436	3.353	0.209
x_2 та x_6	0.00148	3.353	0.0385
x_3 та x_4	0.47	2.414	0.685
x_3 та x_5	0.0436	2.414	0.209
x_3 та x_6	0.00148	2.414	0.0385
x_4 та x_5	0.0436	0.47	0.209
x_4 та x_6	0.00148	0.47	0.0385
x_5 та x_6	0.00148	0.0436	0.0385

Таблиця 2.8 – Матриця парних коефіцієнтів кореляції R :

	DCOUNT x_1	DEXP x_2	TCLOC x_3	TDELTA x_4	TELINE x_5	TFUNC x_6
DCOUNT x_1	1	0,80885377	0,07143	-0,3458135	-0,28553	-0,2193
DEXP x_2	0,8088537	1	-0,0344	-0,4325747	-0,2136	-0,06362
TCLOC x_3	0,0714383	-0,03447114	1	0,338204	-0,18537	0,03038
TDELTA x_4	-0,345813	-0,43257466	0,33820	1	0,44804	0,55744
TELINE x_5	-0,285526	-0,21360163	-0,1853	0,4480408	1	0,35186
TFUNC x_6	-0,219300	-0,06362117	0,03038	0,55744137	0,351865	1

Таблиця 2.9 – Матриця міжфакторної кореляції

	DCOUNT x_1	DEXP x_2	TCLOC x_3	TDELTA x_4	TELINE x_5	TFUNC x_6
DCOUNT x_1	1	-0,03447	-0,4325747	-0,2136	-0,06362	0,219349
DEXP x_2	-0,03447114	1	0,3382041 2	-0,18537	0,030388	-0,24698
TCLOC x_3	-0,43257466	0,338204	1	0,448041	0,557441	-0,24972
TDELTA x_4	-0,21360163	-0,18537	0,4480408	1	0,351865	-0,07159
TELINE x_5	-0,06362117	0,030388	0,5574413 7	0,351865	1	0,214305
TFUNC x_6	0,21934894 4	-0,24698	-0,2497173	-0,07159	0,214305	1

Таблиця 2.10 – Обернена матриця кореляції

1,7	0,3	1,0	0,2	-0,6	-0,2
0,3	1,7	-0,7	0,9	-0,1	0,4
1,0	-0,7	3,3	-0,2	-2,0	0,9
0,2	0,9	-0,2	2,0	-0,9	0,9
-0,6	-0,1	-2,0	-0,9	2,9	-1,5
-0,2	0,4	0,9	0,9	-1,5	2,2

Значення діагональних елементів менше значення критерію Фішера $F_{kp}=3,81456902$, що показує відсутність мультиколінеарності, але потребує більш детального дослідження.

Значення визначника $\det(X^T X) = 788527187,3 \geq 0$. Виходячи з того що чим ближче до 0 визначник матриці міжфакторної кореляції, тим сильніше мультиколінеарність факторів і ненадійніше результати множинної регресії з цим проблеми відсутні.

Якщо в матриці коефіцієнт кореляції $r_{xjxi} > 0.7$, то в даній моделі множинної регресії існує мультиколінеарність. У нашому випадку все парні коефіцієнти кореляції $|r| < 0.7$, що говорить про відсутність мультиколінеарності факторів.

Аналіз першого рядка цієї матриці дозволяє зробити відбір факторних ознак, які можуть бути включені в модель множинної кореляційної залежності. Факторні ознаки, у яких $|r_{yxi}| < 0.5$ виключають з моделі. Можна дати наступну якісну інтерпретацію можливих значень коефіцієнта кореляції: якщо $|r| > 0.3$ – зв'язок практично відсутній; $0.3 \leq |r| \leq 0.7$ – зв'язок середня; $0.7 \leq |r| \leq 0.9$ – зв'язок сильна; $|r| > 0.9$ – зв'язок вельми сильний.

Перевіримо значущість отриманих парних коефіцієнтів кореляції за допомогою t-критерію Стюдента. Коефіцієнти, для яких значення t-статистики по модулю більше знайденого критичного значення, вважаються значущими. Критерій Стюдента(0,05;32-6-1;6-1)=2,063898562.

t-критерій Стюдента для змінних: $t_{x1}=3,365824$; $t_{x2}=3,035229$; $t_{x3}=10,69638$; $t_{x4}=4,515479$; $t_{x5}=8,729906$; $t_{x6}=5,509186$.

Таким чином, зв'язок між y , x_i є суттєвим. Більш об'єктивну характеристику щільності зв'язку дають приватні коефіцієнти кореляції, що вимірюють вплив на результат x_i при незмінному рівні інших факторів.

Колінеарність – залежність між факторами. В якості критерію мультиколінеарності може бути прийнято дотримання наступних нерівностей:

$$r(x_i y) > r(x_k x_j); r(x_k y) > r(x_k x_j).$$

Для відбору найбільш значущих чинників x_i враховуються такі умови:

- зв'язок між результативною ознакою і факторними повинна бути вище міжфакторного зв'язку;
- зв'язок між факторами повинна бути не більше 0.7. Якщо в матриці є міжфакторний коефіцієнт кореляції $r_{x_j x_i} > 0.7$, то в даній моделі множинної регресії існує мультиколінеарності;
- при високому міжфакторному зв'язку для ознаки відбираються фактори з меншим коефіцієнтом кореляції між ними.

У нашому випадку все парні коефіцієнти кореляції $|r| < 0.7$, що говорить про відсутність мультиколінеарності факторів.

Перевіримо змінні на мультиколінеарність за критерієм "хі-квадрат"):

$$\chi^2 = -[32-1-(2*6+5)/6] \ln(0.0615) = 78.56$$

де $m = 6$ – кількість факторів, $n = 32$ – кількість спостережень, $\det[R]$ – визначник матриці парних коефіцієнтів кореляції R . Порівнюємо його з табличним значенням при $v = m / 2 (m-1) = 15$ ступенях свободи і рівні значущості α . Якщо $\chi^2 > \chi^2_{табл}$, то в векторі факторів присутня мультиколінеарність.

$$\chi^2_{табл}(15; 0.05) = 24.99579 < 78.56 \text{ мультиколінеарність відсутня.}$$

Перевіримо змінні на мультиколінеарність за критерієм Фішера.

Таблиця 2.12 – Обернена матриця (до табл. 3.6) R^{-1} :

3.5035	-2.9898	-0.1196	-0.8172	0.3377	0.9838	-0.305
-2.9898	3.9066	-0.1567	1.5732	-0.3129	-1.2111	0.1957
-0.1196	-0.1567	1.4537	-0.9119	0.5554	0.1934	0.1824
-0.8172	1.5732	-0.9119	3.1755	-0.8603	-1.6565	0.6422
0.3377	-0.3129	0.5554	-0.8603	1.5344	-0.03514	0.05628
0.9838	-1.2111	0.1934	-1.6565	-0.03514	2.2253	-0.7314
-0.305	0.1957	0.1824	0.6422	0.05628	-0.7314	1.3703

Розраховані значення критеріїв Фішера порівнюються з табличними при $v_1 = n-m$ і $v_2 = m-1$ ступенях свободи і рівні значущості. Якщо $F_k > F_{табл}$, то k -я змінна мультиколінеарності з іншими.

$$v_1 = 32-6 = 27; v_2 = 6-1 = 6. F_{табл}(27; 6) = 3.81$$

$$F_1 = 11.27 > F_{табл}, \text{ то змінна у мультиколінеарна з іншими.}$$

$$F_2 = 13.08 > F_{табл}, \text{ то змінна } x_1 \text{ мультиколінеарна з іншими}$$

$F_3 = 2.04 \leq F_{табл}$, то змінна x_2 немультіколінеарна з іншими.

$F_4 = 9.79 > F_{табл}$, то змінна x_3 мультіколінеарна з іншими

$F_5 = 2.4 \leq F_{табл}$, то змінна x_4 немультіколінеарна з іншими.

$F_6 = 5.51 > F_{табл}$, то змінна x_5 мультіколінеарна з іншими.

$F_7 = 1.67 \leq F_{табл}$, то змінна x_6 немультіколінеарна з іншими.

Аналіз параметрів рівняння регресії.

Перейдемо до статистичного аналізу отриманого рівняння регресії: перевірці значимості рівняння і його коефіцієнтів, дослідженню абсолютних і відносних недоробок апроксимації. Для незміщеної оцінки дисперсії $\varepsilon = Y - Y(x) = Y - X \cdot s$ виконаємо наступні обчислення.

Таблиця 2.11 – Похибка ε

Y	Y(x)	$\varepsilon = Y - Y(x)$	ε^2	$(Y - Y_{cp})^2$	$ \varepsilon : Y $
6.182	6.815	-0.633	0.401	0.868	0.102
6.182	6.844	-0.662	0.438	0.868	0.107
6.314	6.676	-0.362	0.131	0.641	0.0574
6.358	6.677	-0.319	0.102	0.572	0.0502
6.482	6.678	-0.196	0.0385	0.4	0.0303
6.583	6.631	-0.0479	0.0023	0.282	0.00728
6.605	6.844	-0.239	0.0572	0.259	0.0362
6.749	6.821	-0.0722	0.00521	0.133	0.0107
6.863	7.089	-0.226	0.0512	0.0631	0.033
6.879	6.822	0.0577	0.00333	0.0551	0.00839
6.948	7.155	-0.207	0.0429	0.0276	0.0298
6.972	6.992	-0.0203	0.000412	0.0203	0.00291
7.007	7.117	-0.11	0.0122	0.0115	0.0158
7.015	7.032	-0.0168	0.000281	0.00984	0.00239
7.107	6.689	0.418	0.175	4.3E-5	0.0589
7.124	6.93	0.194	0.0376	0.00011	0.0272
7.153	6.992	0.161	0.0261	0.00152	0.0226
7.169	7.084	0.0856	0.00733	0.00306	0.0119
7.219	7.275	-0.0563	0.00317	0.011	0.0078
7.269	7.456	-0.187	0.035	0.024	0.0257
7.292	7.08	0.211	0.0446	0.0316	0.029
7.301	7.215	0.0857	0.00734	0.035	0.0117
7.375	7.162	0.213	0.0454	0.0682	0.0289
7.383	7.225	0.158	0.0248	0.0722	0.0213
7.428	7.166	0.262	0.0687	0.0984	0.0353
7.441	7.072	0.369	0.136	0.107	0.0496
7.451	7.073	0.379	0.143	0.114	0.0508
7.484	7.073	0.411	0.169	0.137	0.0549
7.707	7.5	0.207	0.0429	0.351	0.0269

Y	Y(x)	$\varepsilon = Y - Y(x)$	ε^2	$(Y - Y_{cp})^2$	$ \varepsilon : Y $
7.902	7.505	0.398	0.158	0.622	0.0503
7.98	7.793	0.187	0.035	0.751	0.0234
8.725	9.16	-0.435	0.19	2.594	0.0499
		всього	2.635	9.23	1.08
			Оцінка дисперсії		

Оцінка дисперсії: $s_e^2 = (Y - Y(X))^T (Y - Y(X)) = 2.635$

Знайдемо оцінку коваріаційної матриці вектору $k = S^2 \cdot (X^T X)^{-1}$

Таблиця 2.13 – Коваріаційна матриця

3.376	-0.000165	-0.0109	-0.0407	-0.0154	0.282	-3.147
-0.000165	1.0E-5	-2.2E-5	8.9E-5	-6.0E-6	-0.000281	-0.000265
-0.0109	-2.2E-5	0.00142	-0.00109	0.00149	0.00196	0.00804
-0.0407	8.9E-5	-0.00109	0.00407	-0.00242	-0.0145	0.0315
-0.0154	-6.0E-6	0.00149	-0.00242	0.0105	-0.00299	0.0107
0.282	-0.000281	0.00196	-0.0145	-0.00299	0.147	-0.265
-3.147	-0.000265	0.00804	0.0315	0.0107	-0.265	2.988

Дисперсії параметрів моделі визначаються співвідношенням $S_i^2 = K_{ii}$, тобто це елементи які знаходяться на головній діагоналі:

Значення множинного коефіцієнта детермінації R^2 і середньої величини відносної похибки $MMRE$ дорівнюють 0,714 та 0,027 відповідно. Значенні відсотка прогнозування $PRED(0.25)=0.897$.

