

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

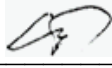
Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

 проф. Приходько С.Б.

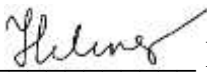
«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

**на тему: Розробка програми для оцінювання кількості дефектів
програмного забезпечення**

Виконав: студентка групи 4151з

 Приходько О.С.

(підпис, ПІБ)

Керівник роботи:

Доц. кафедри ПЗАС, к.т.н., доцент

(посада, науковий ступень вчене звання)

 Макарова Л.М.

(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»



Гарант освітньої програми

доц. Макарова Л.М.

(підпис)

« 08 » 04 2025 р.

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

Студенту гр.4151з Приходько Олені Сергіївні

(Прізвище, ім'я, по батькові)

1. Тема роботи: Розробка програми для оцінювання кількості дефектів програмного забезпечення

Керівник роботи к.т.н., доцент Макарова Л.М.

Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р.

3. Вихідні дані по роботі:

4. Перелік питань, що належать до розробки (найменування розділів) Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів

1. Титульний слайд, 2. Мета та завдання кваліфікаційної роботи, 3. Аналіз практичної задачі інженерії програмного забезпечення, 4. Постановка задачі, 5. Аналіз вимог до програми, 6. Діаграма варіантів використання, 7. Діаграма діяльності, 8. Графічне вікно програми, 9. Емпіричні дані для тестування програми, 10, 11. Результати роботи програми, 12. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проєкту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
11.	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 версій 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.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14.	Подання на кафедру ПЗАС електронних версій 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, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

Студент

(підпис)

Приходько О.С.

(ПІБ)

Керівник роботи

(підпис)

Макарова Л.М.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці програми для оцінювання кількості дефектів програмного забезпечення.

Об'єктом, який розглядається в роботі, є процес розробки програми для оцінювання кількості дефектів програмного забезпечення.

Кваліфікаційна робота викладена на 79 сторінках машинописного тексту, містить 19 рисунків, 2 таблиці, 5 додатків та список використаних джерел з 27 найменувань.

ABSTRACT

The qualification paper is devoted to solving the practical problem of software engineering, characterized by complexity and uncertainty of conditions, namely the development of software to estimate the software defect number.

The object considered in the qualification paper is the process of developing software for estimating the number of software defects.

The qualification paper is presented on 79 pages of typewritten text, it contains 19 figures, 2 tables, 5 appendices, and 27 references.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	10
1.1 Аналіз практичної задачі інженерії програмного забезпечення з розробки програми для оцінювання кількості дефектів програмного забезпечення	10
1.2 Аналіз існуючих інструментів для оцінювання кількості дефектів програмного забезпечення	11
1.3 Аналіз існуючих моделей оцінювання кількості дефектів	16
1.4 Постановка задачі щодо розробки програми для оцінювання кількості дефектів програмного забезпечення	19
2 РОЗРОБКА ПРОЄКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	22
2.1 Розробка ескізного проєкту програми для оцінювання кількості дефектів програмного забезпечення	22
2.2 Розробка технічного проєкту програми для оцінювання кількості дефектів програмного забезпечення	28
2.3 Розробка робочого проєкту програми для оцінювання кількості дефектів програмного забезпечення	32
3 РУЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	39

4 ОХОРОНА ПРАЦІ	44
4.1 Вступ	44
4.2 Аналіз шкідливих і небезпечних факторів в приміщенні з комп'ютерами	49
4.3 Заходи щодо зменшення впливу шкідливих і небезпечних факторів	51
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А Технічне завдання на розробку програми для оцінювання кількості дефектів програмного забезпечення	59
ДОДАТОК Б Текст програми	63
ДОДАТОК В Опис програми	71
ДОДАТОК Г Інструкція користувача	72
ДОДАТОК Д Програма та методика випробувань	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення;
CBO – Coupling Between Object classes);
DIT – Depth of Inheritance Tree;
KLOC – the thousand lines of code;
MMRE – a mean magnitude of relative error;
MRE – a Magnitude of Relative Error;
PRED – percentage of prediction;
QSM – Quantitative Software Management;
SDLC – Software Development LifeCycle;
SDP – Software Defect Prediction;
SLIM – Software Lifecycle Management;
WMC – Weighted Methods for Class.

ВСТУП

Як відомо, одна з практичних задач програмної інженерії полягає в оцінюванні кількості дефектів програмного забезпечення (ПЗ). Це пов'язано з тим, що прогнозування дефектів ПЗ (software defect prediction, SDP) необхідне для управління якістю ПЗ протягом усього життєвого циклу його розробки (software development lifecycle, SDLC). А це в свою чергу призводить до необхідності мати відповідні програмні інструменти, розробка яких також є практичною задачею інженерії ПЗ.

Не зважаючи на існування великої кількості математичних моделей для оцінюванні кількості дефектів, починаючи з відомих рівнянь лінійної регресії Фуміо Акіяма (Fumio Akiyama), запропонованих ще у 1971 році [1], та різноманітних їх програмних реалізацій, роботи у цьому напрямку продовжуються. Підтвердженням актуальності рішення зазначеної задачі може слугувати також і те, що вже з початку 21-ого століття до сьогодні було запропоновано десятки різних моделей прогнозування дефектів ПЗ [2-20], включаючи і регресійні моделі [7-10].

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії ПЗ, що характеризується комплексністю та невизначеністю умов, а саме розробці програми для оцінювання кількості дефектів ПЗ. При цьому невизначеність умов обумовлена в першу чергу статистичною невизначеністю, яка пов'язана з тим, що ми не можемо точно визначити кількість дефектів ПЗ, а можемо лише здійснити їх наближене оцінювання з певною ймовірністю. А відомі програмні інструменти не дозволяють отримувати оцінки кількості дефектів ПЗ із заданою довірчою ймовірністю. А це веде до необхідності розробки оригінальної програми, яка дозволяла би це робити.

Тому метою кваліфікаційної роботи є розробка програми для оцінювання кількості дефектів ПЗ із заданою довірчою ймовірністю. Для досягнення поставленої мети потрібно вирішити такі завдання:

- Здійснити аналіз практичної задачі інженерії програмного забезпечення з розробки програми для оцінювання кількості дефектів ПЗ;
- Здійснити аналіз існуючих інструментів для оцінювання кількості дефектів ПЗ;
- Проаналізувати існуючі регресійні моделі для оцінювання кількості дефектів ПЗ;
- Розробити проєкт програми для оцінювання кількості дефектів ПЗ із заданою довірчою ймовірністю;
- Провести випробування розробленої програми для оцінювання кількості дефектів ПЗ.

Об'єктом, який розглядається в роботі, є процес розробки програми для оцінювання кількості дефектів програмного забезпечення.

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У розділі 1 виконано аналіз практичної задачі інженерії ПЗ з розробки програми для оцінювання кількості дефектів ПЗ, проаналізовано існуючі програмні інструменти та математичні моделі для оцінювання кількості дефектів ПЗ, здійснено постановку задачі на розробку програми для оцінювання кількості дефектів ПЗ.

1.1 Аналіз практичної задачі інженерії програмного забезпечення з розробки програми для оцінювання кількості дефектів програмного забезпечення

Як відомо, одна з практичних задач програмної інженерії полягає в оцінюванні кількості дефектів ПЗ. Це пов'язано з тим, що оцінка кількості дефектів ПЗ необхідна для планування тестування ПЗ (визначення трудомісткості та часу тестування ПЗ), управління якістю ПЗ протягом усього життєвого циклу його розробки. А це в свою чергу призводить до необхідності мати відповідні програмні інструменти, розробка яких також є практичною задачею інженерії ПЗ.

Кваліфікаційна робота присвячена вирішенню саме такої практичної задачі інженерії ПЗ, що характеризується комплексністю та невизначеністю умов, а саме розробці програми для оцінювання кількості дефектів ПЗ. При цьому невизначеність умов обумовлена в першу чергу статистичною невизначеністю, яка пов'язана з тим, що ми не можемо точно визначити кількість

дефектів ПЗ, а можемо лише здійснити їх наближене оцінювання з певною ймовірністю. Вирішення зазначеної практичної задачі інженерії ПЗ обумовлений тим, що, як буде показано у наступному підрозділі, на сьогодні відсутні відповідні програмні інструменти, які би дозволяли отримувати оцінки кількості дефектів ПЗ із заданою довірчою ймовірністю.

1.2 Аналіз існуючих інструментів для оцінювання кількості дефектів програмного забезпечення

В наш час існують різноманітні програмні інструменти в першу чергу для планування розробки та визначення якості ПЗ [21-25], які дозволяють у тому числі здійснювати оцінювання кількості дефектів ПЗ.

Програмний пакет SLIM (Software Lifecycle Management) від компанії QSM (Quantitative Software Management, Inc.) призначений для прийняття бізнес-рішень на кожному етапі життєвого циклу програмного проєкту [21]. Кожен з продуктів SLIM від QSM розроблений для досягнення потужних результатів, як окремий застосунок, так і частина інтегрованого пакету SLIM.

SLIM-Estimate [22] – це застосунок пакету SLIM для визначення різноманітних оцінок, які допомагають робити реалістичні, керовані даними оцінки витрат та розкладів. Переваги SLIM-Estimate:

- Дозволяє оцінювати тривалість, вартість, надійність та ризик вашого програмного проєкту.
- Надає макрорівневі та детальні оцінки за менший час.
- Дозволяє оцінювати на рівні портфоліо за допомогою SLIM-MasterPlan.
- Дозволяє адаптувати свою оцінку до будь-якого процесу проєктування.
- Дозволяє оцінювати обсяг робіт заздалегідь, ще до початку будь-якого планування завдань.

- Оцінює та перевіряє обґрунтованість альтернативних планів проєкту з галузевими даними або власною історією, перш ніж розпочинати будь-яке планування на рівні завдань.

- Перевіряє оцінки за допомогою бази даних QSM, однієї з найбільших галузевих баз даних такого типу.

- Дозволяє узгоджувати розумний графік та бюджет, використовуючи потужні можливості звітності SLIM.

- Дозволяє налаштовувати оцінки одним натисканням кнопки.

- Дозволяє усунути оновлення та подвійне введення за допомогою інтелектуальних компонентів.

- Є інтерфейс з MS Office та веб-сайтом.

Недоліки SLIM-Estimate: не є безкоштовним, не дозволяє отримувати оцінки кількості дефектів ПЗ.

Копія екрану одного з вікон інтерфейсу SLIM-Estimate наведена на рисунку 1.1 [22].

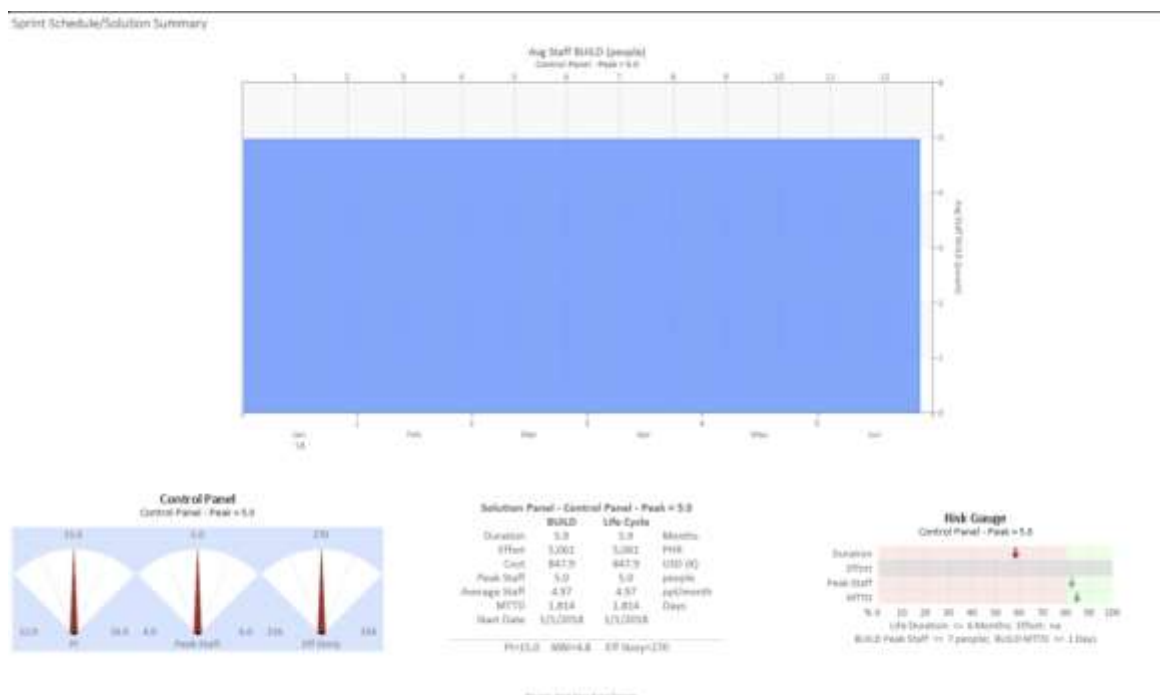


Рисунок 1.1 – Інтерфейс SLIM-Estimate [22]

SLIM-Metrics [23] – це застосунок пакету SLIM для оцінювання різноманітних метрик, включаючи і дефекти ПЗ. SLIM-метрики QSM допомагають з усіма вашими потребами в базовому аналізі та бенчмаркінгу продуктивності програмного проєкту. SLIM-Metrics працює з нашим інструментом репозиторію даних SLIM-DataManager, щоб допомогти вам зберігати історію проєктів розробки ПЗ, оцінювати конкурентні позиції, виявляти вузькі місця, кількісно оцінювати переваги вдосконалення процесів та обґрунтовувати майбутні оцінки проєктів. Дозволяє порівнювати свої дані з галузевими еталонними тенденціями з бази даних QSM, що містить понад 13 000 завершених проєктів розробки ПЗ, або створювати власні еталонні тенденції для використання в SLIM-Metrics, SLIM-Estimate або SLIM-Control.

Переваги SLIM-Metrics:

- Аналізує програмні проєкти після їх завершення.
- Дивіться, які проєкти потребують покращення, а які лідирують.
- Зберігає історію та створюйте власні метрики для вашої організації.
- Створює власні лінії трендів, запити та представлення для аналізу ваших програмних проєктів.
- Показує рентабельність інвестицій вашого проєкту та забезпечує краще розуміння організаційних можливостей.
- Створює та порівнює підмножини даних.
- Створює необмежені кількості діаграм/звітів.
- Використовує поточні галузеві орієнтири.
- Використовує інструменти статистичного аналізу.
- Дозволяє здійснювати перегляд деталей для будь-якої точки даних.
- Дозволяє здійснювати додавання власних нотаток та інших функцій.
- Дозволяє здійснювати створення чудових графіків та таблиць.
- Забезпечує легкий експорт даних.

Недоліки SLIM-Metrics. Не є безкоштовним, не дозволяє отримувати оцінки кількості дефектів ПЗ із заданою довірчою ймовірністю. Дефекти визначаються за шкалою п'ять зірок (рис.1.2)

Five Star Report

Data Set: All Systems

Deviations Calculated From: (SLIM All Systems)

Projects	MB Duration (Months)	MB Effort (Mhrs)	MB Average Staff (People)	PI	MTD (Months)	Defects (UI Defs)	Composite Project Star Rating
Network Unifiling	****	****	****	****	****	****	****
ACC Maintenance	***	****	****	****	****	****	****
Pathto Type	**	****	****	****	****	****	****
ACS Phase 5	****	**	**	**	****	****	****
Benchmark Phase 6	****	****	****	****	****	****	****
Benchmark Phase 1	****	****	****	****	****	****	****
Benefits Package Assessment	****	****	****	****	****	****	****
Benchmark Phase 1/2	****	****	****	****	****	****	****
Benchmark Phase 10	****	****	****	****	****	****	****
Comparison Mgmt	****	****	****	****	****	****	****
Customer Billing	****	****	****	****	****	****	****
Database Entry Handler	****	****	****	****	****	****	****
Database Support	**	**	**	**	**	**	**
Dread Billing	****	****	****	****	**	****	****
Enterprise Analysis	*	**	****	**	**	**	**
Exchange Systems	*	*	*	*	*	*	*
Evolution Systems	****	****	****	****	**	****	****
Fiber Channel Switching	**	*	**	**	*	*	*
Fiber Optic Phase II	****	**	**	**	**	****	****
Fiber Channel Benchmark	**	**	**	**	**	**	**
Fiber Channel Phase 1/2	****	****	**	****	****	**	****
Fiber Channel Phase 10	****	****	****	****	****	**	****
Fiber Optic Testing System	****	*	*	**	****	*	**
File/Port Control System	****	****	****	****	****	**	****
SQL Resolver	****	****	****	****	****	****	****
Graphics Engine	****	****	****	****	****	*	****
HRD Regulations	**	****	****	**	*	****	****
HDS/VHS System	**	**	****	**	****	****	****
Indecor Mgmt	****	****	****	****	****	****	****
ISDN Concepts	**	*	*	*	****	****	****
Line Collection	****	****	****	****	****	****	****
Links and Values	****	****	****	****	**	****	****
Multi-Platform Sys	**	****	****	**	****	*	**
Midwest Sys	**	****	****	****	****	****	****
Multi-Tracking Overview	****	****	****	****	**	**	****
Mutual Fund Mgmt	**	****	****	**	****	**	****
Northwest Sys	**	****	****	**	****	**	****
Online Customer Mgmt	****	****	****	****	****	****	****
Payroll Imaging	****	****	****	****	****	**	****
PDA	**	****	****	**	****	****	****
PE Sys	**	****	****	**	**	**	**
Postal Mgmt	****	****	****	****	*	*	****
Report Writer Engine	**	****	****	**	****	**	****

Star Metrics: 1 Star (1000000) 2 Stars (2000000) 3 Stars (3000000) 4 Stars (4000000) 5 Stars (5000000) 6 Stars (6000000) 7 Stars (7000000) 8 Stars (8000000) 9 Stars (9000000) 10 Stars (10000000)

Рисунок 1.2 – Вікно “Five Star Report” інтерфейсу SLIM-Metrics [23]

ProjectTOP (<https://projecttop.com/en/testing-3/>) – провідне ПЗ для тестування у Фінляндії, яке активно розвивається. ProjectTOP відмінно підходить для приймального тестування ПЗ та бізнес-орієнтованого тестування. ProjectTOP спрощує роботу тестувальників, тому бізнес-тестери сприймають його з ентузіазмом. ProjectTOP дозволяє здійснювати управління дефектами. Копія екрану інтерфейсу ProjectTOP наведена на рисунку 1.3 (<https://b2est037126.smushcdn.com/2037126/wp-content/uploads/2023/09/ProjectTOPin-testitapaus.png?lossy=1&strip=1&webp=1>).

Переваги ProjectTOP. Не потрібне інше ПЗ для комплексного тестування та управління дефектами.

ProjectTOP						
Testirivit						
Testauskoe edistynyt 45%						
	Testausvaihe	Kuvaus	Testausvaihe	Tila	Ohjeet/Toiminto	Testausvaihe
1	Myyntimoduuli (SM01)	Navigoi myyntitilausmoduuliin ja klikkaa "uusi" luodaksesi myyntitilauksen	Myynti			
2		Valitse kotimainen asiakas listasta. Valmistele asiakas, jolla on tuote, jota löytyy varastosta, ja tuote, jota ei löydy varastosta.	Myynti		Myyntitilaus muodostui (muista kirjata tilausnumero)	Tili no. 0132, tuotteet 045, 178 ja 098 lisätty
3		Lisää tilaukseen tuote, jota ei löydy varastosta	Myynti		Järjestelmä ei anna tehdä tilausta, vaan kysyy, tehdäkö jatkuvuuden. Älä tee jatkuvuudesta	Naikakidatuote 047 lisätty
4		Lisää tilaukseen tuote, jota löytyy varastosta	Myynti		Tuote tallentuu myyntitilaukseen	098, 10kg
5		Tallenna tilaus ja aseta se keräilykseen "valmis keräily"	Myynti	✓ X	Tilaukseen siirtyy keräilyjärjestelmään oikalle tilaukseen	Keräilylista 00573 muodostui oikein
6	Varastokeräilymoduuli (WM05)	Tilauksen vapauttaminen keräilyyn ja keräily	Logistiikka			
7		Loggaa varastotilauksen informaatioon, ettei uusia tilauksia voi olla "alusta keräily". Laita tilaus keräilyksi	Logistiikka		Keräilylista näkyy komentitietokoneella	
8		Siirrä keräilyjärjestelmään tilaus	Logistiikka	✗	Järjestelmä ei anna siirtää tilausta järjestelmään, vaan tilaus on okei	Siirretty tilaus on okei, ei siirretty
9		Keräilylistasta määriä	Logistiikka	✓ X	Keräilylista siirrettiin keräilyksi	
10		Varmista, että kuormakirja ja osoitetieto tulostuvat tilaukseen automaattisesti	Logistiikka	✓ X	Varmista, että toimintatiedot ovat oikein	
11		Aseta tuote toimintatietoihin skannaamalla pakkaustarran viivakoodi	Logistiikka	✓ X	Tilaukseen menee "valmis laskutettava" -tilaan	
12	Laskutusmoduuli (LV02)	Tilauksen laskutus	Laskutus			
13		Ettei oikea tilaus ja varmista, että se on "valmis laskutettavaksi" -tilaan. Valitse laskutettavaksi tilaus ja klikkaa "laskuta"	Laskutus	✓ X	Lasku muodostuu oikealle asiakkaalle, kerättyä tuotteita ja oikealla hinnalla	

Рисунок 1.3 – Інтерфейс ProjectTOP

(<https://b2est037126.smushcdn.com/2037126/wp-content/uploads/2023/09/ProjectTOPin-testitapaus.png?lossy=1&strip=1&webp=1>)

Недоліки ProjectTOP. Не є безкоштовним, інтерфейс фінською мовою, не дозволяє отримувати оцінки кількості дефектів ПЗ із заданою довірчою ймовірністю.

ClickUp (<https://clickup.com/>) – це універсальна платформа для Agile-команд, програмний застосунок, який в першу чергу дозволяє легко керувати дорожніми картами продукту, журналами виконання, спринтами, UX-дизайном та ін. ClickUp має:

- Гнучкі інформаційні панелі та звітність про спринт;
- Нативні інтеграції Git та автоматизація без коду;
- Налаштовувані представлення для кожної команди;
- Робочі процеси для Scrum, Kanban тощо.

В цьому є його переваги. Копія екрану інтерфейсу ClickUp наведена на рисунку 1.4 [24].

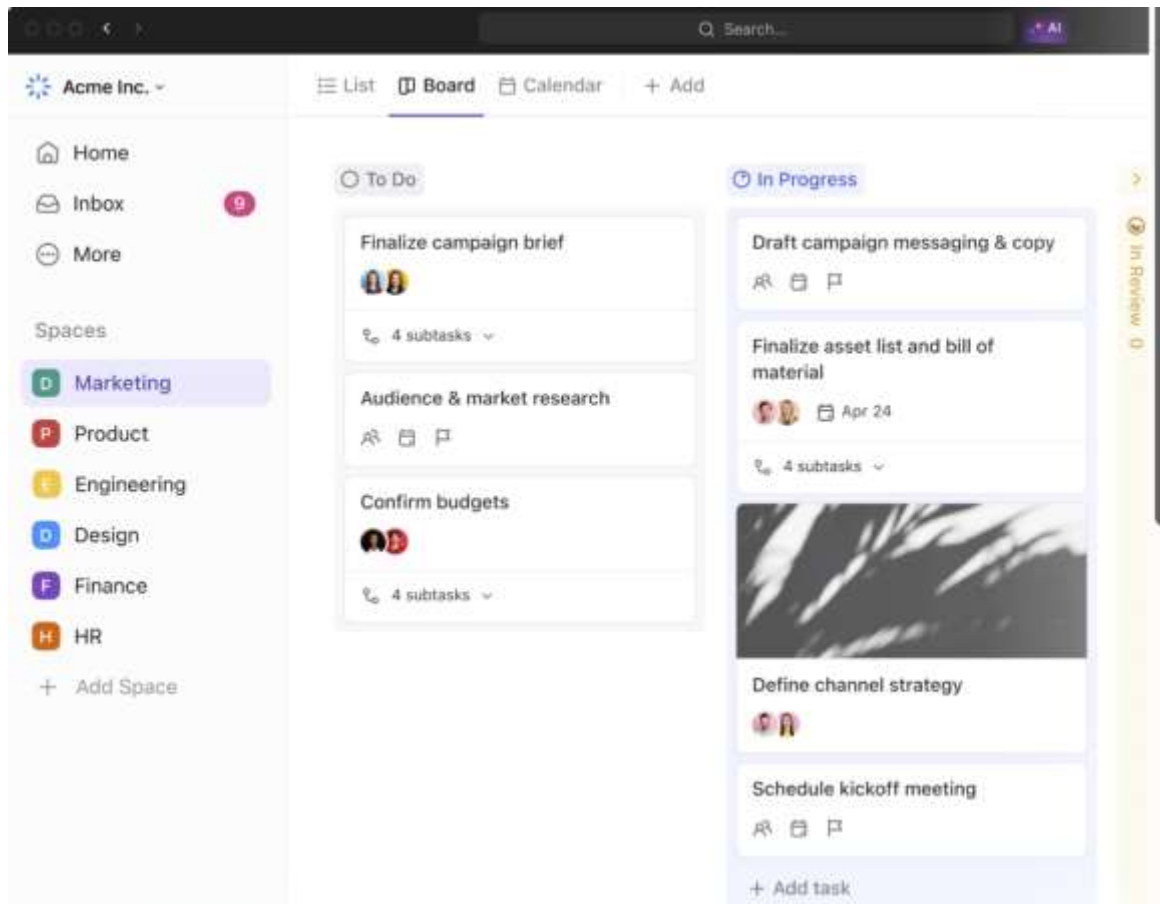


Рисунок 1.4 – Інтерфейс ClickUp [24]

Недоліки ClickUp: не є безкоштовним, не дозволяє отримувати оцінки кількості дефектів ПЗ із заданою довірчою ймовірністю.

1.3 Аналіз існуючих моделей оцінювання кількості дефектів

В наш час існують різноманітні математичні моделі, які дозволяють знаходити оцінки кількості дефектів ПЗ.

Дослідження Фуміо Акіяма (Fumio Akiyama, 1971) показало [1], що лінійні моделі деяких простих показників забезпечують розумні оцінки загальної кількості дефектів D (залежна змінна), яка фактично визначається як сума дефектів, виявлених під час тестування, і дефектів, виявлених протягом двох

місяців після випуску. Акіяма запропонував чотири рівняння регресії, зокрема, таке:

$$D = 4,86 + 0,018L,$$

де L – кількість строк коду (LOC).

Вважається, що Фуміо Акіяма описав перший відомий закон «розміру».

Але наступні дослідження показали, що кількість дефектів залежить не тільки від LOC, а і від інших факторів. Тому в подальшому було розроблено сотні різних моделей прогнозування дефектів [2-20], включаючи і регресійні моделі [7-10]. В [8] автори побудували кілька лінійних рівнянь регресії для оцінювання кількості дефектів ПЗ, що розроблено мовами Java, PHP та .NET, включаючи функціональні дефекти (Functional Defects) та всі дефекти (All Defects), в залежності від семи факторів

$$\begin{aligned} \text{Functional Defects} = & 4.00 - 0.204 \text{ Requirement Error} - 0.631 \text{ Coding Error} + \\ & + 1.90 \text{ KLOC} - 0.140 \text{ Requirement Page} + 0.125 \text{ Design Page} - \\ & - 0.169 \text{ Total Test Cases} + 0.221 \text{ Total Effort Days}; \end{aligned}$$

$$\begin{aligned} \text{All Defects} = & 7.20 - 0.374 \text{ Requirement Error} - 1.37 \text{ Coding Error} + \\ & + 3.07 \text{ KLOC} - 0.245 \text{ Requirement Page} + 0.217 \text{ Design Page} - \\ & - 0.310 \text{ Total Test Cases} + 0.449 \text{ Total Effort Days}, \end{aligned}$$

де Requirement Error – це кількість помилок, які було знайдено у вимогах до ПЗ; Coding Error – це кількість помилок, які було знайдено у процесі кодування; KLOC – це розмір ПЗ у KLOC (the thousand lines of code); Requirement Page – це кількість сторінок документації з вимог до ПЗ; Design Page – це кількість сторінок з проектною документацією; Total Test Cases – це кількість тестових випадків; Total Effort Days – це загальна кількість днів для тестування ПЗ.

Однак, є чотири припущення, які обґрунтовують використання лінійних регресійних рівнянь та моделей, одне з яких – це нормальність розподілу помилок. Але це припущення справедливе лише в окремих випадках [10]. Тому для оцінювання кількості дефектів існує необхідність розробки відповідних нелінійних регресійних рівнянь та моделей [7, 10]. В [7] автори побудували дві нелінійні регресійні моделі для оцінювання кількості дефектів ПЗ, що розроблено мовами Java та C++, в залежності від чотирьох факторів: розміру ПЗ, метрик WMC, CVO та DIT. Використання в цих моделях у якості факторів окрім розміру ПЗ лише метрик, які фактично залежать лише від структури ПЗ, та не врахування інших факторів, які можуть мати вплив на кількість знайдених дефектів, наприклад, кількість тестових випадків, призводить до певних застережень щодо їх застосування.

Тому в [10] автори побудували три нелінійні регресійні моделі для оцінювання кількості дефектів ПЗ, що розроблено мовами Java та PHP, в залежності від трьох факторів: розміру ПЗ, загальні зусилля на етапах до тестування ПЗ та кількість тестових випадків. Нелінійна регресійна модель, яка побудована із застосуванням десяткового логарифму у якості нормалізуючого перетворення є такою [10]:

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1} X_2^{\hat{b}_2} X_3^{\hat{b}_3}, \quad (1.1)$$

де ε – випадкова величина з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$; X_1 – кількість тисяч строк коду (KLOC); X_2 – загальні зусилля (в днях) на етапах до тестування ПЗ; X_3 – кількість тестових випадків; $\hat{b}_0 = -0.3354$; $\hat{b}_1 = 0.5185$; $\hat{b}_2 = -0.4248$; $\hat{b}_3 = 0.6723$.

Зауважимо, що нелінійна регресійна модель (1.1) має прийнятні значення показників якості R^2 , MMRE та PRED(0,25), що дорівнюють 0,6640, 0,2304 та 0,6571 відповідно. Крім того, те, що модель (1.1) побудована із застосуванням

нормалізуючого перетворення у вигляді десяткового логарифму, дозволяє нам знаходити довірчий інтервал вибіркового середнього кількості дефектів ПЗ із заданою довірчою ймовірністю [10]. Враховуючи це та зазначене раніше, в подальшому розробку програми для оцінювання кількості дефектів ПЗ будемо вести із використанням моделі (1.1).

1.4 Постановка задачі щодо розробки програми для оцінювання кількості дефектів програмного забезпечення

Виходячи з аналізу, який було виконано в попередніх підрозділах, сформулюємо постановку задачі. Необхідно створити таку програму, яка би задовольняла наступним вимогам.

Вимоги до програми.

Вимоги функціональних характеристик.

Програма має виконувати такі функції:

1) Вводити з текстового файлу наступні емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

2) Вводити з графічного вікна (інтерфейсу програми) значення наступних величин, за якими потрібно оцінити кількість дефектів ПЗ: довірчу ймовірність, кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

3) Вводити з інтерфейсу програми емпіричне (фактичне) значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

4) Перевіряти правильність вводу початкових (вхідних) даних.

5) Виконувати нормалізацію емпіричних даних за десятковим логарифмом.

6) Знаходити точкову оцінку умовного математичного сподівання (умовне вибіркове середнє) кількості дефектів ПЗ за нелінійною регресією.

7) Обчислювати та виводити значення коефіцієнту детермінації R^2 нелінійної регресії.

8) Обчислювати та виводити значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

9) Знаходити інтервальну оцінку умовного вибіркового середнього (довірчий інтервал нелінійної регресії) кількості дефектів ПЗ.

10) Виводити на екран умовне вибіркове середнє кількості дефектів ПЗ за нелінійною регресією, довірчий інтервал нелінійної регресії кількості дефектів ПЗ, значення коефіцієнту детермінації R^2 , значення відносної похибки MRE (у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ).

11) Виводити у текстовий файл результати оцінювання кількості дефектів ПЗ.

Вхідні дані:

- емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- значення довірчої ймовірності та трьох метрик, за якими потрібно оцінити кількість дефектів ПЗ: кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- емпіричне значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

Вихідні дані:

- значення умовного вибіркового середнього за нелінійною регресією та значення границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

- значення коефіцієнту детермінації;

- значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Більш ретельна постановка задачі у вигляді технічного завдання наведена у додатку А.