Аналіз мультиколінеарності. $\det(X^T X) = 788527187,3 \geq 0$. Вид мультиколінеарності, при якому факторні змінні пов'язані деякою стохастичною залежністю, називається частковою.

Щільність спільного впливу чинників на результат оцінює індекс множинної кореляції. На відміну від парного коефіцієнта кореляції, який може приймати негативні значення, він приймає значення від 0 до 1. Чим щільніше фактичні значення y_i розташовуються щодо лінії регресії, тим менше залишкова дисперсія і, отже, більше величина $R_y(x_1, \dots, x_m)$, див. (1.1).

Таким чином, при значенні R близькому до 1, рівняння регресії краще описує фактичні дані і фактори сильніше впливають на результат. При

значенні R близькому до 0 рівняння регресії погано описує фактичні дані і фактори чинять слабкий вплив на результат.

Коефіцієнт множинної кореляції: $R = 0.8453$. Зв'язок між ознакою $\ln Y$ і факторами X_i сильний.

Для перевірки рівня значущості використовують F -критерій Фішера.

При цьому обчислюють фактичну (що спостерігається) значення F -критерію, через коефіцієнт детермінації R^2 , розрахований за даними безпосереднього спостереження. За таблицями знаходять критичне значення F -критерію ($F_{кр}$). Для цього задаються рівнем значущості α (зазвичай його беруть рівним 0,05) і двома числами ступенів свободи $k1 = m$ і $k2 = n-m-1$. Маємо: $k1 = 6$ і $k2 = n-m-1 = 32 - 6 - 1 = 25$. Табличне значення при цих ступенях свободи, $F_{кр}(6; 25) = 2.51$, де $\det R = 0.06$ – визначник матриці парної кореляції; $\det R_{11} = 0.22$ – визначник матриці міжфакторної кореляції. Уточнене значення коефіцієнту детермінації:

$$\begin{aligned}\overline{R^2} &= 1 - (1 - R^2) \cdot \frac{n-1}{n-m-1} \\ \overline{R^2} &= 1 - (1 - 0.7146) \cdot \frac{32-1}{32-6-1} = 0.646 \\ F &= \frac{R^2}{1-R^2} \cdot \frac{(n-m-1)}{m} = \frac{0.7146}{1-0.7146} \cdot \frac{32-6-1}{6} = 10.431\end{aligned}$$

Оскільки фактичне значення $F = 10,4 > F_{кр} = 2,51$, то коефіцієнт детермінації статистично значимий і рівняння регресії статистично надійно (тобто коефіцієнти спільно значимі).

Перевірка нормальності розподілу залишкової компоненти X_i – квадрат (критерій згоди Пірсона – χ^2) є об'єктивною оцінкою близькості емпіричних розподілів до теоретичних. Перевіримо гіпотезу про те, що X розподілено по нормальному закону з допомогою критерію згоди Пірсона.

Таблиця 2.14 – Розподіл ймовірностей для перевірки залишкової компоненти

ліва	права	f_i	$x_1 = (x_i - x_{cp})/s$	$x_2 = (x_{i+1} - x_{cp})/s$	$\Phi(x_1)$	$\Phi(x_2)$	$p_i = \Phi(x_2) - \Phi(x_1)$	Частота, $32p_i$	Доданки статистики Пірсона, K_i
-0,508	-0,278	1	-2,855	-1,959	-0,489	-0,437	0,052	1,689	0,281
-0,278	-0,049	2	-1,960	-1,064	-0,437	-0,270	0,166	5,334	2,084
-0,049	0,180	1	-1,065	-0,169	-0,270	0,019	0,290	9,286	0,054
0,180	0,409	10	-0,170	0,725	0,019	0,299	0,279	8,947	0,123
0,409	0,639	4	0,725	1,620	0,299	0,448	0,148	4,764	0,122
0,639	0,868	2	1,620	2,515	0,448	0,492	0,043	1,401	0,255

Теоретична (очікувана) частота дорівнює $\hat{f}_i = f_{pi}$, де $f = 29$.

Ймовірність влучення в i -й інтервал: $p_i = \Phi(x_2) - \Phi(x_1)$

Визначимо границю критичної області. Так як статистика Пірсона вимірює різницю між емпіричним і теоретичним розподілами, то чим більше її бачимо значення $K_{набл}$, тим сильніше аргумент проти основної гіпотези.

$$K_{кр} = \chi^2(6-2-1; 0.05) = 7.81473; K_{набл} = 2.68.$$

Спостережуване значення статистики Пірсона не влучає у критичну область: $K_{набл} < K_{кр}$, тому немає підстав відкидати основну гіпотезу. Справедливо припущення про те, що дані вибірки мають нормальний розподіл.

Таблиця 2.15 – Розподіл частот (гістограма)

	лів	пр	кіль-ть	%
1	-0,508	-0,278	1	0,034483
2	-0,278	-0,049	2	0,068966
3	-0,049	0,180	10	0,344828
4	0,180	0,409	10	0,344828
5	0,409	0,639	4	0,137931
6	0,639	0,868	2	0,068966

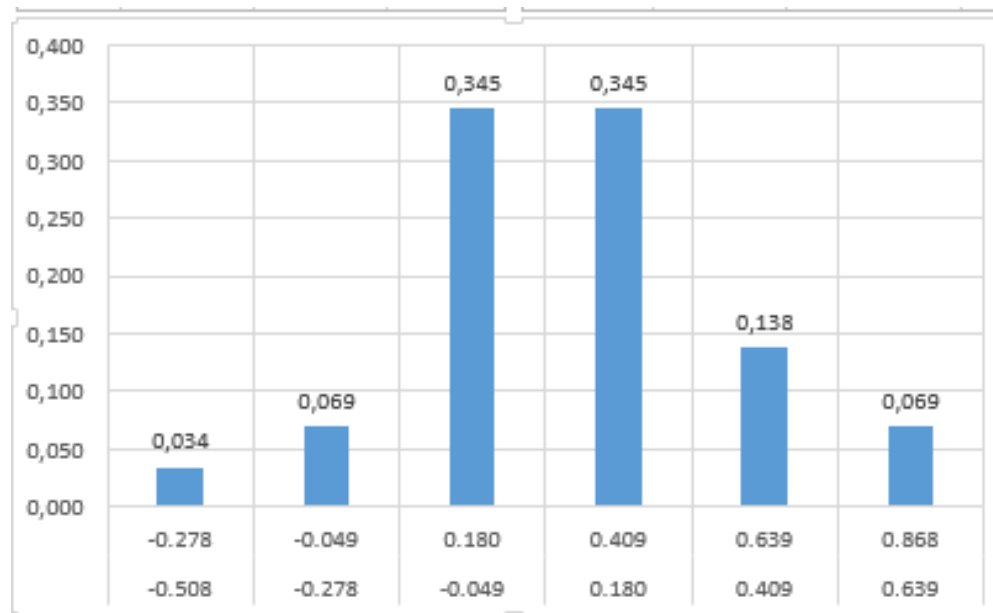


Рисунок 3.2 – Діаграма розподілу частот для залишків

Довірчі та прогнозовані інтервали

Отримаємо нелінійну форму регресійного рівняння:

$$Y = 206,8099 * e^{0.022x_1 + 0.01x_2 + 0.08061x_3 - 0.07548x_4 - 0.7223x_5 + 1.2149x_6}$$

Для отриманого нелінійного рівняння, будемо довірчий та прогнозований інтервали (табл. 3.18).

Таблиця 2.16. Довірчі та прогнозовані інтервали

ліва границя довірчого інтервалу	права границя довірчого інтервалу	ліва границя прогнозного інтервалу	права границя і прогнозного інтервалу
6,83	7,35	6,75	7,48
6,97	7,26	6,79	7,43
6,99	7,24	6,80	7,43
6,99	7,24	6,80	7,43
6,99	7,24	6,80	7,43
7,01	7,22	6,81	7,42
6,97	7,26	6,79	7,43
7,02	7,21	6,81	7,41
7,06	7,17	6,82	7,40
7,01	7,22	6,81	7,42
7,04	7,19	6,82	7,41
6,94	7,28	6,78	7,45
7,02	7,20	6,82	7,41
7,01	7,22	6,81	7,42
6,99	7,24	6,80	7,43
7,03	7,19	6,82	7,41
7,02	7,21	6,81	7,41
7,04	7,19	6,82	7,41
7,01	7,22	6,81	7,42
6,97	7,26	6,79	7,43
7,03	7,19	6,82	7,41
7,06	7,17	6,82	7,40
7,05	7,18	6,82	7,41
7,05	7,17	6,82	7,40
7,03	7,20	6,82	7,41
7,01	7,22	6,81	7,42
7,01	7,22	6,81	7,42
7,01	7,22	6,81	7,42
6,98	7,24	6,80	7,43
6,99	7,23	6,81	7,42

З імовірністю 95% можна гарантувати, що значення Y при необмежено великому числі спостережень не вийде за межі знайдених інтервалів.

Наведемо результати щодо довірчих та прогнозних інтервалів у графічному вигляді на рисунку 2.3.

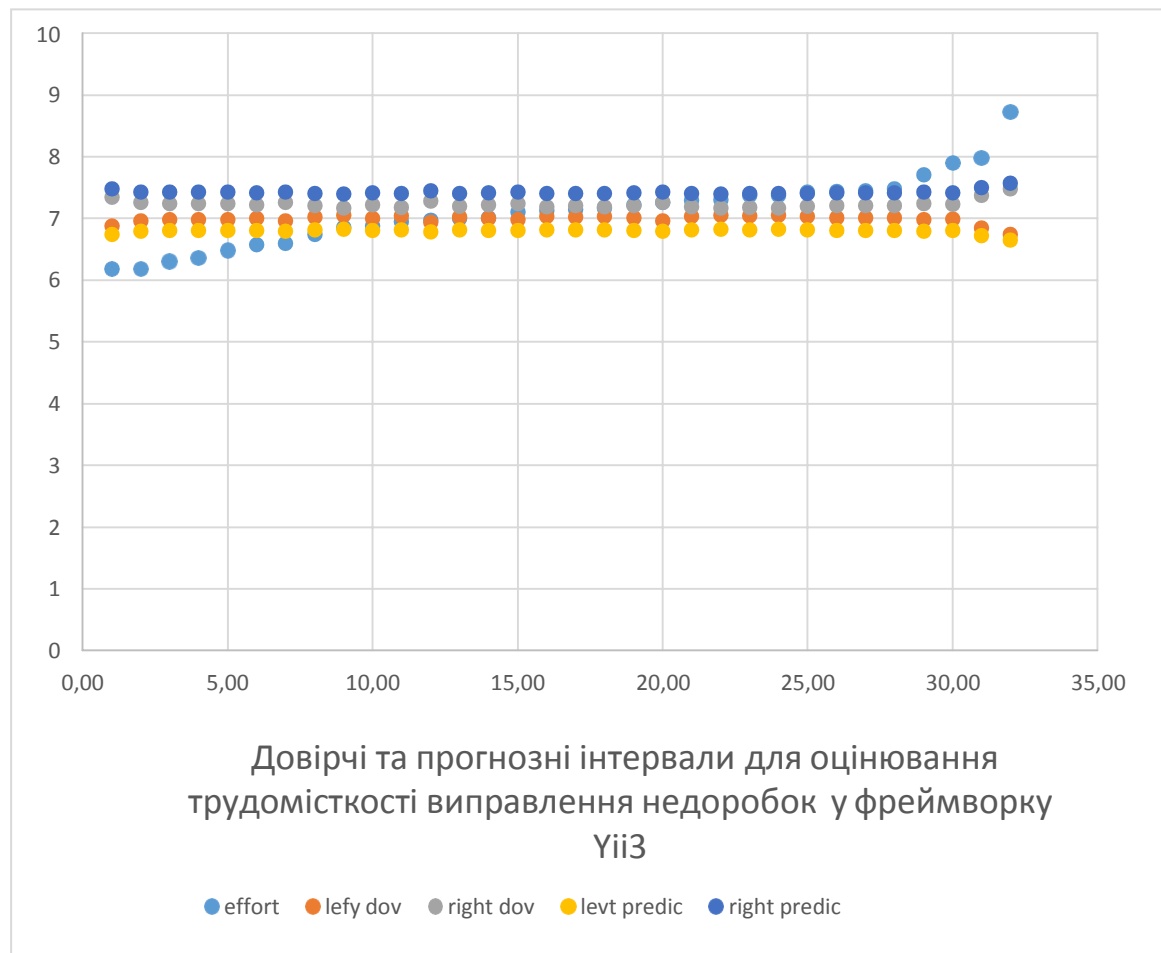


Рисунок 2.3 Довірчі та прогнозні інтервали в залежності від номеру проекту

Висновки. Побудовано математичну модель багатфакторної регресії для оцінювання трудомісткості виправлення недоробок у фреймворку Yii3 параметри якості якої (R^2 і середньої величини відносної похибки $MMRE$ дорівнюють 0,714 та 0,027 відповідно. Значення відсотка прогнозування $PRED(0.25)=0.897$) порівняно з моделлю [1] значно кращі, що дозволяє стверджувати, що першу частину мети роботи про розробку удосконаленої математичної моделі досягнуто.

3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ВИПРАВЛЕННЯ НЕДОРОБОК ВЕБ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ФРЕЙМВОРКА YII3

Для досягнення генеральної мети кваліфікаційної роботи, треба провести декомпозицію вимог до програмного забезпечення для оцінювання трудомісткості виправлення недоробок, що створюються за допомогою фреймворка Yii3.

3.1 Ескізний проект

Використовувати програмне забезпечення може будь-який користувач, без авторизації та реєстрації. Основними варіантами використання системи оцінювання трудомісткості виправлення недоробок, що створюються за допомогою фреймворка Yii3 є введення даних за метриками оцінюваного проекту, введення значення рівня значимості, отримання значень довірчого та прогнозного інтервалу оцінювання трудомісткості виправлення недоробок застосунків, що створюються за допомогою фреймворка Yii3.

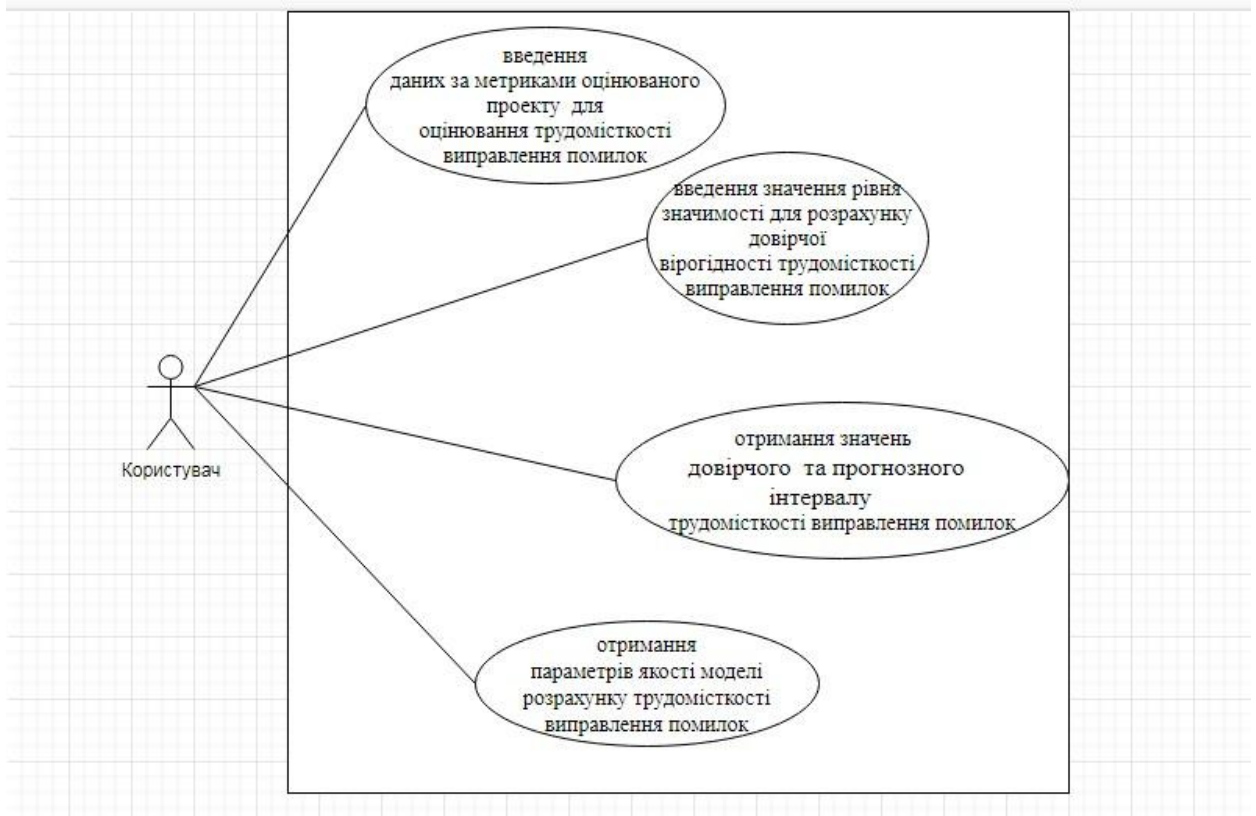


Рисунок 3.1 – Діаграма прецедентів для оцінювання трудомісткості виправлення недоробок застосунків, що створюються за допомогою фреймворка Yii3

Деталізуємо варіанти використання.

Варіант використання «Введення даних за метриками оцінюваного проекту»

Короткий опис: ця функція відкриває для вибору папку для вибору файла формату csv, txt або xls та проводить завантаження даних в якості вхідних параметрів перевіряє заповнення всіх необхідних для розрахунку параметрів та передає керування іншій функції, якщо всі необхідні дані пройшли валідацію.

Передумови: користувач заповнив всі відповідні комірки даними, що пройшли валідацію.

Основний потік подій: за розробленою математичною моделлю програма розраховує параметри, ліву та праву границю довірчого інтервалу

трудомісткості виправлення недоробок.

Альтернативний потік подій: якщо не всі комірки заповнено, а кнопка розрахувати натиснута – вивести повідомлення користувачу про недостатність даних для розрахунку.

Варіант використання «Введення значення рівня значимості для розрахунку довірчої вірогідності»

Короткий опис: ця функція в якості вхідних параметрів отримує рівень значимості, проводить перерахунок границь довірчого та прогнозного інтервалу та довірчої вірогідності значення трудомісткості виправлення недоробок.

Передумови: користувач заповнив відповідну комірку даними, що пройшли валідацію.

Основний потік подій: за розробленою математичною моделлю програма розраховує параметри, ліву та праву границю довірчого інтервалу.

Альтернативний потік подій: якщо не всі комірки заповнено, а кнопка розрахувати натиснута – вивести повідомлення користувачу про недостатність даних для розрахунку.

Варіант використання “Отримання параметрів якості моделі”

Короткий опис: ця функція відкриває діалогове вікно з параметрами для вибору ппку для вибору файл та проводить завантаження в якості вхідних параметрів перевіряє заповнення всіх необхідних для розрахунку параметрів та передає керування іншій функції розрахувати, якщо всі необхідні поля заповнено.

Передумови: користувач заповнив всі відповідні комірки даними, що пройшли валідацію.

Основний потік подій: за розробленою математичною моделлю програма розраховує параметри, ліву та праву границю довірчого інтервалу трудомісткості виправлення недоробок.

Альтернативний потік подій: якщо не всі комірки заповнено, а кнопка розрахувати натиснута – вивести повідомлення користувачу про недостатність даних для розрахунку.

Розробимо сценарії варіантів використання. На рисунку 3.2 наведено діаграму діяльності варіанта використання “Отримання параметрів якості моделі розрахунку трудомісткості виправлення недоробок”



Рисунок 3.2 – Діаграма діяльності варіанта використання “Отримання параметрів якості моделі розрахунку трудомісткості виправлення недоробок, що створюються за допомогою фреймворка YII3 виправлення недоробок”

На наступному рисунку 3.3 наведено порядок дій при настанні варіанту використання «Введення даних за метриками оцінюваного проекту для оцінювання трудомісткості виправлення недоробок»

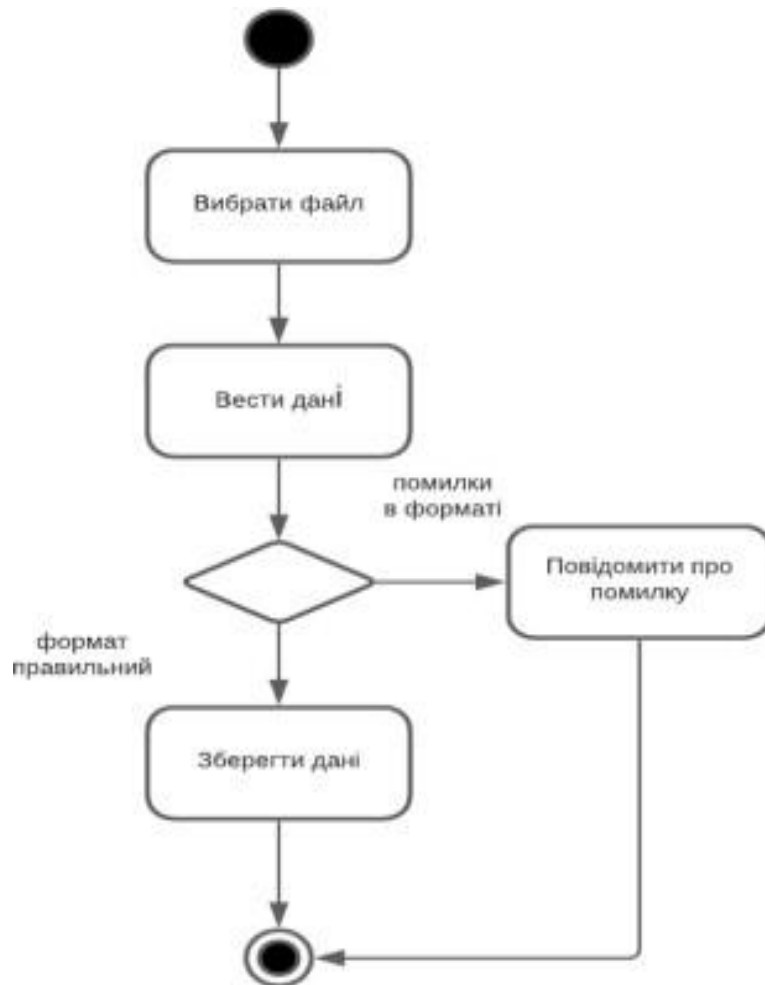


Рисунок 3.3 – Діаграма діяльності варіанта використання “Введення даних за метриками оцінюваного проекту для оцінювання трудомісткості виправлення недоробок що створюються за допомогою фреймворка YII3”

За результати сформуємо концептуальну моделі програмного забезпечення для оцінювання трудомісткості виправлення недоробок яку наведено на рисунку 3.4.