2 РОЗРОБКА ПРОЄКТУ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У розділі 2 наведено проєкт програми для оцінювання кількості дефектів ПЗ. В ескізному проєкті виявлені варіанти використання та описані їх сценарії, наведено зовнішню специфікацію програми для оцінювання кількості дефектів ПЗ та стани цієї програми при виконанні. В технічному проєкті спроектовано програму для оцінювання кількості дефектів ПЗ на основі покрокової деталізації з гори до низу, наведено специфікації модулів програми. В робочому проєкті виконано кодування модулів програми, здійснено тестування програми для оцінювання кількості дефектів ПЗ та її випробування.

2.1 Розробка ескізного проєкту програми для оцінювання кількості дефектів програмного забезпечення

В ескізному проєкті були виявлені варіанти використання та описані їх сценарії, наведено зовнішню специфікацію програми для оцінювання кількості дефектів ПЗ та стани цієї програми при виконанні.

В цьому проєкті є лише один актор – користувач програми, який особисто буде застосовувати всі функції програми. Варіанти використання наведені в таблиці 2.1.

Таблиця 2.1 Варіанти використання

Основні актори	Найменування	Формулювання
Користувач	Ввести дані метрик ПЗ	Здійснює введення даних про фактичну кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.
Користувач	Ввести три метрики ПЗ	Вводить наступні метрики, за якими треба оцінити кількість дефектів ПЗ: кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.
Користувач	Ввести довірчу ймовірність	Здійснює введення довірчої ймовірності, для якої визначають довірчий інтервал нелінійної регресії кількості дефектів ПЗ.
Користувач	Нормалізувати дані	Здійснює нормалізацію введених даних
Користувач	Побудувати нелінійне рівняння регресії	Визначає нелінійну регресію для оцінювання середнього значення кількості дефектів ПЗ
Користувач	Отримати довірчий інтервал нелінійної регресії кількості дефектів ПЗ	Визначає із заданою ймовірністю довірчий інтервал нелінійної регресії для умовного вибіркового середнього кількості дефектів ПЗ

Діаграма варіантів використання програми для оцінювання кількості дефектів ПЗ зображена на рисунку 2.1.

Сценарії варіантів використання зображені у вигляді діаграми діяльності на рисунку 2.2.

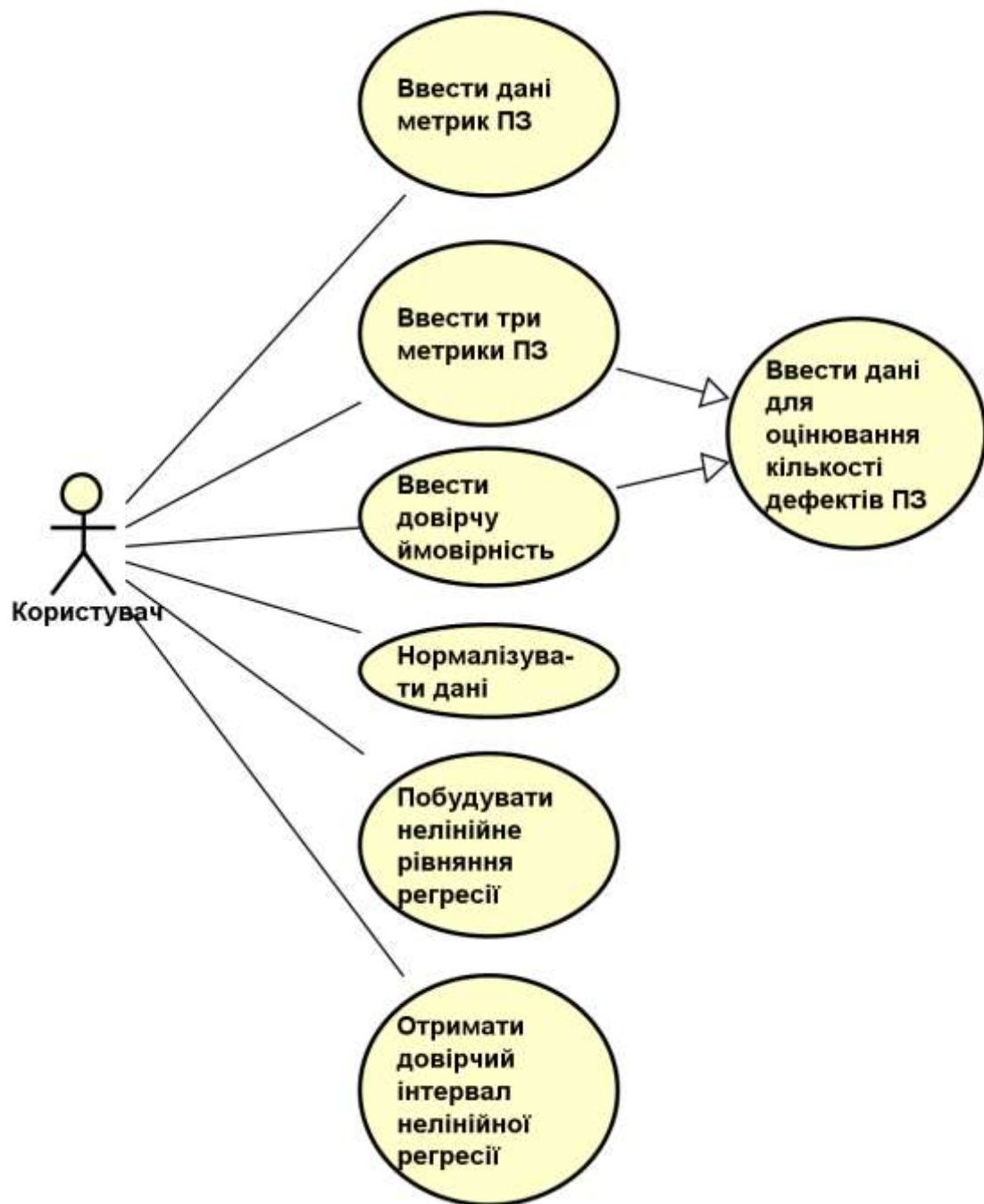


Рисунок 2.1 – Діаграма варіантів використання програми для оцінювання кількості дефектів ПЗ



Рисунок 2.2 – Діаграма діяльності програми

Зовнішня специфікація програми.

Назва програми: SoftDefEst.

Функції, які виконує програма:

1) Здійснювати введення з текстового файлу наступні емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

2) Здійснювати введення з графічного вікна (інтерфейсу програми) значення наступних величин, за якими потрібно оцінити кількість дефектів ПЗ: довірчу ймовірність, кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

3) Здійснювати введення з інтерфейсу програми емпіричне (фактичне) значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

4) Перевіряти правильність вводу початкових (вхідних) даних.

5) Виконувати нормалізацію даних за десятковим логарифмом.

6) Знаходити точкову оцінку умовного математичного сподівання (умовне вибіркове середнє) кількості дефектів ПЗ за нелінійною регресією.

7) Обчислювати та виводити значення коефіцієнту детермінації R^2 нелінійної регресії.

8) Обчислювати та виводити значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

9) Знаходити інтервальну оцінку умовного вибіркового середнього (довірчий інтервал нелінійної регресії) кількості дефектів ПЗ.

10) Виводити на екран умовне вибіркове середнє кількості дефектів ПЗ за нелінійною регресією, її довірчий інтервал, значення коефіцієнту детермінації, значення відносної похибки MRE (у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ).

11) Виводити у текстовий файл результати оцінювання кількості дефектів ПЗ.

Вхідні дані:

- емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- значення довірчої ймовірності та трьох метрик, за якими потрібно оцінити кількість дефектів ПЗ: кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- емпіричне значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

Вихідні дані:

- значення умовного вибіркового середнього за нелінійною регресією та значення границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

- значення коефіцієнту детермінації;

- значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Зовнішні ефекти: поява графічного вікна для вводу даних, обчислення оцінок кількості дефектів ПЗ, вихід з програми (закриття всіх графічних вікон).

Для проєктування програмного інтерфейсу представимо стани програми за допомогою діаграми станів, яка наведена на рисунку 2.3.

На рисунку 2.3 стани позначено наступним чином: 0 – початковий стан, 1 – вікно головного модуля, 2 – вікно візуалізації введених даних, 3 – вікно з результатами оцінювання умовного вибіркового середнього за нелінійною регресією, 4 – вікно з результатами оцінювання границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ, 5 – кінцевий стан.

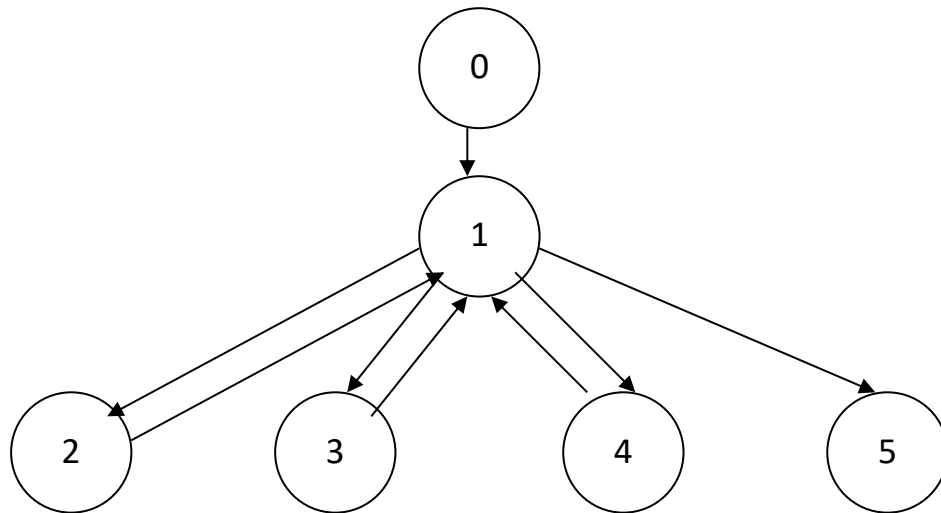


Рисунок 2.3 – Діаграма станів програми для оцінювання кількості дефектів ПЗ

Далі виконуємо розробку технічного проєкту програми для оцінювання кількості дефектів ПЗ.

2.2 Розробка технічного проєкту програми для оцінювання кількості дефектів програмного забезпечення

В технічному проєкті було спроектовано програму для оцінювання кількості дефектів ПЗ на основі покрокової деталізації з гори вниз, виконано специфікації модулів програми, розроблено їх алгоритми.

До програми входять наступні модулі: SoftDefEst (головний модуль), DataInput (модуль для завантаження даних з файлу та їх візуалізації), SoftDefConfIntEst (модуля для визначення нелінійної регресії та довірчого інтервалу нелінійної регресії кількості дефектів ПЗ), W_Close (модуля для закриття вікон). Схема взаємодії модулів програми наведена на рисунку 2.4.

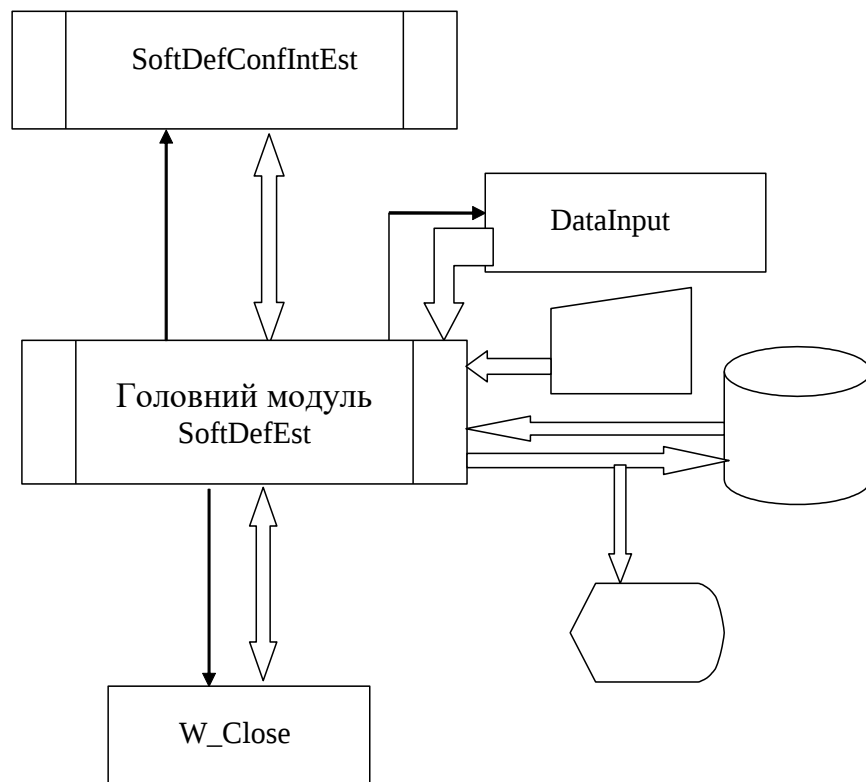


Рисунок 2.4 – Схема взаємодії модулів програми

Специфікація модуля SoftDefEst.

Головний модуль.

Функції, які виконує модуль:

- 1) Відкриття вікна модуля.
- 2) Введення імені файлу з даними.

3) Виклик функції для введення даних з файлу та їх візуалізації у трьох графічних вікнах.

4) Вибір наявності фактичного значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою) та введення цього значення.

5) Виклик функції визначення нелінійної регресії та довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

6) Виклик функції закриття всіх графічних вікон.

Вхідні дані:

- емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- значення довірчої ймовірності та трьох метрик, за якими потрібно оцінити кількість дефектів ПЗ: кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- емпіричне значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

Вихідні дані:

- значення нелінійної регресії та границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

- значення коефіцієнту детермінації;

- значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Зовнішні ефекти: поява графічного вікна для вводу даних, обчислення значень нелінійної регресії та границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ, закриття всіх графічних вікон (завершення роботи програми).

Специфікація модуля DataInput.

Модуль призначений для завантаження даних з файлу та їх візуалізації.

Модуль виконує такі функції:

1) Введення даних з текстового файлу.

2) Створення одномірних та двомірних масивів даних.

3) Відкриття трьох графічних вікон для візуалізації даних, які було введено з текстового файлу.

4) Візуалізацію даних, які було введено з текстового файлу.

Вхідні дані:

- емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

Вихідні дані:

- масиви (одномірні та двомірні) даних, які було введено з текстового файлу;

- кількість точок (рядків) даних, які було введено з текстового файлу;

- три ідентифікатори графічних вікон (для можливості їх подальшого закриття).

Зовнішні ефекти: Поява трьох графічних вікон для візуалізації даних, які було введено з текстового файлу.

Специфікація модуля SoftDefConfIntEst.

Модуль призначений для обчислення значень нелінійної регресії та границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

Функції, які виконує модуль:

1) Обчислює значення нелінійної регресії та границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

2) Обчислює коефіцієнт детермінації.

3) Обчислює значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Вхідні дані:

- масиви емпіричних даних;

- кількість емпіричних даних.

Вихідні дані:

- значення нелінійної регресії, нижньої та верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.
- значення коефіцієнту детермінації;
- значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Зовнішні ефекти: з'являються значення нелінійної регресії, нижньої та верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ, значення коефіцієнту детермінації та значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Специфікація модуля W_Close.

Функції, які виконує модуль:

Закриття всіх графічних вікон.

Вхідні дані:

- ідентифікатори графічних вікон.

Вихідні дані: відсутні.

Зовнішні ефекти: Закриття всіх графічних вікон.

Далі були розроблені алгоритми модулів SoftDefEst та SoftDefConfIntEst.

2.3 Розробка робочого проєкту програми для оцінювання кількості дефектів програмного забезпечення

В робочому проєкті було виконано кодування модулів програми для оцінювання кількості дефектів ПЗ, здійснено тестування та випробування програми.

Спочатку scі-мовою системи моделювання Scilab було виконано кодування модулів за алгоритмами, які були створені в технічному проєкті. Текст програми наведений в додатку Б.

В середовищі Scilab 6.0.0 був сформований sсе-файл розміром 11 КБайт. Підкреслимо, в програмі був застосований динамічний спосіб створення різноманітних компонентів інтерфейсу. Суть цього способу полягає у наступному: при виконанні програми можна створювати і знищувати різноманітні елементи управління (вікна редагування, командні кнопки, радіокнопки та таке інше), а їх властивостям присвоювати відповідні значення. В програмі були використані такі інтерфейсні компоненти: вікно редагування, командні кнопки та радіокнопки. Графічне вікно програми із вказаними компонентами показано на рисунку 2.5.