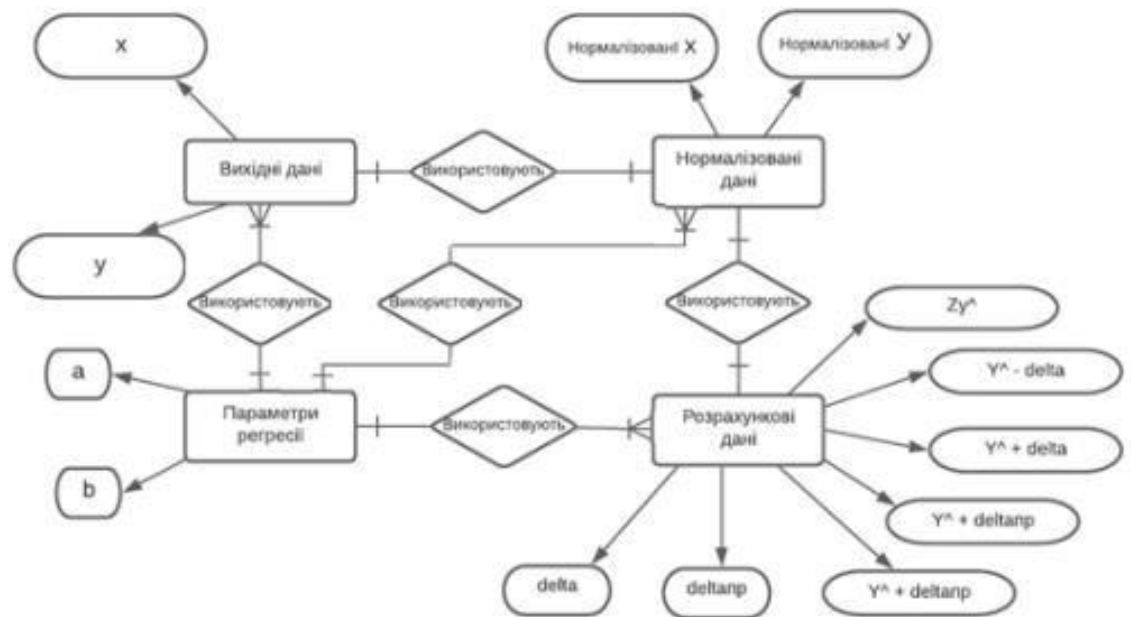


Рисунок 3.4 – Концептуальна модель даних програмного забезпечення для оцінювання трудомісткості виправлення недоробок програмного забезпечення, що створене за допомогою фреймворку Yii3.

Тепер можна перейти до проектування інтерфейсу програмного забезпечення для оцінювання трудомісткості виправлення недоробок програмного забезпечення, що створене за допомогою фреймворку Yii3. Екранна форма наведена на рисунку 3.5.

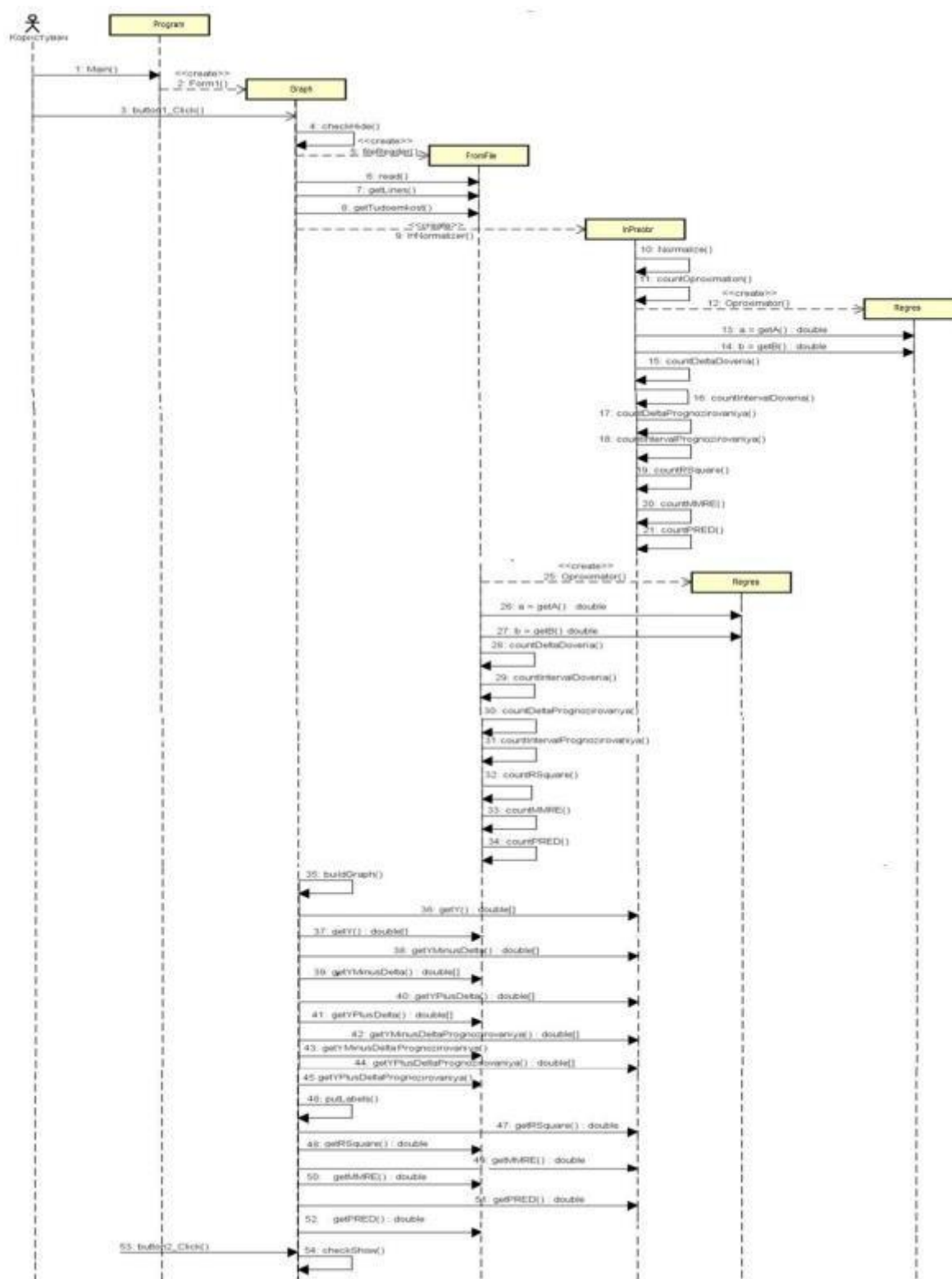


Рисунок 3.5 – Динамічна модель програмного забезпечення

Робочий проект

Програмне забезпечення буде створено на платформі GoDaddy. Інтерфейс програмного забезпечення наведено на рисунках 3.4. Файл доступний для завантаження з прописаними формулами розробленої регресійної моделі для розрахунків буде завантажено до структури сайту і буде доступний користувачу при натисненні кнопки завантажити.



Рисунок 3.4 – Ескіз проекту програмного забезпечення для оцінювання трудомісткості виправлення недоробок, що створюються за допомогою фреймворка YII3

Bugfix Estimator

ГОЛОВНА ПОСЛУГИ ЗВ'ЯЗАТИСЬ З НАМИ

ВІТАЄМО

Bugfix Estimator вітає вас

Please enter the DCOUNT as x1 DEXP as x2 TCLOC as x3 TDELTA as x4 TELINE as x5 and TFUNC as x6

DEXP: експертиза розробника - сума значень рангу всіх тих розробників, які беруть участь у виправленні помилок;

TCLOC загальна кількість рядків вихідного коду;

TCLOC всього змінених рядків коду;

TDELTA загальна кількість змін у вихідних файлах;

TFUNC загальна кількість функцій;

TELINE загальна кількість виконуваних рядків;

TCYCLO загальні показники цикломатичної складності.

Модель: $Y = 206,8099 * e^{0.022 DCOUNT} + 0.01 DEXP + 0.08061 TCLOC - 0.07548 TDELTA - 0.7223 TELINE + 1.2149 TFUNC$

Трудомісткість виправлення недоробок (середнє значення) '___' ліва границя довірчого інтервалу = '___' права границя довірчого інтервалу = '___'

Рисунок 3.5 – Розрахунок трудомісткості виправлення недоробок , що створюються за допомогою фреймворка YII3

Користувачу буде надано можливість ввести параметри проекту для якого йому треба провести оцінювання та завантажити файл з результатами розрахунку який наведений у режимі перегляду формул на рисунках 3.5

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ УПЗ

Актуальність проблеми вибору програмних інструментальних засобів графічного дизайну викликало значний обсяг публікацій з пропозиціями її розв’язання. Головним чином – це вербальні моделі з рекомендаціями, на які характеристики програмних продуктів треба звертати увагу при виборі та їх оцінювання. Вдалим прикладом такої роботи є стаття Ф. Тейксейра [36], де процес вибору розглянуто по кроках від обґрунтування потреби в інструментальних засобах з подальшим визначенням цілей, формулюванням вимог до прийняття власного рішення з вибору. Втім, принаймні два фактори обмежують ефективність такого підходу. По-перше, рекомендації базуються на суб’єктивному, хоча і кваліфікованому, але обмеженому досвіді застосування. По-друге, вибір має бути зроблений за результатами відповідей на велике число питань, які фахівець, що обирає, може розуміти і розв’язувати суб’єктивно через обмеженість досвіду.

В зв’язку з актуальністю даного питання для студентів, що навчаються за напрямком «Програмне забезпечення автоматизованих систем» та з причини відсутності інформаційного забезпечення студентів з даної дисципліни, в рамках даної магістерської роботи, виконується розробка програмного забезпечення для оцінювання якості ПЗ графічного дизайну.

Проаналізувавши вище зазначені проблеми можливо зробити висновок про доцільність розробки моделі якості ПЗ графічного дизайну. В цій роботі, для інформаційної підтримки і зменшення суб’єктивної складової при виборі програмних інструментальних засобів графічного дизайну, запропоновано використовувати методологію, що базується на розробленій моделі якості, специфічній для даного виду ПЗ.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування вартості програмного продукту. Очевидно, що він являє собою дуже специфічний товар з безліччю особливостей.

Створення програмного продукту вимагає одноразових витрат на його розробку, придбання необхідних технічних засобів і поточних витрат на функціонування продукту. Економія від функціонування програмного продукту визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного продукту характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються діючим на момент розрахунку оптовим цінам, тарифам і ставкам заробітної плати.

Витрати на розробку продукту складаються з витрат на зарплату розробника, на зборку ЕОМ, на якій виконується розробка, на експлуатацію цієї ЕОМ, на засоби розробки та витрат на матеріали комплектуючі.

Розробка програмного забезпечення виконується програмістом, місячний оклад якого складає 5000 грн. Додаткова заробітна плата складає 20% від основної. Виходячи з цього, основна і додаткова заробітна плата розроблювана 6000 грн/міс, а вартість сучасного ПК складає 10800 грн. (приведена вартість машини на базі Intel core i3). При вартості кіловат-години електроенергії рівної 0,90 грн., розрахується вартість розробки програми. Витрати на допоміжні матеріали приведені в табл. 5.1.

Таблиця 5.1 – Витрати на допоміжні матеріали

Пункти витрат	Сума (грн.)
Папір	80
Заправлення картриджа до принтера	160
CD-диск	7
Непередбачені витрати	300
Разом матеріали і комплектуючі	547

Вартість програми розрахуємо по формулі:

$$C_{np} = (Z_{zn} + Z_{cz} + Z_{zg} + Z_e) * T + Z_m, де$$

T – тривалість розробки (міс);

Z_{zn} – основна і додаткова заробітна плата обслуговуючого персоналу, грн.:

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

Z_{zg} – загальногосподарські витрати (10% від основної заробітної плати, грн.);

Z_e – витрати на електроенергію;

Z_m – витрати на основні і допоміжні матеріали;

Z_e при споживанні потужності 0,5 кВт, тривалості роботи за місяць, рівної $22*8=176$ год., вартості кіловат-години електроенергії 0,90 грн..

$$Z_e = 176 * 0,90 * 0,5 = 79,2$$

Таблиця 5.2 – Витрати на розробку ПЗ.

Найменування витрат	Одиниця	Кількість
Тривалість розробки	міс.	1,5
Основна і додаткова заробітна	грн.	6000
Відрахування на соціальні	грн.	1140
Загальногосподарські витрати	грн.	250
Витрати на основні і допоміжні	грн.	547
Витрати на електроенергію	грн.	79,2

Відповідно вартість програми:

$$C_{np} = (6000 + 1140 + 250 + 79,2) * 1,5 + 547 = 11750,80 \text{ грн.}$$

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

$$A_{об} = 10800 * 0,6 = 6480 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали (гнучкі диски, папір) визначаються в розмірі 5% вартості основного устаткування:

$$B_m = 10800 * 0,05 = 540 \text{ грн.}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_m = 264,5 * T_z$$

де T_z – середнє місячне завантаження устаткування (близько 4 годин);

264,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПК складе:

$$\Phi_m = 264,5 * 4 = 1058 \text{ годин}$$

Витрати електроенергію $З_e$ при 1013 годинах роботи устаткування в рік складуть

$$Z_e = 1058 * 0,5 * 0,90 = 476,10 \text{ грн.}$$

Експлуатаційні витрати для ПК за рік складатимуть:

$$Z_{зр} = 6480 + 540 + 476,10 = 7496,10 \text{ грн.}$$

Отже, у перший рік витрати на створення й експлуатацію програми складуть:

$$Z_{ce} = 11750,80 + 7496,10 = 19246,90 \text{ грн.}$$

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

Крім багатьох інших негативних ефектів, ручна обробка інформації спричиняє наступні негативні економічні ефекти:

- високі витрати на складування паперових документів (сейфи, шафи, папки й ін.);
- підвищені витрати на канцтовари;
- витрати, пов'язані з роботою виявлення раніше допущених помилок (людський фактор).

До числа основних факторів, що визначають приріст прибутку в зв'язку з упровадженням програми, відносяться:

- підвищення продуктивності праці;
- вивільнення робочого часу.

Крім того, не піддається прямій грошовій оцінці підвищення оперативності керування, якість одержуваних результатів, поліпшення організації праці і т.д.

Обов'язковою умовою визначення економічної ефективності програми для агенства є порівнянність усіх показників у часі, за цінами й іншими нормами, використовуваними для визначення показників, за змістом і колом елементів витрат.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного продукту вивільняє 0,6 працівника (за експертною оцінкою фахівців підприємства).

Зарплата 0,6 працівника в рік складає:

$$(6000 * 12) * 0,6 = 43200,00 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$\mathcal{E}_{\text{год}} = \Delta C_n - E_n * k,$$

де ΔC_n – вивільнені кошти після впровадження програмного продукту (43200,00 грн.) мінус експлуатаційні витрати (7496,10 грн.);

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації - 0,6);

k – одноразові витрати на впровадження продукту (11750,80 грн.).

$$\mathcal{E}_{\text{год}} = 35703,90 - (0,6 * 11750,80) = 28653,42 \text{ грн.}$$

Строк окупності програми розраховується по формулі:

$$T = k / \Delta C_n = 11750,80 / 28653,42 = 0.41 \approx 5 \text{ (місяців)}$$

Отже строк окупності програмного продукту складає приблизно 5 місяців.

Витрати на перший рік витрати на створення й експлуатацію програми складуть: 43200,00 грн. Строк окупності програмного продукту складає приблизно 5 місяців.

5 ОХОРОНА ПРАЦІ

Охорона праці – діюча на підставі відповідних законодавчих та інших нормативних актів система соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, що забезпечують збереження здоров'я і працездатності людини в процесі праці.

Умови праці – це сукупність взаємозв'язаних санітарно-гігієнічних, психофізіологічних, естетичних і соціальних факторів конкретної праці, обумовлених рівнем розвитку продуктивних сил суспільства, які визначають стан середовища та впливають на здоров'я і працездатність людини Система управління охороною праці (СУОП) – це сукупність суб'єкта та об'єкта управління, які на підставі комплексу нормативної документації проводять цілеспрямовану, планомірну діяльність з метою забезпечення здорових, безпечних і високопродуктивних умов праці.

Охорона праці базується на законах та інших нормативно-правових актах, які є головним джерелом зовнішньої інформації, що надходить до СУОП. Створення СУОП здійснюється шляхом послідовного визначення мети і об'єкта управління, завдань і заходів щодо охорони праці, функцій і методів управління, побудови організаційної структури управління, складання нормативно-методичної документації. Організаційно-методичну роботу по управлінню охороною праці, підготовку, правлінських рішень і контроль за їх своєчасною реалізацією здійснює служба охорони праці підприємства, що підпорядкована безпосередньо керівнику підприємства (головному інженеру). Суб'єкт управління аналізує інформацію про стан охорони праці в структурних 59 підрозділах підприємства та приймає рішення спрямовані на приведення фактичних показників охорони праці у відповідність з нормативними. Об'єктом управління в СУОП є діяльність

структурних підрозділів та служб по забезпеченню безпечних і здорових умов праці на робочих місцях. Охорона праці базується на законодавчих, директивних та нормативнотехнічних документах. При управлінні охороною праці не повинні прийматись рішення та проходити заходи, що суперечать чинному законодавству, державним нормативним актам про охорону праці, стандартам безпеки праці, правилам та нормам охорони праці. До основних функцій управління охороною праці належать: - облік показників стану умов і безпеки праці; - аналіз та оцінка стану умов і безпеки праці; - прогнозування і планування робіт, їх фінансування; - організація та координація робіт; - контроль за функціонуванням СУОП. Основні завдання управління охороною праці: - навчання працівників безпечним методам праці та пропаганда питань охорони праці; - забезпечення безпечності технологічних процесів, устаткування, будівель і споруд; - професійної підготовки і підвищення кваліфікації працівників з питань охорони праці, пропаганди безпечних методів праці; - вибору оптимальних режимів праці і відпочинку працівників; - професійного добору виконавців для визначених видів робіт. Правові та організаційні питання: - нормалізація санітарно-гігієнічних умов праці; - забезпечення працівників засобами індивідуального захисту; - забезпечення оптимальних режимів праці та відпочинку; - організація лікувально-профілактичного обслуговування; - удосконалення нормативної бази з питань охорони праці.

Шкідливим фактором називається такий фактор, вплив якого на робітника в певних умовах призводить до захворювання або зниження працездатності. Небезпечний фактор – фактор, дія якого за певних умов може призвести до травм або іншого раптового погіршення здоров'я працівника. Відповідно до ДСТ 12 0.003-74 небезпечні та шкідливі фактори за природою дії поділяються на такі групи: - фізичні; - хімічні; - біологічні; - психофізіологічні. - Фізичні виробничі фактори поділяються на: - рухомі машини та механізми; - рухомі частини виробничого обладнання; - шум; -

вібрацію (загальну і локальну); - інфразвук; - ультразвук; - електромагнітні поля радіочастотного діапазону; - електричні поля промислової частоти; - електростатичні поля; - лазерне випромінювання; - іонізуюче випромінювання; - освітленість; - ультрафіолетове випромінювання; 61 - мікроклімат у виробничому приміщенні (температура повітря, швидкість руху повітря, відносна вологість повітря, інтенсивність - інфрачервоного (теплого) випромінювання); - аероіонізація повітря; - атмосферний тиск (підвищений, знижений). Хімічні виробничі фактори поділяються за характером впливу на організм людини на: - токсичні; - дратівливі; - канцерогенні; - мутагенні і т.д. Хімічні виробничі фактори за способом проникнення в організм людини поділяються на проникаючі через: - органи дихання; - шлунково-кишковий тракт; - шкірні покриви і слизові оболонки. Біологічні виробничі фактори включають біологічні об'єкти: - патогенні мікроорганізми (бактерії, віруси, гриби найпростіші і т.д.); - продукти їх життєдіяльності; - мікроорганізми (рослини, тварини); - природні компоненти організму.

Психофізіологічні виробничі фактори визначаються показниками важкості та напруженості трудового процесу. Основними показниками важкості трудового процесу є: - величина фізичної динамічного навантаження; - разова величина вантажу, що піднімається вручну (маса що піднімається і переміщуваного вантажу вручну); - статичне навантаження; - робоча поза і переміщення в просторі (робоча поза; нахили корпусу; - переміщення в просторі, обумовлені технологічним процесом); - темп роботи (стереотипні робочі рухи); Основними показниками напруженості трудового процесу є: - інтелектуальні навантаження; - напруженість уваги (сенсорні навантаження); - напруженість функцій аналізаторів (сенсорні навантаження); - монотонність (монотонність навантажень); - емоційне напруження (емоційні навантаження); - естетичний дискомфорт; - фізіологічний дискомфорт; - змінність (режим роботи).

До сучасного виробничого освітлення пред'являються високі вимоги як гігієнічного, так і техніко-економічного характеру. Правильно спроектоване і виконане освітлення забезпечує високий рівень працездатності, надає позитивний психологічний вплив на працюючих, сприяє підвищенню продуктивності праці. До систем освітлення висувають наступні вимоги: - відповідність рівня освітленості робочих місць характеру виконуваної зорової роботи; - досить рівномірний розподіл яскравості на робочих поверхнях і в навколишньому просторі; - сталість освітленості в часі; - оптимальна спрямованість випромінюваного освітлювальними приладами світлового потоку; - довговічність, економічність, електро- та пожежобезпечність, естетичність, зручність і простота експлуатації. 63 У тих випадках, коли одного природного освітлення в приміщенні недостатньо, створюють поєднане освітлення. При цьому додаткове штучне освітлення застосовують не тільки в темний, але й у світлий час доби. По конструктивному виконанню штучне освітлення може бути загальними і місцевим. При загальному освітленні всі робочі місця одержують освітлення від загальної освітлювальної установки. Комбіноване освітлення поряд із загальним включає місцеве освітлення, яке зосереджує світловий потік безпосередньо на робочих місцях. Застосування тільки місцевого освітлення неприпустимо, тому що виникає необхідність частоті адаптації зору, створюються глибокі і різкі тіні та інші несприятливі чинники. Для штучного освітлення приміщень використовують люмінесцентні лампи, у яких висока світлова віддача і тривалий термін служби.