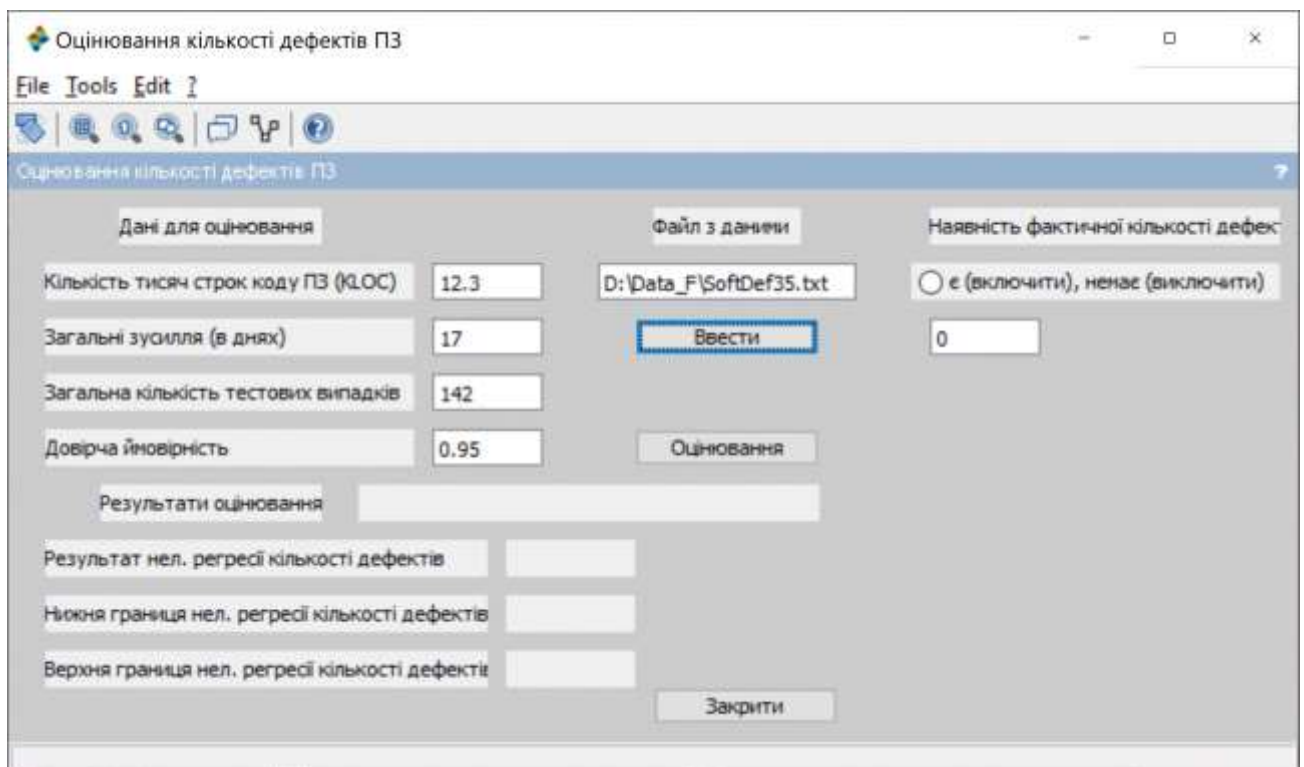


Рисунок 2.5. – Графічне вікно програми

Було проведено тестування як окремих модулів, так і програми в цілому. Результати тестування програми виконувалося для емпіричних даних з [10]: кількості дефектів з 35 програмних застосунків в залежності від кількості тисяч строк коду (KLOC); загальних зусиль (в днях) на етапах до тестування ПЗ та кількості тестових випадків. Були визначені значення нелінійної регресії, нижньої та верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ, значення коефіцієнту детермінації для нелінійної регресії. Результати візуалізації емпіричних даних з кількості дефектів з 35 програмних застосунків в залежності від кількості тисяч строк коду ПЗ (KLOC) показані на рис.2.6.

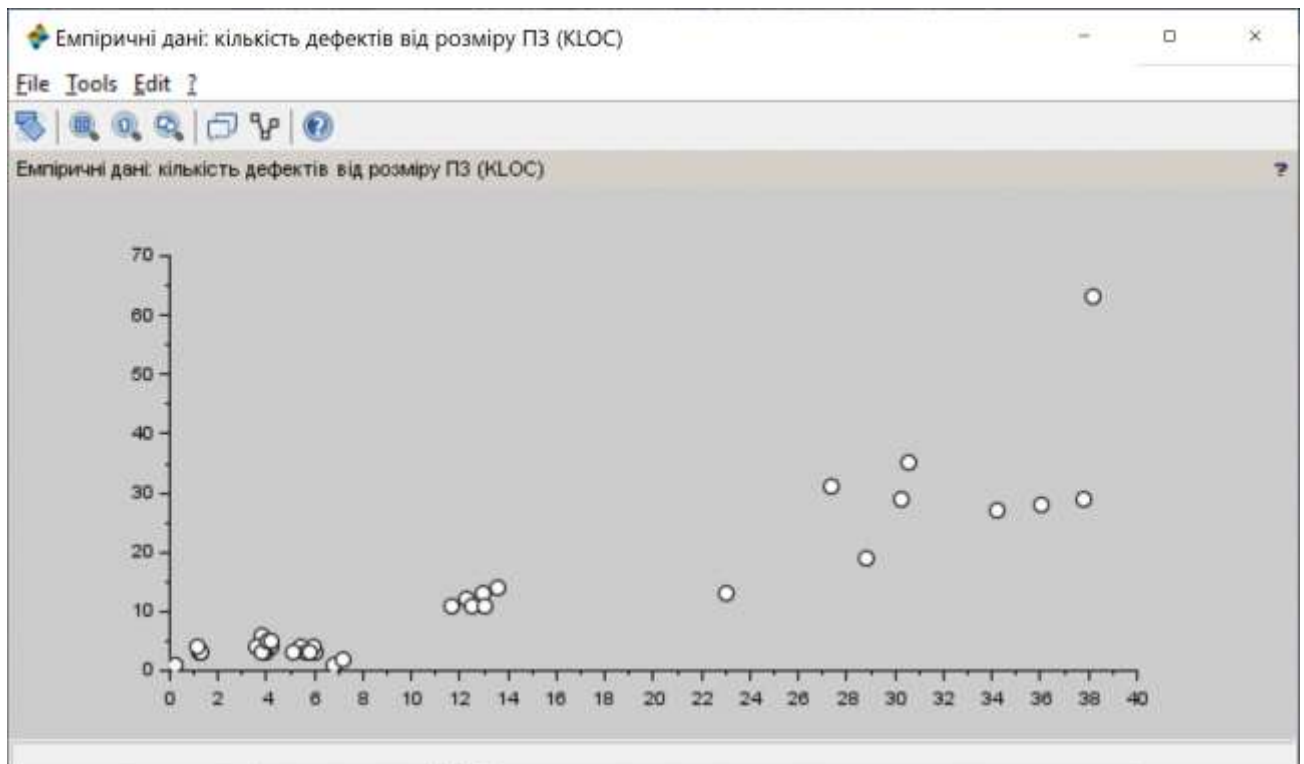


Рисунок 2.6. – Емпіричні дані з кількості дефектів з 35 програмних застосунків в залежності від розміру ПЗ (KLOC)

Результати візуалізації емпіричних даних з кількості дефектів з 35 програмних застосунків в залежності від загальних зусиль (в днях) на етапах до тестування ПЗ показані на рис.2.7.

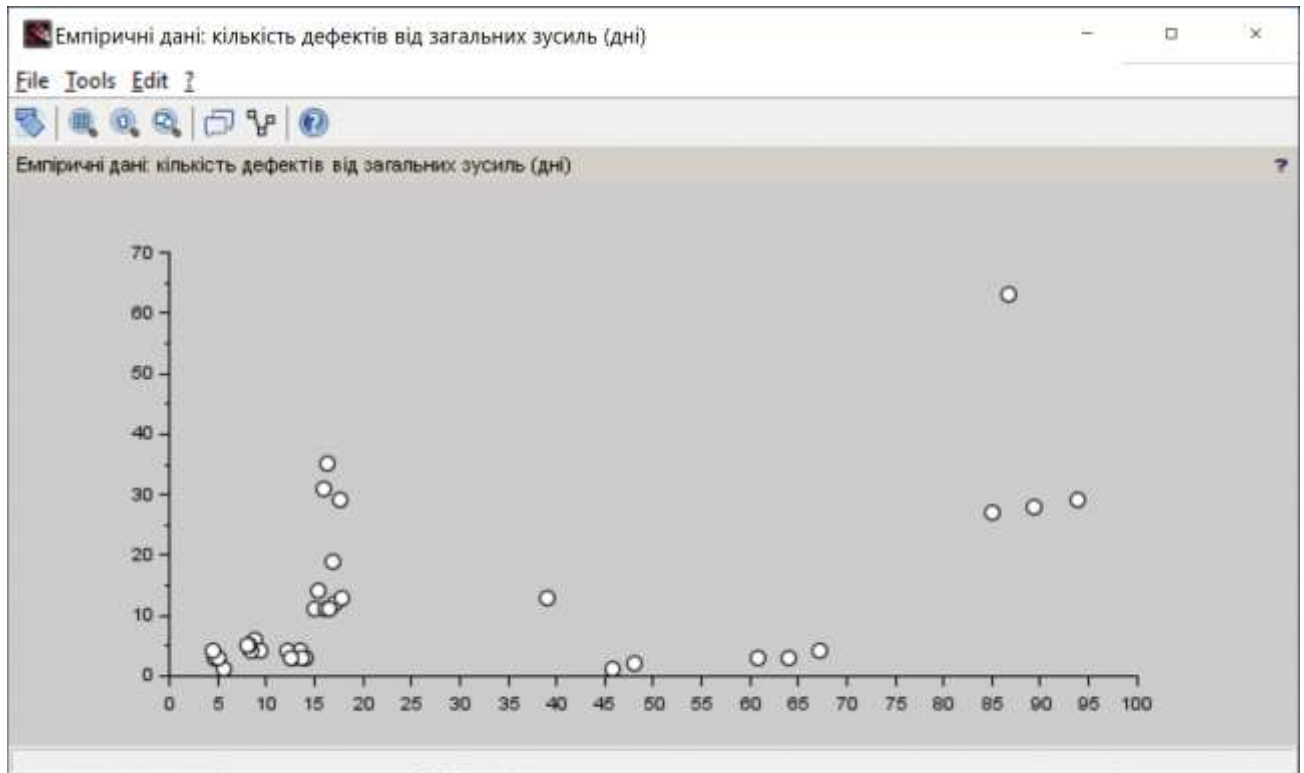


Рисунок 2.7. – Емпіричні дані з кількості дефектів з 35 програмних застосунків в залежності від загальних зусиль (в днях) на етапах до тестування ПЗ

Результати візуалізації емпіричних даних з кількості дефектів з 35 програмних застосунків в залежності від кількості тестових випадків наведені на рис.2.8.

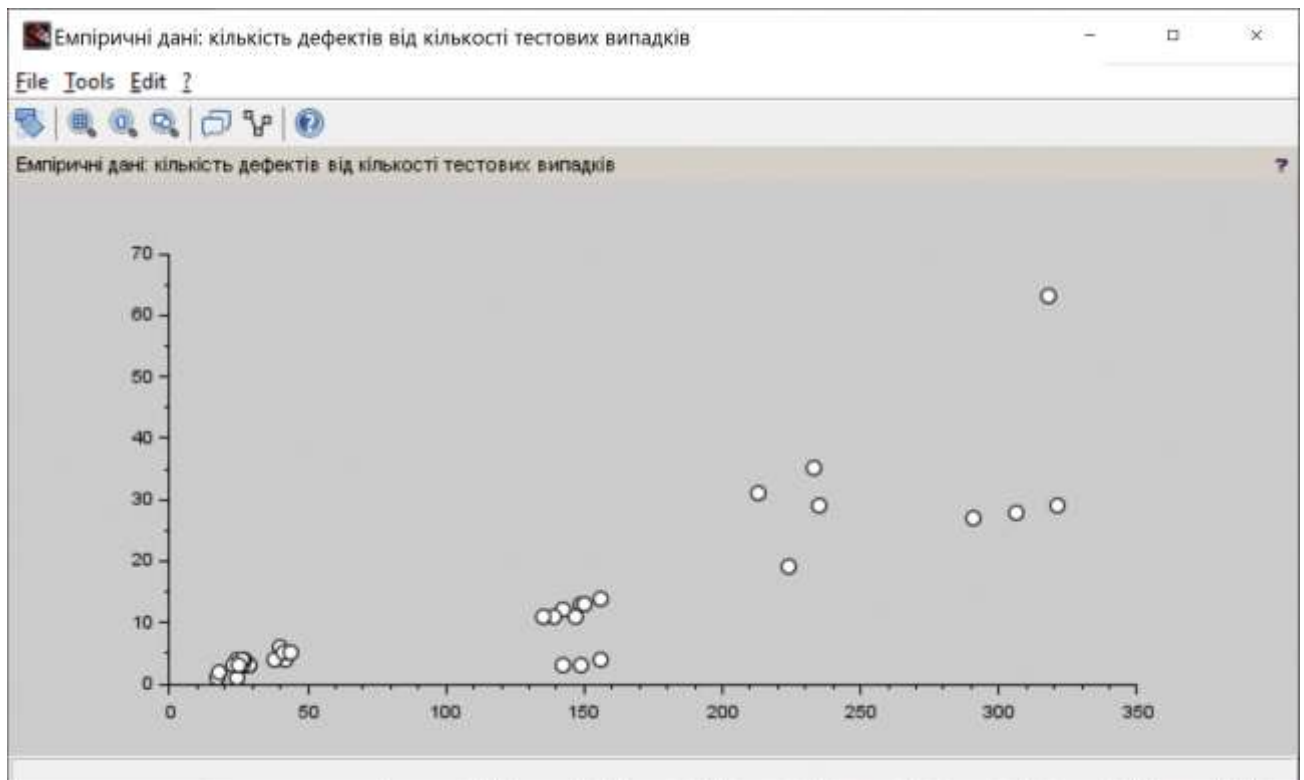


Рисунок 2.8. – Емпіричні дані з кількості дефектів з 35 програмних застосунків в залежності від кількості тестових випадків

Результати обчислення оцінок нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ та результати обчислення коефіцієнту детермінації у разі, коли не здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ наведені на рис.2.9.

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання

Кількість тисяч строк коду ПЗ (KLOC)

Загальні зусилля (в днях)

Загальна кількість тестових випадків

Довірча ймовірність

Файл з даними

Наявність фактичної кількості дефектів

☐ є (включити), ☒ немає (виключити)

Результати оцінювання Коефіцієнт детермінації $R^2=0.6639$

Результат нел. регресії кількості дефектів

Нижня границя нел. регресії кількості дефектів

Верхня границя нел. регресії кількості дефектів

Рисунок 2.9. – Результати обчислення оцінок нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ у разі, коли не здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ

Результати обчислення оцінок нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ, наведені на рис.2.10. У цьому разі виводиться значення відносної похибки MRE результату оцінювання значень нелінійної регресії в порівнянні з фактичною кількістю дефектів ПЗ для зазначених метрик.

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання		Файл з даними	Наявність фактичної кількості дефектів
Кількість тисяч строк коду ПЗ (KLOC)	22.99	D:\Data_F\SoftDef35.txt	☒ є (включити), ☐ немає (виключити)
Загальні зусилля (в днях)	39.1	Ввести	13
Загальна кількість тестових випадків	150		
Довірча ймовірність	0.95	Оцінювання	

Результати оцінювання

Коефіцієнт детермінації $R^2=0.6639$

Відносна похибка MRE=0.1047

Результат нел. регресії кількості дефектів	=14.3611
Нижня границя нел. регресії кількості дефектів	=12.2487
Верхня границя нел. регресії кількості дефектів	=16.8378

Закрити

Рисунок 2.10. – Результати обчислення оцінок нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ

Отримані результати практично є такими, що наведені в [10]. Це вказує на працездатність розробленої програми.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМИ ДЛЯ ОЦІНЮВАННЯ КІЛЬКОСТІ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В результаті виконання кваліфікаційної роботи було здійснено розробку програми SoftDefEst для оцінювання кількості дефектів ПЗ. Програму було створено ссі-мовою системи моделювання Scilab. В середовищі Scilab 6.0.0 було створено sse-файл розміром 11 КБайт.