Розрахунок штучного освітлення будемо виконувати методом коефіцієнта використання світлового потоку. Розрахуємо потрібну кількість світильників для приміщення з комп'ютерами при загальному рівномірному освітленні.

Довжина приміщення $A=7$ м, ширина приміщення $B=5$ м, висота приміщення $H=3$ м. Висота робочої поверхні $h_p=1$ м. Стеля обладнається

світильниками з люмінесцентними лампами денного світла ЛД40. Коефіцієнт відбиття стелі $S_p=70\%$, стін $S_c=50\%$, робочої поверхні $S_r=10\%$. Напруга мережі 220В. Відстань від стелі до робочої поверхні:

$$H_0 = H - h_r = 3 - 1 = 2 \text{ м.}$$

$$\text{Відстань від стелі до світильника: } h_c = 0,2 * H_0 = 0,2 * 2 = 0,4 \text{ м.}$$

$$\text{Висота підвісу світильника над освітлюваною поверхнею: } h = H_0 - h_c = 2 - 0,4 = 1,6 \text{ м.}$$

$$\text{Висота підвісу світильника над підлогою: } H_p = h + h_r = 1,6 + 1 = 2,6 \text{ м.}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення $L/h=1,5$.

Тоді відстань між центром і світильником: $L = 1,5 * h = 1,5 * 1,6 = 2,4 \text{ м.}$
Потрібна кількість світильників: $N = S / L^2 = (7*5)/2,4^2 = 6,076$ світильників.
Приймаємо 6 світильників (2 ряди по 3 світильники);

Індекс приміщення: $i = (A*B)/(h*(A+B)) = (7*5)/(1,6(7+5)) = 1,82$;
(найближче значення в таблиці 2).

По таблиці коефіцієнтів використання світлового потоку при $i=2$, $S_p=70\%$, $S_c=50\%$, $S_r=10\%$ для світильника типу «ЛД» коефіцієнт використання світлового потоку $\eta=0,59$.

Світловий потік однієї лампи: $\Phi = (E_{\min} * S * K_z * Z) / (N * \eta) = (100 * 35 * 1,5 * 1,1) / (6 * 0,59) = 1631 \text{ лм}$; де K_z – коефіцієнт запасу, який враховує запилення світильників і знос джерел світла в процесі експлуатації; для приміщень, освітлюваних люмінесцентними лампами $K_z=1,5$ за умови чищення світильників не рідше двох разів на рік; Z – коефіцієнт нерівномірності освітлення ($Z=1,1$).

По знайденому значенню світлового потоку кожної лампи з таблиці, при напрузі мережі 220В, визначаємо її потужність, що має світловий потік

2340 лм, найбільш близькій до розрахункового. При цьому фактична освітленість E_f буде: $E_f = E_{min}(\Phi_l / \Phi_p) = 100 * (2340 / 1631) = 143$ лк.

Загальна потужність: $P_{общ} = P_l * N = 40 * 6 = 240$ Вт. 65 6.4

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Основними нормативно-правовими актами України, спрямованими на забезпечення екологічної безпеки на її території, є Конституція, Закони „Про охорону навколишнього природного середовища”, „Про охорону атмосферного повітря”, „Про природно-заповідний фонд України” тощо, Земельний, Водний, Лісовий Кодекси, Кодекс України „Про надра”, постанови Кабінету Міністрів України в законодавстві передбачені права та обов’язки природокористувачів.

Закон України „Про охорону навколишнього природного середовища” [42] є основоположним законодавчим актом, визначає поняття екологічної безпеки та заходи щодо їх забезпечення, екологічні вимоги до розміщення, проектування будівництва, реконструкції, введення в дію підприємств та інших об’єктів, про застосування мінеральних добрив; передбачає заходи щодо охорони навколишнього природного середовища від шкідливого біологічного впливу; шкідливого впливу фізичних факторів та радіоактивного забруднення, від забруднення виробничими, побутовими та іншими відходами.

Даний закон не лише проголошує, але й передбачає систему гарантій екологічної безпеки людини, вносить певну впорядкованість в систему управління в галузі природокористування, він закріплює право громадян України на безпечне для життя навколишнє середовище. Це невід’ємне право реалізується шляхом участі громадян в обговоренні проектів законодавчих актів та інших рішень в галузі охорони навколишнього середовища; участі в розробці та здійсненні заходів щодо охорони природного середовища, раціонального використання природних ресурсів; об’єднання в громадські

природоохоронні організації; отримання повної та достовірної інформації про стан навколишнього природного середовища.

Він надає громадянам України право звертатися до суду з позовом на підприємства, установи щодо відшкодування збитків, заподіяних здоров'ю та майну внаслідок негативного впливу на навколишнє середовище. Метою екології є вивчення загальних закономірностей впливу антропогенної діяльності на навколишнє середовище, зокрема промисловості, сільського господарства, транспорту, комунального господарства тощо.

Охорону навколишнього середовища розглядають зазвичай як комплекс заходів міжнародного, державного, регіонального, локального рівнів, котрі спрямовані на збереження і забезпечення раціонального природокористування, відновлення, охорону та примноження природних ресурсів країни для блага людського суспільства і підтримання біологічної та екологічної рівноваги біосфери. Охорона довкілля включає джерела забруднення та їхній вплив на окремі екосистеми та біосферу з метою запобігання шкідливого впливу.

Найбільш раціональним заходом, спрямованим на зменшення забруднення атмосфери опалювальним обладнанням, є ліквідація пічних систем завдяки розвитку централізованого теплопостачання. При цьому внаслідок підвищення ККД котелень зменшується кількість спалюваного палива, а , отже і забруднення навколишнього середовища. Окрім того при централізованому теплопостачанні в великих котельнях можливе очищення димових газів перед викидом їх в атмосферу.

Позитивне значення для розвитку не тільки централізації теплопостачання, але і каналізації, а також охорони природи може дати застосування колекторного прокладання інженерних комунікацій, - а саме - глибокого закладення. Таке прокладання здійснюється тунельним способом без зняття рослинного шару. Пошкодження рослин, порушення існуючих будівель і дорожнього покриття. А також велике значення для покращення

екологічного стану повітряного басейну є відмова від використання вугілля в міських котельнях та перехід їх на природний газ.

Одним з потужних джерел забруднення міського повітря є автомобільний транспорт. У зв'язку з цим виникла необхідність розробки ряду заходів, що дозволяють запобігти забрудненню біосфери. Одним з таких заходів перехід автомобілів з бензиновими та дизельними двигунами на електромобілі, що діють від під заряджених на станціях батареях – акумуляторів. Електромобілі мають кілька переваг: вони безшумні, бездимні та прості у використанні. Іншим засобом, який сьогодні найчастіше використовується, є встановлення на автомобілях фільтрів чи використання як палива природного газу, котрий в порівнянні з іншими видами палива менше забруднює повітря.

Щодо видалення побутових відходів, то останнім часом в Швеції почали застосовувати пневматичний транспорт для видалення сміття з сміттєпроводів по горизонтальних підземних каналах до станції, що надає послуги декільком будинкам (мікрорайон). В США, Великобританії, Італії та деяких інших країнах застосовується сплав в каналізацію посріблених відходів з квартир, готелів, ресторанів та інших об'єктів. З цією метою біля раковин ставлять механічні подрібнювачі, з котрих подрібнене сміття разом зі стічною водою видаляється в каналізацію, де воно знешкоджується в очисних установках. У нас в країні сміття збирається в контейнери та сміттєприймальні камери.

Для збору та тимчасового схоронності відходів організовують спеціальні площадки з твердим покриттям, який забезпечує запобігти забрудненню ґрунту. Періодичність знешкодження накопичених відходів визначається згідно з діючими санітарними нормами та правилами та залежить від середньоденної температури повітря, при якій відбувається розкладання залишків органічних продуктів.

На ТОВ «BASoft» періодично розробляються щорічні плани заходів з охорони навколишнього середовища та раціональному використанню

природних ресурсів. Основними заходами, передбаченими підприємством на 2019 рік, є наступні:

1. Охорона земельних ресурсів та зелених насаджень:

- утримання в належному стані зелених зон та прилеглих територій;

2. Охорона водного басейну:

- забезпечення регулярного лабораторного контролю за якістю води артезіанської свердловини;

- не допущення скидання стічних вод в міську каналізацію з підвищеним рівнем ГДК, забезпечення контролю якості стічних вод в міську каналізацію;

3. Охорона повітряного басейну:

- здійснення контролю за дотриманням нормативів викидів речовин, що забруднюють довкілля, на стаціонарних джерелах викидів відповідно графіку;

- проведення режимно-налагоджувальних робіт відповідно графіку;

- своєчасна розробка документації з обґрунтуванням обсягів викидів забруднюючих речовин в атмосферне повітря.

4. Промислово-побутові відходи:

- укладання договорів на утилізацію відпрацьованих люмінесцентних ламп, відходів, відпрацьованих автомобільних шин, акумуляторних батарей;

- контроль за спаленням сміття й листя, вирубкою дерев та кущів.

1. Природоохоронна діяльність в країні регулюється рядом нормативно-правових актів, таких як Закон „Про охорону навколишнього природного середовища”, Лісовий, Водний, Земельний Кодекс тощо.

2. Підприємство, яким є ТОВ «BASoft» в процесі своєї діяльності здійснює негативний вплив на екологію міста. До основних факторів, які забруднюють довкілля внаслідок діяльності даного підприємства є:

- скидання у водні об'єкти неочищених чи недостатньо очищених стічних вод, а також поверхневих стоків з міської території;

- викиди в атмосферне повітря продуктів згорання, диму, пилу котелень;

3. Щороку на підприємстві розробляються плани заходів з охорони довкілля та раціональному використанню природних ресурсів. Відповідно до таких планів, природоохоронні заходи підприємства об'єднуються у чотири групи: охорона земельних ресурсів та зелених насаджень, охорона водного басейну, охорона повітряного басейну, промислово-побутові відходи. Кожна з цих груп передбачає проведення ряду робіт з поліпшення стану довкілля.

ВИСНОВКИ

В результаті дослідження було підвищено достовірність оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

Для досягнення поставленої мети вирішено такі завдання:

1) проаналізовано існуючі алгоритми та методи оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3, сформулювати постановку задачі;

2) удосконалено математичну модель оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3;

3) розроблено програмне забезпечення для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IST TECHNICAL REPORT IST TECHNICAL REPORT IST-TR-2009-04-24, APRIL 2009 Program File Bug Fix Effort Estimation Using Machine Learning Methods Syed Nadeem Ahsan Javed Ferzund Franz Wotawa
2. Приходько С.Б. Приходько Н.В. Книрик К.О. ТРИФАКТОРНЕ НЕЛІНІЙНЕ РЕГРЕСІЙНЕ РІВНЯННЯ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ У ФАЗІ ПЛАНУВАННЯ // Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки Том 30 (69) Ч. 1 № 5 2019.
http://www.tech.vernadskyjournals.in.ua/journals/2019/5_2019/part_1/27.pdf
3. Boehm B.W., Abts, C., & Brown, A. W. et al. (2000). Software Cost Estimation with COCOMO II. Upper Saddle River, NJ: Prentice Hall PTR.
4. С. Архипенков Основы метопики COCOMO II citforum.ru/SE/project/arkhipenkov_lectures/13.shtml [режим доступу 15.10.2021]
5. Prykhodko S.B. Constructing the non-linear regression equation to estimate the software size of open source PHP based information systems / S. B. Prykhodko, N. V. Prykhodko, T. G. Smykodub, A. V. Spinov // Проблеми інформаційних технологій. – 2018. – № 1 (023). – С.118-125. – ISSN 1998-7005
6. Prykhodko S. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate NonGaussian Data / S. Prykhodko, N. Prykhodko, L. Makarova, A. Pukhalevych // Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, February 20–24, 2018. – P. 962-965. DOI: 10.1109/TCSET.2018.8336353

7. Tan, H. B. K., Zhao, Y., & Zhang, H. (2006). Estimating LOC for information systems from their conceptual data models. Proceedings from ICSE'06: Software Engineering: the 28th International Conference. (pp. 321-330). Shanghai, China. DOI: 10.1145/1134285.1134331.
8. Tan, H. B.K., Zhao, Y., & Zhang, H. (2009). Conceptual data model-based software size estimation for information systems. Transactions on Software Engineering and Methodology, 19 (2), article no. 4. DOI: 10.1145/1571629.1571630.
9. Prykhodko S.B. Constructing the non-linear regression equation to estimate the software size of open source PHPbased information systems / S. B. Prykhodko, N. V. Prykhodko, T. G. Smykodub, A. V. Spinov // Проблеми інформаційних технологій. – 2018. – № 1 (023). – С.118-125. – ISSN 1998-7005
10. Приходько, С. Б. Інтервальне оцінювання статистичних моментів негаусівських випадкових величин на основі нормалізуючих перетворень [Текст] / С. Б. Приходько // Математичне моделювання: науковий журнал. – Дніпродзержинськ: ДДТУ. – 2011. – № 1 (24). – С. 9–13.
11. Приходько, С. Б. Метод побудови нелінійних рівнянь регресії на основі нормалізуючих перетворень [Текст] : тези доп. міждерж. наук.-методич. конф. / С. Б. Приходько // Проблеми математичного моделювання. – Дніпродзержинськ: ДДТУ, 2012. – С. 31–33.
12. Приходько, С. Б. Розробка нелінійної регресійної моделі тривалості програмних проєктів на основі нормалізуючого перетворення Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Радіоелектронні і комп'ютерні системи. – 2012. – № 4 (56). – С. 90–93.
13. Prykhodko S.B. Constructing the non-linear regression equation to estimate the software size of open source PHPbased information systems / S. B. Prykhodko, N. V. Prykhodko, T. G. Smykodub, A. V. Spinov // Проблеми інформаційних технологій. – 2018. – № 1 (023). – С.118-125. – ISSN 1998-7005

14. Prykhodko, N. V., & Prykhodko, S. B. (2018). Constructing the non-linear regression models on the basis of multivariate normalizing transformations. *Electronic modeling*, 40 (6). 101-110. DOI: 10.15407/emodel.40.06.101.
15. Приходько, С. Б., Приходько, Н. В., Фаріонова, Т. А., Ворона, М. В. (2020). Трьохфакторна нелінійна регресійна модель для оцінювання розміру Php-застосунків з відкритим кодом. *Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки»*, 31, (70), по. 1. 124-131. DOI: 10.32838/2663-5941/2020.1-1/23.
16. Приходько, С.Б., Приходько, Н.В., Ворона, М.В., Беловол, І.О. (2021). Нелінійна регресійна модель для оцінювання розміру web-застосунків, що створюються з використанням фреймворку Laravel. *Інформаційні технології та комп'ютерна інженерія*, 2021.1.
17. Lépin , J.-F. (n.d.). *PhpMetrics: Static analyzys for PHP*. (2021). Retrieved from <https://www.phpmetrics.org/documentation/index.html>. [4] Визуализация качества кода с PhpMetrics. (2021). Retrieved from <https://habr.com/ru/post/254941/>. [5] PHP parser alternatives - PHP code analysis. LibHunt. (2021). Retrieved from <https://php.libhunt.com/php-parser-alternatives>.
18. Prykhodko S. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate NonGaussian Data / S. Prykhodko, N. Prykhodko, L. Makarova, A. Pukhalevych // *Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, Lviv-Slavske, Ukraine, February 20–24, 2018. – P. 962-965. DOI: 10.1109/TCSET.2018.8336353
19. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. -СПб: Питер, 2002. 496 с. ил.
20. Приходько, Н. В. Нелінійне регресійне рівняння для оцінювання розміру програмного забезпечення інформаційних систем з відкритим кодом на PHP = The non-linear regression equation to estimate the software size of open source PHP-based information systems / Н. В. Приходько, С. Б. Приходько //

Вісн. СНУ ім. В. Даля. Сер. Технічні науки. – Сєверодонецьк, 2018. – № 6 (247). – С. 135–139.

21. Приходько, С. Б. Оцінювання розміру PHP-застосунків з відкритим кодом за нелінійними регресійними моделями з різними факторами = Estimating the size of open-source PHP-based apps by nonlinear regression models with various factors / С. Б. Приходько, М. В. Ворона // 36. наук. пр. НУК. – Миколаїв : НУК, 2021. – № 1 (484). – С. 92–98.

22. . Ponomarenko T.V BUILDING THE MATHEMATICAL MODELS OF THE SOFTWARE APPLICATION SIZE ESTIMATION BASED ON THE SMALL DATASETS ., [Текст] / T.V Ponomarenko, Y.M. Mysko, M.V. Beregkov // Матеріали IV Всеукраїнської науково-практичної інтернет-конференції студентів, аспірантів та молодих вчених за тематикою «Сучасні комп'ютерні системи та мережі в управлінні»: збірка наукових праць / Під редакцією Г.О. Райко. – Херсон: Видавництво ФОП Вишемирський В. С., 2021. – 287 с

23. Закони України: «Про охорону праці»; «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві і професійного захворювання, що спричинили втрату працездатності»; «Про забезпечення санітарного та епідемічного благополуччя населення»; Кодекс цивільного захисту України; «Про дорожній рух»; «Про фізичний захист ядерних установок, ядерних матеріалів, радіоактивних відходів, інших джерел іонізуючого випромінювання». [Електронний ресурс]. – Режим доступу: <http://zakon4.rada.gov.ua/laws>

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Назва програми: «Yii3_Innovate».

Область застосування програми обмежена сферою розробки програмного забезпечення з використанням фреймворку Yii3.

1 Підстави для розробки

Підставою для розробки програмного забезпечення є завдання на кваліфікаційну роботу магістра “Регресійна модель для оцінювання трудомісткості виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3, та розробка програми для її реалізації.” затверджена наказом №1239-уч від 13.10.2021.

2 Призначення програми

2.1 Функціональне призначення

Програмне забезпечення призначене для оцінювання виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3.

2.2 Експлуатаційне призначення

Програма може використовуватися для оцінювання необхідного обсягу людино-годин на виправлення недоробок веб застосунків, що створюються з використанням фреймворку Yii3

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу функцій, що виконується

Програмне забезпечення повинно забезпечувати можливість виконання наступних функцій:

- введення інформації про метрики додатку;
- обчислення інтервальних та точкових оцінок недоробок додатку.

3.1.2 Вимоги до організації вхідних та вихідних даних

Введення вхідних даних здійснюється користувачем у відповідні поля веб сторінки.

Вихідні дані виводиться на тій же сторінці після натиснення кнопки обчислити.

3.2 Вимоги до надійності

3.2.1 Вимоги до забезпечення надійності функціонування програми

Вимоги, що забезпечують стійке функціонування програмного забезпечення повинні включати:

- засоби перевірки введеної інформації (програмними засобами, повинні бути розроблені функції верифікації);
- засоби, що забезпечують надійне збереження інформації (за рахунок використання резервних копій даних).

3.2.2 Відмова виконання програмного забезпечення через некоректні дії користувача

Відмова програмного забезпечення можлива внаслідок невірних дій користувача. Задля усунення вказаних відмов потрібно забезпечити верифікацію введення даних.

3.3 Умови експлуатації

Проектоване програмне забезпечення повинно функціонувати при наступних умовах її експлуатації:

- із програмним забезпеченням повинний працювати користувач, що вміє працювати у браузерях chrome, mozilla, firefox.
- умови експлуатації повинні відповідати вимогам до експлуатації апаратної частини ПЕОМ.

3.4 Вимоги до апаратного і програмного забезпечення

Мережне устаткування: Для роботи в локальній мережі потрібно інсталяція на комп'ютері мережної карти і необхідних системних налаштувань. Швидкість передачі, що рекомендується, даних 10 Мб/с і більше.

3.5 Вимоги до документації

До програмної документації повинні входити наступні документи:

- технічне завдання;
- опис програмного забезпечення;
- текст програми;
- інструкція користувача;
- програма і методика випробувань.

Мова документації – українська або англійська.

4 Стадії та етапи розробки

Стадії та етапи розробки наведено в таблиці А.1.

Таблиця А1 – Стадії та етапи розробки програмного забезпечення

Стадія створення ПЗ	Етапи робіт на стадії	Початок	Кінець
Технічне завдання	Визначення призначення розробки та вимог	12.09.2021	22.09.2021
Ескізний проект	Прийняття принципів рішень по структурі ПЗ. Розробка концептуальної моделі даних.	23.09.2021	15.10.2021
Технічний проект	Вибір інструменту мови кодування. Розробка документації. Розробка логічної моделі даних.	16.10.2021	01.11.2021
Робочий проект	Виготовлення компонентів ПЗ. Розробка фізичної моделі даних.	02.11.2021	15.11.2021
Введення в дію	Дослідне функціонування ПЗ з метою перевірки дієздатності, корегування документації	16.11.2021	30.11.2021

5 Порядок прийому та контролю

Для контролю и прийому повинно бути надане інструкція користувача та опис програми.

Перевірка документації програми здійснюється представником замовника з метою зафіксувати факт відповідності (або невідповідності) створеного програмного забезпечення всім пунктам технічної документації.

Порядок контролю и прийому даної розробки здійснюється представником замовника у присутності представника розробника згідно з програмою та методикою випробувань шляхом зіставлення характеристик системи з вимогами контракту.

За результатами прийому складається акт, який підписується представником замовника та представником розробника та утверджується керівниками організації-замовника і організації-розробника.

Якщо програма не пройшла випробування, складається акт о виявлених помилках, що підписується представниками замовника та розробника, та протокол помилок, до якого вносять виявлені помилки. Виконавець зобов'язаний виправити помилки та недоліки у строк, не більш ніж 1 місяць зі дня випробування та звідомити замовника про повторне проведенні перевірки не пізніше 2-х тижнів до початку прийому програмного продукту. Після чого проводиться додаткове тестування тієї частини програми, де були виявлені помилки.

Після виправлення помилок або у разі відсутності таких і при відповідності програми встановленим вимогам підписується акт прийому програмного забезпечення.