Була розроблена наступна програмна документація: технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань, що наведені відповідно у додатках А, Б, В, Г, Д. Були проведені випробування програми згідно з програмою та методикою випробувань, яка наведена в додатку Д. При випробуванні програми ми зазначили використання тестів за методологією «чорної скриньки». Результати випробування програми показано на рис.3.1-3.5.

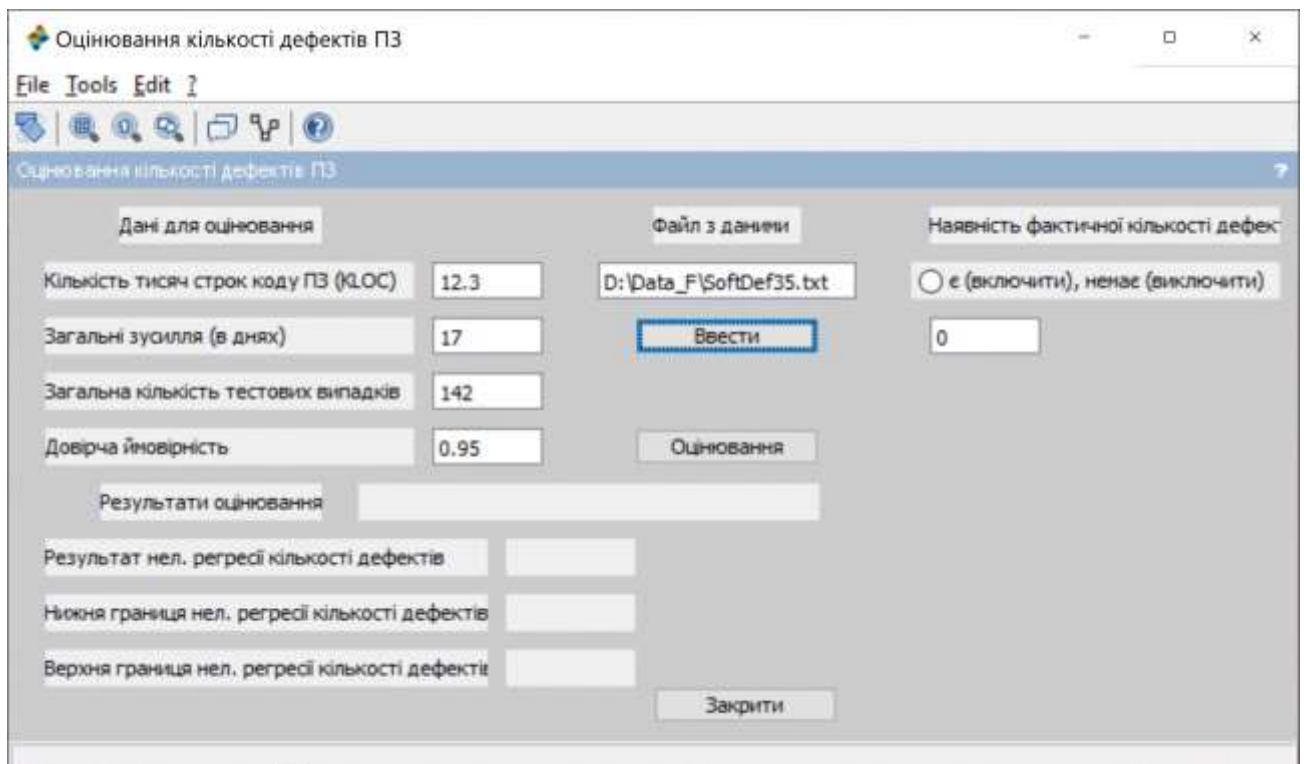


Рисунок 3.1 – Графічне вікно програми після її завантаження

Результати роботи програми після натискання на командну кнопку «Ввести» показано на рисунку 3.2.

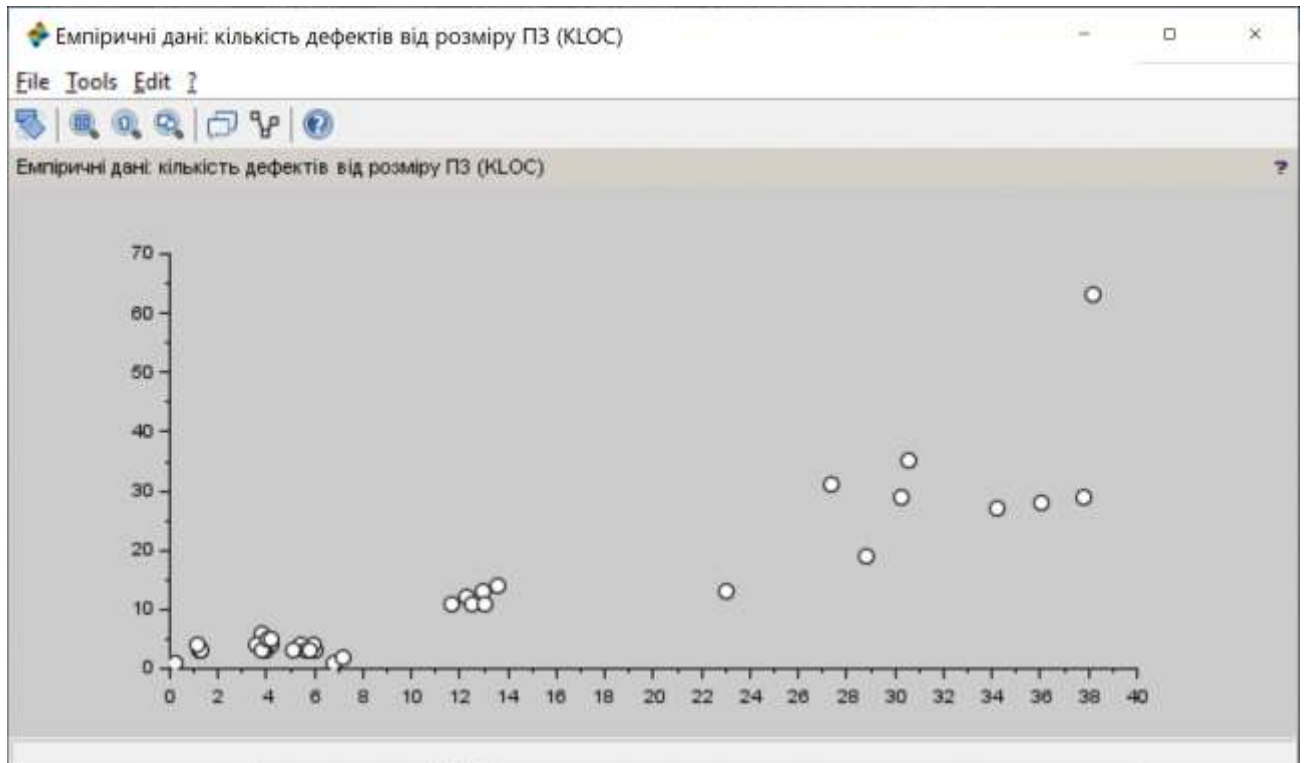


Рисунок 3.2 – Результати роботи програми після натискання на командну кнопку «Ввести»

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка виключена, показано на рисунку 3.3.

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання

Кількість тисяч строк коду ПЗ (KLOC) 12.3

Загальні зусилля (в днях) 17

Загальна кількість тестових випадків 142

Довірча ймовірність 0.95

Файл з даними D:\Data_F\SoftDef35.txt

Ввести

Наявність фактичної кількості дефектів

☐ є (включити), немає (виключити)

0

Оцінювання

Результати оцінювання Коефіцієнт детермінації $R^2=0.6639$

Результат нел. регресії кількості дефектів =14.2562

Нижня границя нел. регресії кількості дефектів =12.2939

Верхня границя нел. регресії кількості дефектів =16.5318

Закрити

Рисунок 3.3 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка виключена

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені фактична кількість дефектів ПЗ, що дорівнює 12, і значення довірчої ймовірності 0,95, наведені на рисунку 3.4.

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання

Кількість тисяч строк коду ПЗ (KLOC) 12.3

Загальні зусилля (в днях) 17

Загальна кількість тестових випадків 142

Довірча ймовірність 0.95

Файл з даними D:\Data_F\SoftDef35.txt

Ввести

Наявність фактичної кількості дефектів

☒ є (включити), немає (виключити)

12

Оцінювання

Результати оцінювання

Коефіцієнт детермінації $R^2=0.6639$

Відносна похибка MRE=0.1880

Результат нел. регресії кількості дефектів =14.2562

Нижня границя нел. регресії кількості дефектів =12.2939

Верхня границя нел. регресії кількості дефектів =16.5318

Закрити

Рисунок 3.4 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені фактична кількість дефектів ПЗ, що дорівнює 12, і значення довірчої ймовірності 0,95

Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені фактична кількість дефектів ПЗ, що дорівнює 12, і значення довірчої ймовірності 0,99, показано на рисунку 3.5.

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання		Файл з даними	Наявність фактичної кількості дефектів
Кількість тисяч строк коду ПЗ (KLOC)	12.3	D:\Data_F\SoftDef35.txt	☒ є (включити), немає (виключити)
Загальні зусилля (в днях)	17	Ввести	12
Загальна кількість тестових випадків	142		
Довірча ймовірність	0.99	Оцінювання	
Результати оцінювання		Коефіцієнт детермінації $R^2=0.6639$	
Результат нел. регресії кількості дефектів	=14.2562	Відносна похибка MRE=0.1880	
Нижня границя нел. регресії кількості дефектів	=11.6808		
Верхня границя нел. регресії кількості дефектів	=17.3995		
Закрити			

Рисунок 3.5 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введені фактична кількість дефектів ПЗ, що дорівнює 12, і значення довірчої ймовірності 0,99

Результати випробувань показали працездатність розробленої програми для оцінювання кількості дефектів ПЗ.

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Згідно статті 1 Закону України «Про охорону праці» [25]: «Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності».

Згідно статті 4 Закону України «Про охорону праці» [25] «Державна політика в галузі охорони праці базується на принципах:

пріоритету життя і здоров'я працівників, повної відповідальності роботодавця за створення належних, безпечних і здорових умов праці;

підвищення рівня промислової безпеки шляхом забезпечення суцільного технічного контролю за станом виробництв, технологій та продукції, а також сприяння підприємствам у створенні безпечних та нешкідливих умов праці;

комплексного розв'язання завдань охорони праці на основі загальнодержавної, галузевих, регіональних програм з цього питання та з урахуванням інших напрямів економічної і соціальної політики, досягнень в галузі науки і техніки та охорони довкілля;

соціального захисту працівників, повного відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань;

встановлення єдиних вимог з охорони праці для всіх підприємств та суб'єктів підприємницької діяльності незалежно від форм власності та видів діяльності;

адаптації трудових процесів до можливостей працівника з урахуванням його здоров'я та психологічного стану;

використання економічних методів управління охороною праці, участі держави у фінансуванні заходів щодо охорони праці, залучення добровільних внесків та інших надходжень на ці цілі, отримання яких не суперечить законодавству;

інформування населення, проведення навчання, професійної підготовки і підвищення кваліфікації працівників з питань охорони праці;

забезпечення координації діяльності органів державної влади, установ, організацій, об'єднань громадян, що розв'язують проблеми охорони здоров'я, гігієни та безпеки праці, а також співробітництва і проведення консультацій між роботодавцями та працівниками (їх представниками), між усіма соціальними групами під час прийняття рішень з охорони праці на місцевому та державному рівнях;

використання світового досвіду організації роботи щодо поліпшення умов і підвищення безпеки праці на основі міжнародного співробітництва.»

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці.

Згідно п.15 Наказу Міністерство охорони здоров'я України [26]: «Власник або уповноважений ним орган закладу, установи, організації, підприємства зобов'язаний:

- правильно організовувати працю працівників, щоб кожен працював за своєю спеціальністю і кваліфікацією, мав закріплене за ним робоче місце,

своєчасно до початку роботи, що доручається, був ознайомлений із встановленим завданням і забезпечений роботою протягом усього робочого дня (зміни); забезпечити здорові і безпечні умовами праці, справний стан інструментів, верстатів та іншого обладнання, а також нормативними запасами сировини, ліків, засобів медичного призначення, матеріалами та іншими ресурсами, необхідними для безперебійної і ритмічної роботи;

- створювати умови для забезпечення охорони здоров'я населення і високого рівня надання йому медичної допомоги, зростання продуктивності праці шляхом впровадження новітніх досягнень науки, техніки і наукової організації праці, організовувати вивчення і впровадження передових прийомів і методів праці, раціональних форм розподілу і кооперації праці, удосконалювати організацію і обслуговування робочих місць, механізовувати важкі і трудомісткі роботи; впроваджувати технічно обґрунтовані і переглядати застарілі норми праці (часу обслуговування);

- всіляко розвивати форми організації і стимулювання праці, здійснювати заходи щодо підвищення ефективності праці колективу, організовувати вивчення, поширення і впровадження передових прийомів та методів праці;

- своєчасно доводити до виробничих підрозділів планові завдання, забезпечувати їх виконання з найменшими затратами трудових, матеріальних і фінансових ресурсів, вживати заходи, спрямовані на більш повне виявлення і використання внутрішніх резервів, забезпечення науково обґрунтованого нормування витрат сировини і матеріалів, енергії та палива, раціонального й економного їх використання, підвищувати рентабельність виробництва, а також поліпшувати інші показники роботи;

- постійно удосконалювати організацію оплати праці залежно від кінцевих результатів роботи, підвищувати якість нормування праці; забезпечувати матеріальну зацікавленість працівників у результатах їх особистої праці та загальних підсумках роботи, економного і раціонального витрачання

фонду оплати праці та інших фондів; не допускати порушень у застосуванні діючих умов оплати і нормування праці;

- забезпечувати трудову і виробничу дисципліну, постійно здійснюючи організаторську, економічну і виховну роботу, спрямовану на її зміцнення, усунення втрат робочого часу, раціональне використання трудових ресурсів, формування стабільних трудових колективів; застосовувати дисциплінарні стягнення до порушників трудової дисципліни, враховуючи при цьому думку трудових колективів;

- неухильно дотримуватись законодавства про працю та охорону праці; поліпшувати умови праці, забезпечувати належне технічне обладнання всіх робочих місць і створювати на них умови роботи, що відповідають правилам з охорони праці (правилам з техніки безпеки, санітарним нормам і правилам та ін.). У разі відсутності в правилах вимог, дотримання яких при виконанні робіт необхідне для забезпечення безпечних умов праці, власник або уповноважений ним орган закладу, установи, організації, підприємства за погодженням із профспілковим органом вживає заходів щодо забезпечення безпечних умов праці;

- вживати необхідних заходів щодо профілактики виробничого травматизму, професійних та інших захворювань працівників, своєчасно надавати пільги і компенсації у зв'язку зі шкідливими умовами праці у випадках, передбачених законодавством (скорочений робочий день, додаткові відпустки, лікувально-профілактичне харчування та ін.), забезпечувати працівників відповідно до діючих норм і положень спеціальним одягом, спеціальним взуттям та іншими засобами індивідуального захисту, організовувати належний догляд за їх використанням;

- постійно контролювати рівень знань та дотримання працівниками вимог з техніки безпеки, виробничої санітарії, гігієни праці та протипожежної охорони;

- розвивати професійні змагання, створюючи умови трудовому колективу для всілякого підвищення продуктивності і ефективності суспільного

виробництва, поліпшення якості робіт та зниження їх собівартості, раціонального використання робочого часу, сировини, матеріалів, енергії, інших ресурсів, виконання умов договорів і замовлень, успішного виконання зобов'язань, своєчасного підведення підсумків, підвищення ролі морального та матеріального стимулювання високопродуктивної праці, вирішення питань про заохочення передових колективів і працівників; забезпечувати поширення передового досвіду та цінних ініціатив працівників цього та інших трудових колективів;

- своєчасно розглядати і впроваджувати винаходи та раціоналізаторські пропозиції, підтримувати і заохочувати новаторів виробництва, сприяти масовій технічній творчості;

- забезпечувати систематичне підвищення професійної (виробничої) кваліфікації працівників та рівня їх економічних і правових знань, створювати необхідні умови для поєднання роботи із навчанням на виробництві та в учбових закладах;

- створювати для трудового колективу необхідні умови для виконання ним своїх повноважень, передбачених законодавством та колективним договором, сприяти створенню у трудовому колективі ділової, творчої обстановки, всіляко підтримувати і розвивати ініціативу та активність працюючих, забезпечувати їх участь в управлінні закладами, установами, організаціями, підприємствами, своєчасно розглядати критичні зауваження та повідомляти колективу про вжиті заходи;

- уважно ставитись до потреб і запитів працівників, поліпшувати умови їх праці та побуту, здійснювати будівництво, ремонт і утримання в належному стані житлових будинків, гуртожитків, дитячих дошкільних установ, організовувати облік працівників, що потребують поліпшення житлових умов, розподіляти житлову площу відповідно до чинного законодавства і забезпечувати широку гласність при вирішенні цих питань.