ДОДАТОК Б – ТЕКСТ ПРОГРАМИ

```

<!doctype html>
<html class="no-js" lang="">

<head>
  <meta charset="utf-8">
  <title></title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <meta property="og:title" content="">
  <meta property="og:type" content="">
  <meta property="og:url" content="">
  <meta property="og:image" content="">

  <link rel="manifest" href="site.webmanifest">
  <link rel="apple-touch-icon" href="icon.png">
  <!-- Place favicon.ico in the root directory -->

  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/main.css">

  <meta name="theme-color" content="#fafafa">
</head>

<body>

  <!-- Add your site or application content here -->
  <p>This is Bugfixestimator. Please enter the DCOUNT as x1 DEXP as x2 TCLOC as x3 TDELTA as x4
  TELINE as x5 and TFUNC as x6 </p>

  <p>експертизу розробника DEXP: вона отримана шляхом додавання значень рангу всіх тих розробників,
  які беруть участь у виправленні помилки;</p>
  <p>TLOC загальна кількість рядків вихідного коду;</p>
  <p>TCLOC всього змінених рядків коду;</p>
  <p>TDELTA загальна кількість змін у вихідних файлах;</p>
  <p>TFUNC загальна кількість функцій;</p>
  <p>TELINE загальна кількість виконуваних рядків;</p>
  <p>TCYCLO загальні показники цикломатичної складності.</p>

  <form action="<?=$_SERVER['PHP_SELF']?>" method="post" title="Bugfix estimator">
    <p>Модель:  $Y = 206,8099 * e^{0.022}$  <input name="DCOUNT" type="float" placeholder="DCOUNT"> +
    0.01<input name="DEXP" type="float" placeholder="DEXP"> + 0.08061<input name="TCLOC" type="float"
    placeholder="TCLOC"> - 0.07548<input name="TDELTA" type="float" placeholder="TDELTA"> - 0.7223<input
    name="TELINE" type="text" placeholder="TELINE"> + 1.2149 <input name="TFUNC" type="text"
    placeholder="TFUNC"> = <input name="Y" type="text" placeholder="Y"> </p>
    <p><input type="submit" value="Розрахувати"></p>
  </form>
  <?php
    if (isset($_REQUEST['DCOUNT']) && isset($_REQUEST['DEXP']) && isset($_REQUEST['TCLOC'])
    && isset($_REQUEST['TDELTA']) && isset($_REQUEST['TELINE']) && isset($_REQUEST['TFUNC'])){
      $DCOUNT = intval($_REQUEST['DCOUNT']);
      $DEXP = intval($_REQUEST['DEXP']);
      $TCLOC = intval($_REQUEST['TCLOC']);

```

```

$TDELTA = intval($_REQUEST['TDELTA']);
$TELINE = intval($_REQUEST['TELINE']);
$TFUNC = intval($_REQUEST['TFUNC']);

if(empty($DCOUNT) || empty($DEXP) || empty($TCLOC)|| empty($TDELTA)|| empty($TELINE)||
empty($TFUNC)) {
    echo 'Будь-ласка введіть значення метрик';

} else {
    echo 'Трудомісткість = ' . Y = 206,8099* e^0.022*$DCOUNT + 0.01*$DEXP + 0.08061*$TCLOC -
0.07548*$TDELTA - 0.7223* $TELINE + 1.2149*$TFUNC.$y3.' ліва границя довірчого інтервалу = '.$ldi.' ліва
границя довірчого інтервалу = '.$pdi.'; </p>
    <p> Трудомісткість = '.$y3.' ліва границя довірчого інтервалу = '.$ldi.' ліва границя довірчого
інтервалу = '.$pdi.' </p>
    ?>

<script src="js/vendor/modernizr-3.11.2.min.js"></script>
<script src="js/plugins.js"></script>
<script src="js/main.js"></script>

<!-- Google Analytics: change UA-XXXXX-Y to be your site's ID. -->
<script>
    window.ga = function () { ga.q.push(arguments) }; ga.q = []; ga.l = +new Date;
    ga('create', 'UA-XXXXX-Y', 'auto'); ga('set', 'anonymizeIp', true); ga('set', 'transport', 'beacon'); ga('send',
'pageview')
</script>
<script src="https://www.google-analytics.com/analytics.js" async></script>
</body>

</html>

```

ДОДАТОК В – ПРОГРАМА І МЕТОДИКА ВИПРОБОВУВАНЬ

1. Об'єкт випробувань

Назва розроблюваного проекту: «Yii3_Innovate».

Галузь застосування – підприємства софтверної галузі, що застосовують у розробці фреймворк Yii3.

2. Мета випробувань

Мета проведення випробувань оцінки експлуатаційних характеристик програми, перевірка і підтвердження працездатності програми в умовах, максимально наближених до умов реальної експлуатації.

3. Вимоги до програми

Програма має реалізувати функції, описані в технічному завданні, яке наведено в додатку А.

4. Вимоги до програмної документації

Для проведення тестування програми, повинна бути надана наступна документація:

- технічне завдання на розробку програми;
- текст програми;
- опис програми;
- інструкція користувача;
- програма і методика випробувань ПЗ.

5. Склад, порядок та методи випробувань

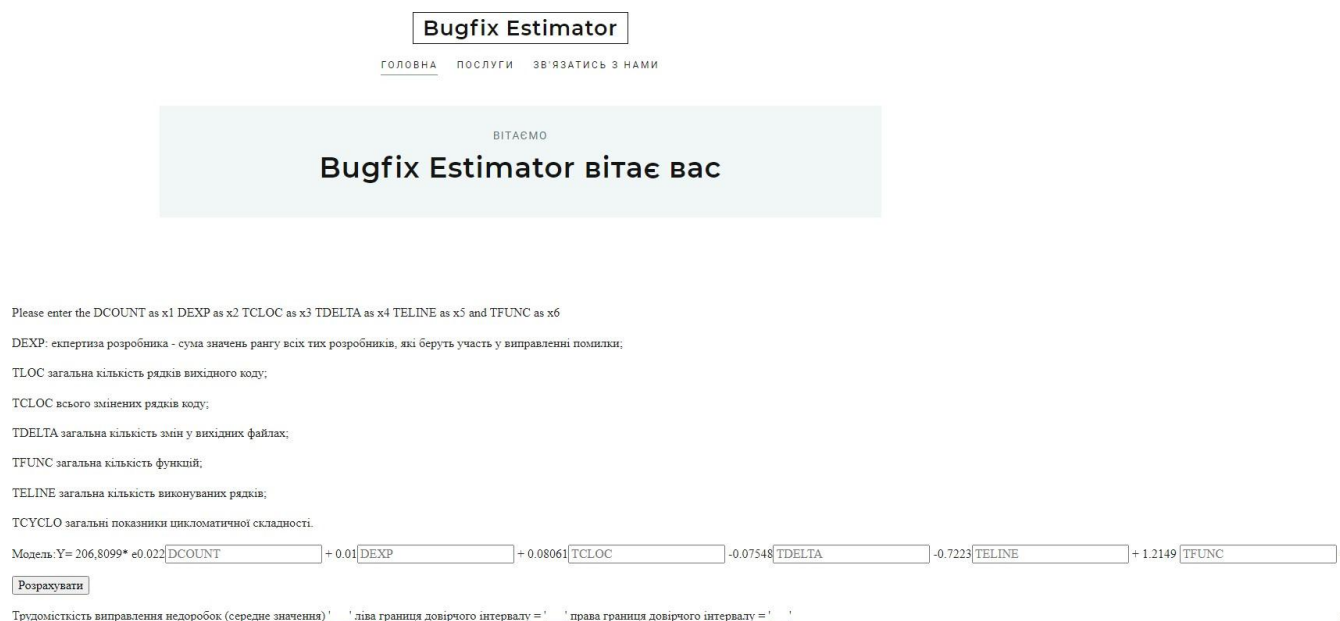
5.1 Перевірка програми на відповідність технічному завданню:

- авторизація з використанням аккаунту github;
- введення інформації про метрики додатку, що оцінюється з метою прогнозування його розмірів;
- перегляд списку додатків, що мають схожий функціонал;

- редагування списку додатків, що мають схожий функціонал;
- обчислення інтервальних та точкових оцінок розміру додатку;
- перегляд звіту з оцінювання розміру майбутнього додатку.

ДОДАТОК Г – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програмне забезпечення «Yii3_Innovate» це динамічна вебсторінка зовнішній вигляд якої наведено на рисунку Г.1.



The screenshot shows the 'Bugfix Estimator' web application. At the top, there is a navigation bar with links: 'ГОЛОВНА' (Home), 'ПОСЛУГИ' (Services), and 'ЗВ'ЯЗАТИСЬ З НАМИ' (Contact Us). Below the navigation bar, a light blue banner displays 'ВІТАЄМО' (Welcome) and 'Bugfix Estimator вітає вас' (Bugfix Estimator welcomes you). The main content area contains instructions in English and Ukrainian. The English instructions ask the user to enter values for DCOUNT, DEXP, TCLOC, TDELTA, TELINE, and TFUNC. The Ukrainian instructions provide definitions for these metrics: DEXP (expert evaluation of bugs), TCLOC (total lines of code), TDELTA (total number of changes), TFUNC (total number of functions), TELINE (total number of lines of code), and TCYCLO (total number of lines of code). Below the instructions, a mathematical model is presented:
$$\text{Model: } Y = 206,8099 \cdot e^{0.022 \cdot \text{DCOUNT}} + 0.01 \cdot \text{DEXP} + 0.08061 \cdot \text{TCLOC} - 0.07548 \cdot \text{TDELTA} - 0.7223 \cdot \text{TELINE} + 1.2149 \cdot \text{TFUNC}$$
 Each variable in the formula is preceded by a text input field. A 'Розрахувати' (Calculate) button is located below the formula. At the bottom, there is a line for the result: 'Трудомісткість виправлення недоробок (середнє значення)' followed by a text input field, and 'ліва границя довірчого інтервалу = ' followed by a text input field, and 'права границя довірчого інтервалу = ' followed by a text input field.

При введенні значень метрик програмне забезпечення виводить середнє значення трудомісткості в людино-годинах, довірчий та прогнозний інтервал

ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА

Якщо вам треба оцінити трудомісткість виправлення помилок розробниками

Просто введіть значення метрик у комірки 1,2,3,4,5,6 та натисніть кнопку розрахувати. Ви отримаєте точкову та інтервальні оцінки трудомісткості у людино годинах як наведено на рисунку Д1.

Bugfix Estimator

ГОЛОВНА ПОСЛУГИ ЗВ'ЯЗАТИСЬ З НАМИ

ВІТАЄМО

Bugfix Estimator вітає вас

Please enter the DCOUNT as x1 DEXP as x2 TCLOC as x3 TDELTA as x4 TELINE as x5 and TFUNC as x6

DEXP: експертиза розробника - сума значень рангу всіх тих розробників, які беруть участь у виправленні помилок;

TCLOC загальна кількість рядків вихідного коду;

TCLOC всього змінених рядків коду;

TDELTA загальна кількість змін у вихідних файлах;

TFUNC загальна кількість функцій;

TELINE загальна кількість виконуваних рядків;

TCYCLO загальні показники цикломатичної складності.

Модель: $Y = 206,8099 * e^{0.022} \text{DCOUNT} + 0.01 \text{DEXP} + 0.08061 \text{TCLOC} - 0.07548 \text{TDELTA} - 0.7223 \text{TELINE} + 1.2149 \text{TFUNC}$

Трудомісткість виправлення недоробок (середнє значення) = ' ' ліва границя довірчого інтервалу = ' ' права границя довірчого інтервалу = ' ' 1