Власник або уповноважений ним орган здійснює свої обов'язки у відповідних випадках спільно або за погодженням із профспілковим органом закладу, установи, організації, підприємства, а також з урахуванням повноважень трудового колективу.»

Згідно [27]: «Охорона праці – один з центральних інститутів трудового права. Він має виключне практичне значення. Недодержання вимог охорони праці створює небезпеку для здоров'я і життя працівників. У свою чергу й ті, кого законодавець називає власником або уповноваженим ним органом, несуть сувору, в тому числі і кримінальну, відповідальність за порушення правил охорони праці.»

4.2 Аналіз шкідливих і небезпечних факторів в приміщенні з комп'ютерами

Працівники, що працюють в приміщенні з комп'ютерами або персональними комп'ютерами (ПК) піддаються шкідливій дії комп'ютерів, тому робочі місця користувачів повинні відповідати нешкідливим і безпечним умовам праці.

У зв'язку з цим треба розробити комплекс заходів, що забезпечують нешкідливі і безпечні умови праці.

При виконанні посадових обов'язків на працівників в приміщенні, де вони працюють з ПК (постійно або періодично), діють наступні шкідливі та небезпечні фактори:

- 1) забруднення повітря шкідливими речовинами, пилом, мікроорганізмами;
- 2) невідповідність нормам параметрів мікроклімату;

- 3) виникнення на екрані монітора електростатичних зарядів, що змушують часточки пилу рухатися до найближчого заземленого предмету, часто нею виявляється особа користувача;
- 4) підвищений рівень шуму на робочому місці;
- 5) підвищений рівень статичної електрики при невірно спроектованій робочій зоні;
- 6) небезпечний рівень напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;
- 7) широкий спектр випромінювання від дисплея, що включає рентгенівську ультрафіолетову й інфрачервону області, а так само широкий діапазон електромагнітних випромінювань інших частот;
- 8) підвищений рівень електромагнітних випромінювань;
- 9) підвищений рівень іонізуючих випромінювань (м'яке рентгенівське та гамма-випромінювання);
- 10) відсутність або недолік природного світла;
- 11) недостатня освітленість робочої зони;
- 12) підвищена яскравість світла;
- 13) знижена контрастність;
- 14) пряма і зворотна блескність;
- 15) підвищена пульсація світлового потоку (мерехтіння зображення);
- 16) тривале перебування в тому самому положенні, і повторення тих самих рухів приводить до синдрому тривалих статичних навантажень (СТСН);
- 17) нераціональна організація робочого місця;
- 18) невідповідність ергономічних характеристик устаткування нормованим величинам;
- 19) великий обсяг інформації, що переробляється, приводить до значних навантажень на органи зору та нервову систему;
- 20) монотонність праці;
- 21) нервово-психічні навантаження;

- 22) нервово-емоційні стресові навантаження;
- 23) небезпека виникнення пожежі.

4.3 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів

Згідно з ДБН В. 2.5-28-2006 "Організація робочих місць" мінімальна освітленість робочого місця в офісі повинна бути не менше 300 лк. Нормована освітленість робочого столу з ПК повинна бути в межах 300-500 лк. Допускається установка світильників місцевого освітлення для підсвічування документів. При цьому місцеве освітлення так само, як і загальне, не повинно створювати відблисків на екрані відеомоніторів. Бажано, щоб природне світло падало ліворуч, хоча допускається і таке розміщення робочого місця, коли віконні прорізи розташовані праворуч. Зовнішня освітленість екрана монітора не повинна перевищувати 300 лк. Вікна повинні бути обладнані пристроями типу жалюзі, зовнішніх козирків і інше.

Зниження шуму, який створюється ПК, є дуже важливою задачею. Зниження шуму в джерелі випромінювання можна забезпечити застосуванням пружних прокладок між ПК і опорною поверхнею. Як прокладки використовуються гума, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, прокладки з м'якої гуми, повсті товщиною 6-8 мм. Кріплення прокладок можливо шляхом приклейки їх до опорних частин.

Можливо також застосування звукоізолюючих кожухів, що не заважають технологічному процесові. Не менш важливим для зниження шуму в процесі

експлуатації є питання правильного і своєчасного регулювання, змазування і заміни механічних вузлів шумливого устаткування.

Раціональне планування приміщення, розміщення устаткування є важливим чинником, що дозволяє знизити шум при існуючому устаткуванні ПК. Приміщення для сервісної апаратури необхідно розташовувати удалині від шумливого і вібруючого устаткування.

Зниження рівня шуму, що проникає у виробниче приміщення ззовні, може бути досягнуто збільшенням звукоізоляції конструкцій, що обгороджують, ущільненням по периметру притворів вікон, дверей.

У такий спосіб для зниження шуму створюваного на робочих місцях внутрішніми джерелами, а також шуму, що проникає ззовні, потрібно:

- послабити шум самих джерел (застосування екранів, звукоізолюючих кожухів);
- знизити ефект сумарного впливу відбитих звукових хвиль (звуковбирні поверхні конструкцій);
- застосовувати раціональне розташування устаткування;
- використовувати архітектурно-планувальні і технологічні рішення ізоляцій джерел шуму.

До робочого столу теж є певні вимоги. Його конструкція повинна забезпечувати оптимальне розміщення устаткування, яке використовується працівником. Робочий стілець працівника необхідно розташувати з таким розрахунком, щоб він дозволяв змінювати позу працюючого, а також був підйомно-поворотним. Це має допомогти уникнути стомлення м'язів шийно-плечової області і спини. Оптимальна відстань від очей до екрану монітора – 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків і символів. Недотримання зазначеної відстані може призвести до швидкого розвитку стомлення та перевтоми зорового аналізатора і зі збільшенням тривалості роботи сприяти розвитку короткозорості. Відстань між робочими столами з відеомоніторами повинне бути не менш 2,0 м, а

відстань між бічними поверхнями відеомоніторів – не менш 1,2 м. Робочі місця з ПК у приміщеннях із джерелами шкідливих виробничих факторів розміщаються в ізольованих кабінах з організованим повітрообміном.

ВИСНОВКИ

В кваліфікаційній роботі було вирішено практичну задачу інженерії ПЗ, що характеризується комплексністю та невизначеністю умов, а саме розробці програми для оцінювання кількості дефектів ПЗ.

В кваліфікаційній роботі було вирішені такі завдання:

- Здійснено аналіз практичної задачі інженерії програмного забезпечення з розробки програми для оцінювання кількості дефектів ПЗ;
- Здійснено аналіз існуючих інструментів для оцінювання кількості дефектів ПЗ;
- Проаналізовано існуючі регресійні моделі для оцінювання кількості дефектів ПЗ;
- Розроблено проєкт програми для оцінювання кількості дефектів ПЗ із заданою довірчою ймовірністю;
- Проведено випробування розробленої програми.

Також була розроблена наступна програмна документація: технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань, що наведені відповідно у додатках А, Б, В, Г, Д.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AkiyAverma, F. An example of software system debugging / Fumio AkiyAverma // In *Proceedings of International Federation of Information Processing Societies Congress (IFIPS'71)*. – 1971. – P. 353-359.
2. Tanwar, H. A Review of Software Defect Prediction Models / H. Tanwar M. Kakkar // in Balas V., Sharma N., Chakrabarti A. (eds) *Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing*, vol. 808, Singapore: Springer, 2019. DOI: 10.1007/978-981-13-1402-5_7
3. Can, H. A new model for software defect prediction using Particle Swarm Optimization and support vector machine / H. Can, X. Jianchun, Z. Ruide, L. Juelong, Y. Qiliang, X. Liqiang // in *25th Chinese Control and Decision Conference*, 25-27 May 2013, GuiyAverng, China, 2013. – P. 4106–4110. DOI: 10.1109/CCDC.2013.6561670
4. Mahajan, R. Design of Software Defect Forecast model using BR technique / R. Mahajan, S. K. Gupta, R. K. Bedi // *Procedia Computer Science*, vol.46, 2015. – P. 849-858.
5. Sethi, T. Improved approach for software defect prediction using artificial neural networks / T. Sethi, Gagandeep // in *5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, 7-9 September 2016, Noida, India, 2016. – P. 480–485. DOI: 10.1109/ICRITO.2016.7785003
6. Kumar, L. Effective fault prediction model developed using Least Square Support Vector Machine (LSSVM) / L. Kumar, S. K. Sripada, A. Sureka, S. K. Rath // *Journal of Systems and Software*, 2017. – P. 1-28.
7. SubramanyAverm, R. Empirical analysis of CK metrics for object-oriented design complexity: implications for software defects / R. SubramanyAverm, M. S. Krishnan // in *IEEE Transactions on Software Engineering*, vol. 29, no. 4. – P. 297-310, April 2003. DOI: 10.1109/TSE.2003.1191795.

8. Suffian, M. D. M. A prediction model for system testing defects using regression analysis / M. D. M. Suffian, S. Ibrahim // *International Journal of Soft Computing and Software Engineering*, vol. 2, No. 7, 2012. – P. 69–78. DOI: 10.7321/jscse.v2.n7.6

9. Valles-Barajas, F. A comparative analysis between two techniques for the prediction of software defects: fuzzy and statistical linear regression / F. Valles-Barajas // *Innovations in Systems and Software Engineering*, vol. 11, issue 4, 2015. – P. 277–287. DOI: 10.1007/s11334-015-0256-4

10. Prykhodko, S. A Non-Linear Regression Model to Estimate the Number of Defects for Software Testing Phase / Sergiy Prykhodko, Natalia Prykhodko, Lidiia Makarova, Kateryna Prykhodko, Andrii Pukhalevych, TatyAverna Smykodub // *Proceedings of the 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, July 2 - 6, 2019, Lviv, Ukraine. – P. 965-969. DOI: 10.1109/UKRCON.2019.8880027

11. Pradhan, S. On the Defect Prediction for Large Scale Software Systems – From Defect Density to Machine Learning / S. Pradhan, V. Nanniyur, P. K. Vissapragada // *In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, Macau, China, 2020. – P. 374-381. DOI: 10.1109/QRS51102.2020.00056

12. Okumoto, K. Early Software Defect Prediction: Right-Shifting Software Effort Data into a Defect Curve / K. Okumoto // *in 2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Charlotte, NC, USA, 2022, – P. 43-48. DOI: 10.1109/ISSREW55968.2022.00037

13. Nandwana, P. Software Defect Prediction using Cutting Edge Clustering Module and Machine Learning / P. Nandwana, S. Lenka, P. Khatri, Z. T. Khan, D. Sakhare, A. Salunkhe // *in 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2022. – P. 717-720. DOI: 10.1109/ICIRCA54612.2022.9985569

14. Marapelli, B. Software Defect Prediction Using ROS-KPCA Stacked Generalization Model / B. Marapelli, A. Carie, S.M.N. Islam// In: Bhateja, V., Y.Averng, X.S., Lin, J.CW., Das, R. (eds) *Evolution in Computational Intelligence. FICTA 2022. Smart Innovation, Systems and Technologies*, vol 326. Springer, Singapore, 2023. DOI: 10.1007/978-981-19-7513-4_51

15. Lee, G. An Empirical Comparison of Model-Agnostic Techniques for Defect Prediction Models / G. Lee, S. U.-J. Lee // in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Taipa, Macao, 2023. – P. 179-189. DOI: 10.1109/SANER56733.2023.00026

16. Mehmood, I. A Novel Approach to Improve Software Defect Prediction Accuracy Using Machine Learning / I. Mehmood et al. // *IEEE Access*, vol. 11, 2023. – P. 63579-63597. DOI: 10.1109/ACCESS.2023.3287326

17. Wang, Y. Research on Software Defects Prediction and Performance Analysis based on Machine Learning / Y. Wang, S. Huang, Y. Xu, H. Zhang // in *2024 7th International Conference on Computer Information Science and Application Technology (CISAT)*, Hangzhou, China, 2024. – P. 774-777. DOI: 10.1109/CISAT62382.2024.10695196

18. Zhang, Y. Software Defect Prediction Model Based on Deep Learning / Y. Zhang // in *2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEEC)*, Athens, Greece, 2024. – P. 954-959. DOI: 10.1109/PEEEEC63877.2024.00177

19. Jadhav, S. Manikandan and D. Ryu, Enhancing the Software Defect Prediction using AutoInt / S. Jadhav, S. Manikandan, D. Ryu // in *2025 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kota Kinabalu, Malaysia, 2025. – P. 103-104. DOI: 10.1109/BigComp64353.2025.00029

20. Mao, X. Towards Defect Prediction for Quantum Software / X. Mao, Z. Huang, J. Ge, C. Wang, W. Wang and L. Cai // in *2025 IEEE/ACM International Workshop on Quantum Software Engineering (Q-SE)*, Ottawa, ON, Canada, 2025. – P. 1-8. DOI: 10.1109/Q-SE66736.2025.00007

21. Estimate your software projects with confidence. Режим доступу: <https://www.qsm.com/>

22. SLIM-Estimate for Software Project Estimation. Режим доступу: <https://www.qsm.com/tools/slim-estimate>

23. SLIM-Metrics for Software Benchmarking. Режим доступу: <https://www.qsm.com/tools/slim-metrics>

24. The all-in-one platform for Agile teams. Режим доступу: https://clickup.com/lp/teams/agile?utm_source=google&utm_medium=cpc&utm_campaign=gs_cpc_arlv_nnc_nb_trial_all-devices_troas_lp_x_all-departments_x_methodology&utm_content=all-countries_kw-target_text_x_all-features_all-use-cases_methodology-scrum&utm_term=e_scrum%20board&utm_creative=713191054458_dynamic_rsa&utm_custom1=&utm_custom2=&utm_lptheme=&utm_lpmod=&gad_source=1&gad_campaignid=20988031127&gbraid=0AAAAACR5vIJDD0OhVYCzJv0oBzVdW7sH_&gclid=Cj0KCQjw9O_BBhCUARIsAHQMjS5BkPBeKEapXyJf0oPCJhkc5IWFIVJEC1D4y2swYz9b3estTLyF6mS8DUaAnhiEALw_wcB

25. Закону України "Про охорону праці". Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

26. Наказ Міністерство охорони здоров'я України "Про затвердження Галузевих правил внутрішнього трудового розпорядку" (18.12.2000 N 204-о) Режим доступу: <https://zakon.rada.gov.ua/rada/show/vb2est04282-00#Text>

27. НАУКОВО-ПРАКТИЧНИЙ КОМЕНТАР до статті 153 Кодексу законів про працю України. Режим доступу: <https://ips.ligazakon.net/document/KK000167>

Додаток А – Технічне завдання на розробку програми для оцінювання кількості дефектів ПЗ

Вступ

Назва розробки – «Програма для оцінювання кількості дефектів ПЗ», далі програма.

Підстави для розробки

Завдання на кваліфікаційну роботу.

Призначення розробки:

Програма призначена для оцінювання кількості дефектів ПЗ за нелінійною регресійною моделлю.

Вимоги до програми:

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

1) Вводити з текстового файлу наступні емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

2) Вводити з графічного вікна (інтерфейсу програми) значення наступних величин, за якими потрібно оцінити кількість дефектів ПЗ: довірчу ймовірність, кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків.

3) Вводити з інтерфейсу програми емпіричне (фактичне) значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

4) Перевіряти правильність вводу початкових (вхідних) даних.

5) Виконувати нормалізацію емпіричних даних за десятковим логарифмом.

6) Знаходити точкову оцінку умовного математичного сподівання (умовне вибіркове середнє) кількості дефектів ПЗ за нелінійною регресією.

7) Обчислювати та виводити значення коефіцієнту детермінації R^2 нелінійної регресії.

8) Обчислювати та виводити значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

9) Знаходити інтервальну оцінку умовного вибіркового середнього (довірчий інтервал нелінійної регресії) кількості дефектів ПЗ.

10) Виводити на екран умовне вибіркове середнє кількості дефектів ПЗ за нелінійною регресією, довірчий інтервал нелінійної регресії кількості дефектів ПЗ, значення коефіцієнту детермінації R^2 , значення відносної похибки MRE (у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ).

11) Виводити у текстовий файл результати оцінювання кількості дефектів ПЗ.

Вхідні дані:

- емпіричні дані: фактична кількість дефектів; кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- значення довірчої ймовірності та трьох метрик, за якими потрібно оцінити кількість дефектів ПЗ: кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ та кількість тестових випадків;

- емпіричне значення кількості дефектів ПЗ у разі необхідності (коли це значення відоме і треба його порівняти з оцінкою).

Вихідні дані:

- значення умовного вибіркового середнього за нелінійною регресією та значення границь довірчого інтервалу нелінійної регресії кількості дефектів ПЗ.

- значення коефіцієнту детермінації;

- значення відносної похибки MRE у разі, коли здійснюється введення з інтерфейсу програми емпіричного значення кількості дефектів ПЗ.

Вимоги до умов експлуатації не висуваються.

Вимоги до надійності не висуваються.

Вимоги до складу й параметрів технічних засобів.

Вимоги до складу й параметрів технічних засобів є такими: процесор з інструкціями SSE2, оперативна пам'ять (RAM) не менше 2 Гб, простір на жорсткому диску 600 Мб мінімум.

Вимоги до інформаційної та програмної сумісності.

Вимоги до інформаційної та програмної сумісності є такими: операційна система Windows 10 (або вища), система моделювання Scilab 6.0.0 (або вища), компілятор C (Visual Studio 2010 або Visual Express 2010 або вищі).

Вимоги до маркування та пакування.

Не висуваються.

Вимоги до транспортування та зберігання.

Не висуваються.

Вимоги до програмної документації.

До програмної документації мають входити наступні документи: технічне завдання на розробку програми, текст програми, опис програми, інструкція користувача, програма та методика випробувань.

Техніко-економічні показники не розраховуються.

Стадії та етапи розробки.

Стадії та етапи розробки програми представлені у таблиці.

Таблиця – Стадії та етапи розробки

Стадія розробки	Етапи робіт	Термін виконання
1. Технічне завдання	Розробка та затвердження технічного завдання	21.03.2025
2. Ескізний проєкт	Розробка ескізного проєкту	24.03.2025
	Затвердження ескізного проєкту	26.03.2025
3. Технічний проєкт	Розробка технічного проєкту	08.04.2025
	Затвердження технічного проєкту	10.04.2025
4. Робочий проєкт	Розробка програми	05.05.2025
	Розробка програмної документації	12.05.2025
	Випробування програми	21.05.2025

Порядок контролю і приймання

Порядок контролю та приймання програми здійснюється представниками замовника у присутності представників розробника.

Для приймання програми надається програмна документація, яка зазначена у технічному завданні (Вимоги до програмної документації). Представники замовника установлюють відповідність програми технічному завданню.

За результатами приймання програми представниками замовника складається акт приймання програми.

Додаток Б – Текст програми

```
//Оцінювання кількості дефектів ПЗ
f=figure();
set(f,'position',[0,0,700,300])
set(f,'figure_name',...
'Oцінювання кількості дефектів ПЗ');

lab_text0=uicontrol(f,'style','text','string',...
'Файл з даними','position',...
[350,270,80,20]);
edit_text=uicontrol(f,'style','edit','string',...
'D:\Data_F\SoftDef35.txt','position',[320,240,140,20]);

BtDatInp=uicontrol('style','pushbutton','string','Ввести',...
'CallBack','[x,yx,f1,f2,f3,n]=DataInput()','position',[340,210,100,20]);

lab_text2=uicontrol(f,'style','text','string',...
'Наявність фактичної кількості дефектів ПЗ','position',...
[500,270,190,20]);
lab_text3=uicontrol(f,'style','text','string',...
'Дані для оцінювання','position',...
[60,270,110,20]);
lab_x1i=uicontrol(f,'style','text','string',...
'Кількість тисяч строк коду ПЗ (KLOC)','position',...
[20,240,200,20]);
edit_x1i=uicontrol(f,'style','edit','string',...
'12.3','position',[230,240,60,20]);
lab_x2i=uicontrol(f,'style','text','string',...
'Загальні зусилля (в днях)','position',...
[20,210,200,20]);
edit_x2i=uicontrol(f,'style','edit','string',...
'17','position',[230,210,60,20]);
lab_x3i=uicontrol(f,'style','text','string',...
'Загальна кількість тестових випадків','position',...
[20,180,200,20]);
edit_x3i=uicontrol(f,'style','edit','string',...
'142','position',[230,180,60,20]);
lab_Pconfid=uicontrol(f,'style','text','string',...
'Довірча ймовірність','position',...
[20,150,200,20]);
edit_Pconfid=uicontrol(f,'style','edit','string',...
'0.95','position',[230,150,60,20]);
lab_text8=uicontrol(f,'style','text','string',...
'Результати оцінювання','position',...
[50,120,120,20]);
lab_text9=uicontrol(f,'style','text','string',...
'Результат нел. регресії кількості дефектів','position',...
[50,120,120,20]);
```

```

[20,90,240,20]);
lab_text10=uicontrol(f,'style','text','string',...
'Нижня границя нел. регресії кількості дефектів','position',...
[20,60,240,20]);
lab_text11=uicontrol(f,'style','text','string',...
'Верхня границя нел. регресії кількості дефектів','position',...
[20,30,240,20]);
//
text_y_result=uicontrol(f,'style','text','string','', 'position',...
.
[270,90,70,20]);
text_yLBCONF=uicontrol(f,'style','text','string','', 'position',...
[270,60,70,20]);
text_yUBCONF=uicontrol(f,'style','text','string','', 'position',...
[270,30,70,20]);
text_r_2=uicontrol(f,'style','text','string','', 'position',...
[190,120,250,20]);

//Прапорець, який відповідає за вибір
radio_bikv_x=uicontrol('style','radiobutton','string',...
'є (включити), немає (виключити)', 'value',1, 'position',...
[490,240,200,20]);

if (get(radio_bikv_x, 'value')==0)

else

    edit_text12=uicontrol(f,'style','edit','string',...
    '0', 'position', [500,210,60,20]);
end;

Btnr_pred_int=uicontrol('style','pushbutton','string','Оцінювання'
, 'CallBack', ...
'SoftDefEst(x,yx,n)', 'position', [340,150,100,20]);

BtClose=uicontrol('style','pushbutton','string','Закрити',...
'CallBack', 'Wind_Close()', 'position', [350,10,100,20]);

function [x, yx, f1, f2, f3, n]=DataInput()
dir_name=get(edit_text, 'string');

file_D_1=mopen(dir_name, 'r');
file_D_2=mopen('D:\Data_F\SoftDef35_data2.txt', 'w+');
file_D_3=mopen('D:\Data_F\SoftDef35_data3.txt', 'w+');
i=1;
//
while (feof(file_D_1))<1
    y(i)=mfscanf(file_D_1, '%f'); x1(i)=mfscanf(file_D_1, '%f');
    x2(i)=mfscanf(file_D_1, '%f'); x3(i)=mfscanf(file_D_1, '%f');
    i=i+1;
end;

```

```

n=i-1;

fclose(file_D_1);

for i = 1:n
    x(i,1)=x1(i); x(i,2)=x2(i); x(i,3)=x3(i);
    yx(i,1)=y(i); yx(i,2)=x1(i); yx(i,3)=x2(i); yx(i,2)=x3(i);
end;

    fprintf(file_D_2, '%f\t%f\t%f\t%f\n', y, x1, x2, x3);

for i = 1:n
    for j = 1:3 fprintf(file_D_3, "%5.3f ", x(i,j)); end;
    fprintf(file_D_3, "\n");
end;

fclose(file_D_2);
fclose(file_D_3);
f1=figure();
set(f1, 'position', [50,50,700,300]);
set(f1, 'figure_name', 'Емпіричні дані: кількість дефектів від розміру ПЗ (KL0C)');
plot2d(x1,y,-9);

f2=figure();
set(f2, 'position', [60,60,710,310]);
set(f2, 'figure_name', 'Емпіричні дані: кількість дефектів від загальних зусиль (дні)');
plot2d(x2,y,-9);

f3=figure();
set(f3, 'position', [70,70,720,320]);
set(f3, 'figure_name', 'Емпіричні дані: кількість дефектів від кількості тестових випадків');
plot2d(x3,y,-9);

x1Aver=0; x2Aver=0; x3Aver=0;
for i=1:n
    x1Aver=x1Aver+x(i,1); x2Aver=x2Aver+x(i,2);
    x3Aver=x3Aver+x(i,3);
end;
x1Aver=x1Aver/n; x2Aver=x2Aver/n; x3Aver=x3Aver/n;
Dx1x1=0; Dx2x2=0; Dx3x3=0;
Dx1x2=0; Dx1x3=0; Dx2x3=0;
for i=1:n
    Dx1x1=Dx1x1+(x(i,1)-x1Aver)^2;
    Dx1x2=Dx1x2+(x(i,1)-x1Aver)*(x(i,2)-x2Aver);
    Dx1x3=Dx1x3+(x(i,1)-x1Aver)*(x(i,3)-x3Aver);
    Dx2x2=Dx2x2+(x(i,2)-x2Aver)^2;
    Dx2x3=Dx2x3+(x(i,2)-x2Aver)*(x(i,3)-x3Aver);
    Dx3x3=Dx3x3+(x(i,3)-x3Aver)^2;

```

```

end;
Sxx(1,1)=Dx1x1; Sxx(1,2)=Dx1x2; Sxx(1,3)=Dx1x3;
Sxx(2,1)=Dx1x2; Sxx(2,2)=Dx2x2; Sxx(2,3)=Dx2x3;
Sxx(3,1)=Dx1x3; Sxx(3,2)=Dx2x3; Sxx(3,3)=Dx3x3;

Sxx_1=1/Sxx;

z1Aver=0; z2Aver=0; z3Aver=0;
for i=1:n
    z1Aver=z1Aver+log10(x(i,1)); z2Aver=z2Aver+log10(x(i,2));
    z3Aver=z3Aver+log10(x(i,3));
end;
z1Aver=z1Aver/n; z2Aver=z2Aver/n; z3Aver=z3Aver/n;
Dz1z1=0; Dz2z2=0; Dz3z3=0;
Dz1z2=0; Dz1z3=0; Dz2z3=0;
for i=1:n
    Dz1z1=Dz1z1+(log10(x(i,1))-z1Aver)^2;
    Dz1z2=Dz1z2+(log10(x(i,1))-z1Aver)*(log10(x(i,2))-z2Aver);
    Dz1z3=Dz1z3+(log10(x(i,1))-z1Aver)*(log10(x(i,3))-z3Aver);
    Dz2z2=Dz2z2+(log10(x(i,2))-z2Aver)^2;
    Dz2z3=Dz2z3+(log10(x(i,2))-z2Aver)*(log10(x(i,3))-z3Aver);
    Dz3z3=Dz3z3+(log10(x(i,3))-z3Aver)^2;
end;
Szz(1,1)=Dz1z1; Szz(1,2)=Dz1z2; Szz(1,3)=Dz1z3;
Szz(2,1)=Dz1z2; Szz(2,2)=Dz2z2; Szz(2,3)=Dz2z3;
Szz(3,1)=Dz1z3; Szz(3,2)=Dz2z3; Szz(3,3)=Dz3z3;

Szz_1=1/Szz;

zi(1)=log10(x(1,1))-z1Aver; zi(2)=log10(x(1,2))-z2Aver;
zi(3)=log10(x(1,3))-z3Aver;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Df=n-4;
P=0.975;
Q=1-P;

[T]=cdft("T",Df,P,Q);

b0est=-0.3354;
b1est=0.5185;
b2est=-0.4248;
b3est=0.6723;

Dzyzyn4=0;
for i=1:n
    Dzyzyn4=Dzyzyn4+..

```

```

(b0est+b1est*log10(x(i,1))+b2est*log10(x(i,2))+b3est*log10(x(i,3))
-log10(yx(i,1)))^2;
end;
Dzyzyn4=Dzyzyn4/(n-4)
Delta_z=T*(Dzyzyn4*(1/n+result))^0.5;

zy_result=b0est+b1est*log10(x(1,1))+b2est*log10(x(1,2))+b3est*log1
0(x(1,3));

zyLBCONF=zy_result-Delta_z; zyUBCONF=zy_result+Delta_z;

y_result=10^zy_result;
yLBCONF=10^zyLBCONF; yUBCONF=10^zyUBCONF;

file_D_5=fopen('D:\Data_F\SoftDef35_data5.txt','w+');

    fprintf(file_D_5, "%5.6f ", yUBCONF);

fclose(file_D_5);

file_D_4=fopen('D:\Data_F\SoftDef35_data4.txt','w+');
for i = 1:3
    for j = 1:3 fprintf(file_D_4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_D_4, "\n");
end;
fclose(file_D_4);

endfunction

function SoftDefEst(x, yx, n)
//оцінювання кількості дефектів ПЗ

x1i=eval(get(edit_x1i,'string'));
x2i=eval(get(edit_x2i,'string'));
x3i=eval(get(edit_x3i,'string'));
Pconfid=eval(get(edit_Pconfid,'string'));

x1Aver=0; x2Aver=0; x3Aver=0; yAver=0;
for i=1:n
    x1Aver=x1Aver+x(i,1); x2Aver=x2Aver+x(i,2);
x3Aver=x3Aver+x(i,3); yAver=yAver+yx(i,1);
end;
x1Aver=x1Aver/n; x2Aver=x2Aver/n; x3Aver=x3Aver/n; yAver=yAver/n;
Dx1x1=0; Dx1x2=0; Dx2x2=0;
Dx1x3=0; Dx2x3=0; Dx3x3=0; SyAver=0;
for i=1:n
    Dx1x1=Dx1x1+(x(i,1)-x1Aver)^2;
    Dx1x2=Dx1x2+(x(i,1)-x1Aver)*(x(i,2)-x2Aver);
    Dx1x3=Dx1x3+(x(i,1)-x1Aver)*(x(i,3)-x3Aver);
    Dx2x2=Dx2x2+(x(i,2)-x2Aver)^2;

```

```

    Dx2x3=Dx2x3+(x(i,2)-x2Aver)*(x(i,3)-x3Aver);
    Dx3x3=Dx3x3+(x(i,3)-x3Aver)^2;
    SyAver=SyAver+(y(x(i,1))-yAver)^2;
end;
Sxx(1,1)=Dx1x1; Sxx(1,2)=Dx1x2; Sxx(1,3)=Dx1x3;
Sxx(2,1)=Dx1x2; Sxx(2,2)=Dx2x2; Sxx(2,3)=Dx2x3;
Sxx(3,1)=Dx1x3; Sxx(3,2)=Dx2x3; Sxx(3,3)=Dx3x3;

Sxx_1=1/Sxx;

z1Aver=0; z2Aver=0; z3Aver=0;
for i=1:n
    z1Aver=z1Aver+log10(x(i,1)); z2Aver=z2Aver+log10(x(i,2));
    z3Aver=z3Aver+log10(x(i,3));
end;
z1Aver=z1Aver/n; z2Aver=z2Aver/n; z3Aver=z3Aver/n;
Dz1z1=0; Dz2z2=0; Dz3z3=0;
Dz1z2=0; Dz1z3=0; Dz2z3=0;
for i=1:n
    Dz1z1=Dz1z1+(log10(x(i,1))-z1Aver)^2;
    Dz1z2=Dz1z2+(log10(x(i,1))-z1Aver)*(log10(x(i,2))-z2Aver);
    Dz1z3=Dz1z3+(log10(x(i,1))-z1Aver)*(log10(x(i,3))-z3Aver);
    Dz2z2=Dz2z2+(log10(x(i,2))-z2Aver)^2;
    Dz2z3=Dz2z3+(log10(x(i,2))-z2Aver)*(log10(x(i,3))-z3Aver);
    Dz3z3=Dz3z3+(log10(x(i,3))-z3Aver)^2;
end;
Szz(1,1)=Dz1z1; Szz(1,2)=Dz1z2; Szz(1,3)=Dz1z3;
Szz(2,1)=Dz1z2; Szz(2,2)=Dz2z2; Szz(2,3)=Dz2z3;
Szz(3,1)=Dz1z3; Szz(3,2)=Dz2z3; Szz(3,3)=Dz3z3;

Szz_1=1/Szz;

zi(1)=log10(x1i)-z1Aver; zi(2)=log10(x2i)-z2Aver;
zi(3)=log10(x3i)-z3Aver;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

Df=n-4;
alpha=1-Pconfid;
P=1-alpha/2;
//P=0.975;
Q=1-P;

[T]=cdf('T',Df,P,Q);

b0est=-0.3354;
b1est=0.5185;
b2est=-0.4248;
b3est=0.6723;

```

```

Dzyzyn4=0; Dyy=0;
for i=1:n
    Dzyzyn4=Dzyzyn4+..

(b0est+b1est*log10(x(i,1))+b2est*log10(x(i,2))+b3est*log10(x(i,3))
-log10(yx(i,1)))^2;

    Dyy=Dyy+(yx(i,1)-
10^b0est*x(i,1)^b1est*x(i,2)^b2est*x(i,3)^b3est)^2;
end;
Dzyzyn4=Dzyzyn4/(n-4)
Delta_z=T*(Dzyzyn4*(1/n+result))^0.5;

r_2=1-Dyy/SyAver;

zy_result=b0est+b1est*log10(x1i)+b2est*log10(x2i)+b3est*log10(x3i)
;

zyLBCONF=zy_result-Delta_z; zyUBCONF=zy_result+Delta_z;

y_result=10^zy_result;
yLBCONF=10^zyLBCONF; yUBCONF=10^zyUBCONF;

file_D_5=fopen('D:\Data_F\SoftDef35_data5.txt','w+');

    fprintf(file_D_5, "%5.6f ", yUBCONF);

fclose(file_D_5);

file_D_4=fopen('D:\Data_F\SoftDef35_data4.txt','w+');
for i = 1:3
    for j = 1:3 fprintf(file_D_4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_D_4, "\n");
end;
fclose(file_D_4);

set(text_y_result,'string', sprintf("=%1.4f",y_result));
set(text_yLBCONF,'string', sprintf("=%1.4f",yLBCONF));
set(text_yUBCONF,'string', sprintf("=%1.4f",yUBCONF));
set(text_r_2,'string',..
    sprintf("Коефіцієнт детермінації R^2=%1.4f",r_2));

if (get(radio_bikv_x,'value')==0)

else
    yfact=eval(get(edit_text12,'string'));
    MRE=abs((yfact-y_result)/yfact);
    text_MRE=uicontrol(f,'style','text','string','', 'position',...
    [500,120,160,20]);
    set(text_MRE,'string',..
    sprintf("Відносна похибка MRE=%1.4f",MRE));

```

```
end;  
  
endfunction  
  
function Wind Close()  
  //закриття вікон  
  
  close(f3);  
  close(f2);  
  close(f1);  
  close(f);  
  
endfunction
```

ДОДАТОК В – Опис програми

Загальні відомості

Програма для оцінювання кількості дефектів ПЗ для системи моделювання Scilab 6.0.0. Файл програми SoftDefEst.sce має розмір 11 КБайт.

Функціональне призначення

Програма SoftDefEst призначений для оцінювання кількості дефектів ПЗ для системи моделювання Scilab 6.0.0.

Технічні засоби, що використовуються

Згідно технічного завдання вимоги до складу й параметрів технічних засобів є такими: процесор з інструкціями SSE2, оперативна пам'ять (RAM) не менше 2 Гб, простір на жорсткому диску 600 Мб мінімум.

Виклик та завантаження

Вимоги до інформаційної та програмної сумісності є такими: операційна система Windows 10 (або вища), система моделювання Scilab 6.0.0 (або вища), компілятор C (Visual Studio 2010 або Visual Express 2010 або вищі).

Для запуску та роботи з програмою необхідна операційна система (ОС) Windows 10 (або вища), система моделювання Scilab 6.0.0 (або вища).

Завантаження програми здійснюється шляхом відкриття файлу SoftDefEst.sce стандартними засобами ОС.

ДОДАТОК Г – Інструкція користувача

Загальний опис роботи з ПЗ

Після завантаження файлу SoftDefEst.sce на екрані з'являється графічне вікно інтерфейсу початкового стану програми для оцінювання кількості дефектів ПЗ (рисунок Г.1).

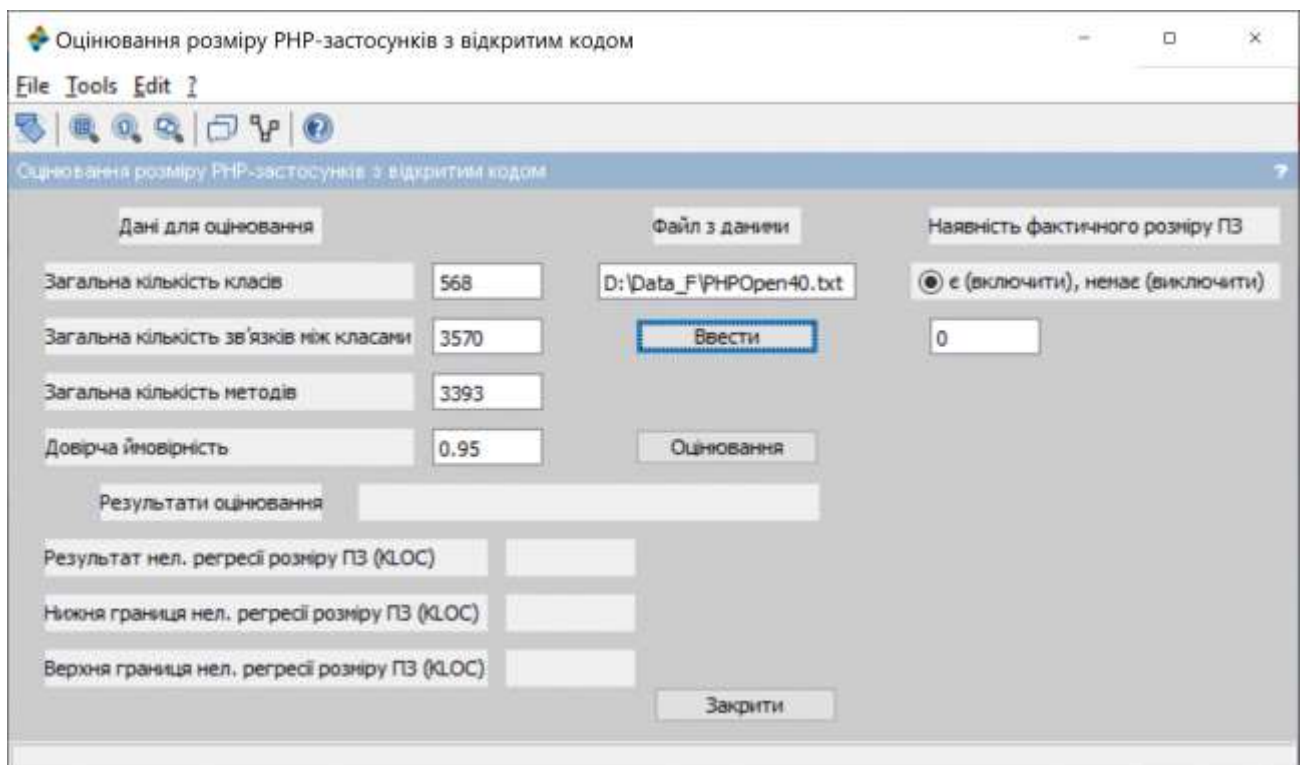


Рисунок Г.1 – Графічне вікно інтерфейсу початкового стану програми

Далі у вікні редагування «Файл з даними» графічного вікна «Оцінювання кількості дефектів ПЗ» (рис.Г.1) потрібно відредагувати ім'я текстового файлу з початковими емпіричними даними, що будуть використовуватися при роботі програми.

Після натискання на командну кнопку «Ввести» з текстового файлу вводяться такі дані: фактична кількість дефектів ПЗ, кількість тисяч строк коду

(KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ, кількість тестових випадків), які виводяться у окремих графічних вікнах.

Ділі у графічному вікні «Оцінювання кількості дефектів ПЗ» (рис. Г.1) у відповідних вікнах редагування потрібно відредагувати: кількість тисяч строк коду (KLOC); загальні зусилля (в днях) на етапах до тестування ПЗ, кількість тестових випадків, довірчу ймовірність.

Крім того у графічному вікні «Оцінювання кількості дефектів ПЗ» (рис. Г.1) потрібно за допомогою радіо кнопки здійснити вибір наявності фактичної кількості дефектів ПЗ та ввести її для подальшого порівняння з відповідною оцінкою кількості дефектів ПЗ.

Далі потрібно натиснути на командну кнопку «Оцінювання». Після цього з'являться результати оцінювання кількості дефектів ПЗ.

І на останок у графічному вікні «Оцінювання кількості дефектів ПЗ» головного модуля програми (рис. Г.1) потрібно натиснути на командну кнопку «Закрити». Це приведе до закриття всіх графічних вікон та завершення програми.

ДОДАТОК Д – Програма та методика випробувань

Об’єкт випробувань: Програма для оцінювання кількості дефектів ПЗ SoftDefEst.

Ціль випробувань: перевірка відповідності програми SoftDefEst до вимог технічного завдання і повноти реалізації.

Склад програмної документації:

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

Технічні вимоги.

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: процесор Pentium IV (або еквівалентний) з інструкціями SSE2, оперативна пам’ять 2 Гб (1 Гб мінімум), простір на жорсткому диску 600 Мб мінімум.

Методи випробувань.

Випробування програми виконується представником замовника в присутності представника розробника. Відповідно до опису програми виконується запуск програми SoftDefEst. Далі відбувається перевірка роботи програми. Виконується перевірка всіх графічних вікон програми, а також всіх допустимих для виконання операцій всередині графічних вікон згідно з інструкцією користувача. Якщо програма демонструє адекватне функціонування згідно до вимог ТЗ, то робота з розробки програми вважається виконаною, і здійснюється впровадження (введення) програми до експлуатації. В протилежному випадку програма має бути відправлено на доробку розробникові.

При випробуванні програми використовуються тести за методологією «чорної скриньки» (рис.Д.1-Д.5).

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання

Кількість тисяч строк коду ПЗ (KLOC) 12.3

Загальні зусилля (в днях) 17

Загальна кількість тестових випадків 142

Довірча ймовірність 0.95

Файл з даними

D:\Data_F\SoftDef35.txt

Ввести

Наявність фактичної кількості дефектів

☒ є (включити), немає (виключити)

0

Оцінювання

Результати оцінювання

Результат нел. регресії кількості дефектів

Нижня границя нел. регресії кількості дефектів

Верхня границя нел. регресії кількості дефектів

Закрити

Рисунок Д.1 – Графічне вікно головного модуля програми

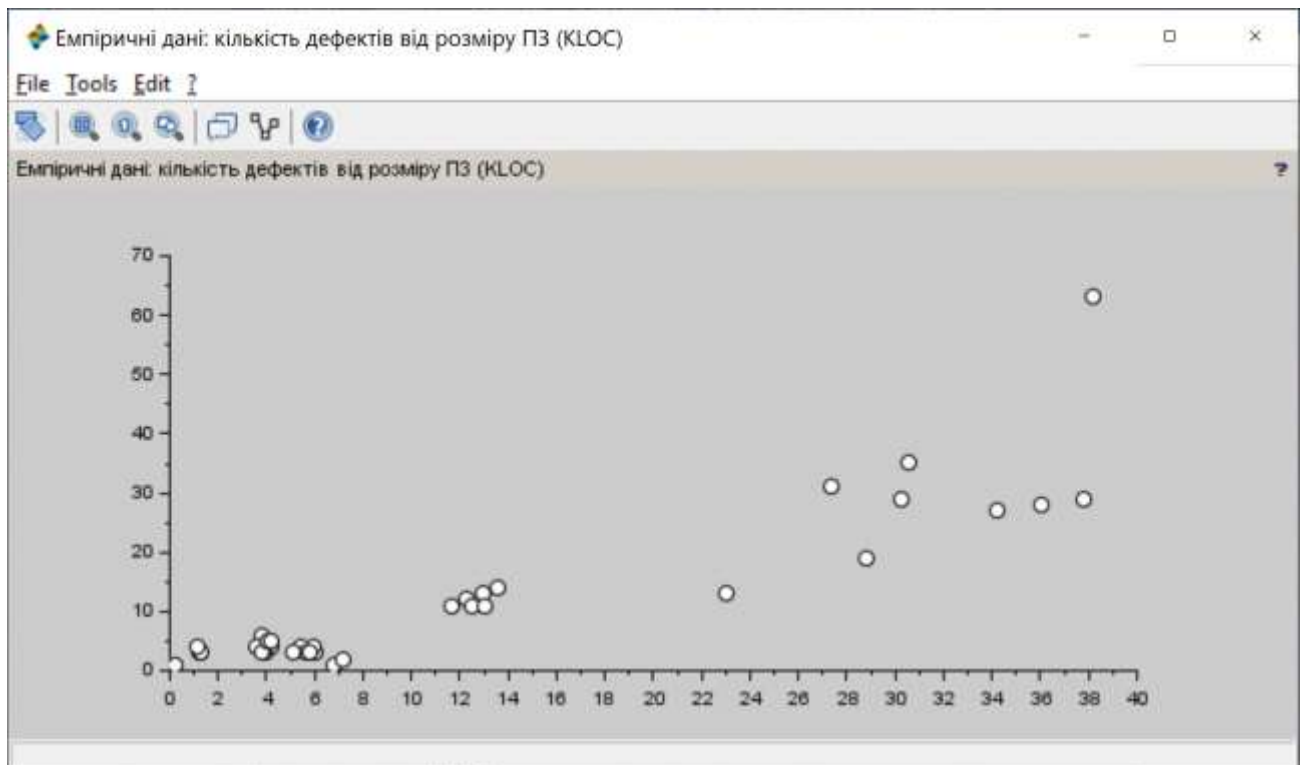


Рисунок Д.2 – Візуалізація емпіричних даних.

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання

Кількість тисяч строк коду ПЗ (KLOC) 12.3

Загальні зусилля (в днях) 17

Загальна кількість тестових випадків 142

Довірча ймовірність 0.95

Файл з даними D:\Data_F\SoftDef35.txt

Наявність фактичної кількості дефектів ☐ є (включити), немає (виключити)

Ввести

Оцінювання

Результати оцінювання Коефіцієнт детермінації $R^2=0.6639$

Результат нел. регресії кількості дефектів =14.2562

Нижня границя нел. регресії кількості дефектів =12.2939

Верхня границя нел. регресії кількості дефектів =16.5318

Закрити

Рисунок Д.3 – Результати роботи програми після натискання на кнопку «Оцінювання» у разі, коли радіо кнопка виключена

Оцінювання кількості дефектів ПЗ

File Tools Edit ?

Оцінювання кількості дефектів ПЗ

Дані для оцінювання

Кількість тисяч строк коду ПЗ (KLOC) 12.3

Загальні зусилля (в днях) 17

Загальна кількість тестових випадків 142

Довірча ймовірність 0.95

Файл з даними

D:\Data_F\SoftDef35.txt

Ввести

Наявність фактичної кількості дефектів

☒ є (включити), ☐ немає (виключити)

12

Оцінювання

Результати оцінювання

Коефіцієнт детермінації $R^2=0.6639$

Відносна похибка MRE=0.1880

Результат нел. регресії кількості дефектів =14.2562

Нижня границя нел. регресії кількості дефектів =12.2939

Верхня границя нел. регресії кількості дефектів =16.5318

Закрити

Рисунок Д.4 – Результати роботи програми після натискання на кнопку «Оцінювання» у разі, коли радіо кнопка включена та введена фактична кількість дефектів ПЗ, що дорівнює 12

Рисунок Д.5 – Результати роботи програми після натискання на командну кнопку «Оцінювання» у разі, коли радіо кнопка включена та введене значення довірчої ймовірності 0,99

При випробуванні програми потрібно ввести наступні дані із текстового файлу:

```

19 28.8 16.79 224
1 6.8 45.69 17
4 5.4 13.44 24
3 1.2 4.72 28
3 4 64 149
1 0.2 5.63 24
28 36 89.42 306
12 12.3 17 142
6 3.8 8.86 40
29 30.24 17.63 235
2 7.14 47.98 18
3 5.67 14.11 25
3 1.26 4.96 29
4 4.21 67.21 156
29 37.79 93.89 321
13 12.92 17.85 149
4 3.99 9.3 42
11 12.49 15.02 139

```

13 22.99 39.1 150
4 3.61 8.42 38
31 27.36 15.95 213
4 1.14 4.48 27
3 5.13 12.76 23
3 3.81 60.8 142
5 4.06 8.24 41
11 11.68 16.15 135
27 34.21 84.95 291
3 6.01 13.68 26
4 5.94 12.1 26
11 13.04 16.49 147
35 30.52 16.28 233
14 13.53 15.3 156
5 4.18 7.98 44
3 5.77 12.5 25
63 38.16 86.73 318