

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

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

«Допущений до захисту»
Завідувач кафедри
_____ проф. Приходько С.Б.

«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

на тему: Математична модель для моделювання трудомісткості agile
тестування невеликих веб проєктів і розробка програмного забезпечення для її
реалізації

Виконав: студент групи 6151м

(підпис, ПІБ) Сорокіна Є.В.

Керівник роботи:
Доцент кафедри ПЗАС, к.т.н.
(посада, науковий ступень вчене звання)

(підпис, ПІБ) Пухалевич А.В.

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
 Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми
 _____ проф. С.Б.Приходько
 (підпис)
 «10» жовтня 2022 р.

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

Студенту Сорокіній Євгенії Володимирівні _____
 (Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель для моделювання трудомісткості agile тестування невеликих веб проєктів і розробка програмного забезпечення для її реалізації _____

Керівник роботи Пухалевич А.В. _____

Затверджені наказом ректора № 798 уч від «28» _____ 09 _____ 2022 р.

2. Термін подання роботи: 14.11.2022 р. _____

3. Вихідні дані по роботі: _____

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

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

5. Перелік презентаційних матеріалів: Актуальність теми. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації. Існуючі нелінійні регресійні моделі для моделювання трудомісткості agile тестування невеликих веб проєктів. Нелінійна регресійна модель для моделювання трудомісткості agile тестування невеликих веб проєктів. Постановка задачі на розробку проекту програми. Діаграма варіантів використання. Діаграма діяльності. Графічне вікно програми із інтерфейсними компонентами. Емпіричні дані. Результати випробувань програми. Висновки. _____

РЕФЕРАТ

Сорокіна Євгенія Володимирівна

«Математична модель для моделювання трудомісткості agile тестування невеликих веб проєктів і розробка програмного забезпечення для її реалізації»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2022 р.

Обсяг роботи: 107 стор., 8 табл., 9 рис., 23 використаних джерела, 5 додатків.

Актуальність теми роботи: необхідність підвищення достовірності моделювання трудомісткості agile тестування невеликих веб проєктів.

Мета та завдання дослідження. Метою є підвищення достовірності моделювання трудомісткості agile тестування невеликих веб проєктів. Завдання дослідження: проаналізувати існуючі регресійні моделі для моделювання трудомісткості agile тестування невеликих веб проєктів; удосконалити регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів; розробити програму для моделювання трудомісткості agile тестування невеликих веб проєктів.

Об'єктом дослідження є процес моделювання трудомісткості agile тестування невеликих веб проєктів.

Предметом дослідження є регресійні моделі для моделювання трудомісткості agile тестування невеликих веб проєктів.

Методи дослідження. У вирішенні поставлених завдань використано методи теорії ймовірності, математичної статистики, регресійного аналізу.

Наукова новизна одержаних результатів. Удосконалено регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів, на основі нормалізуючого перетворення у формі десяткового логарифму, яка дозволяє підвищити достовірність моделювання трудомісткості agile тестування невеликих веб проєктів, в порівнянні з існуючими регресійними моделями.

Практичне значення одержаних результатів. Розроблено програму для моделювання трудомісткості agile тестування невеликих веб проєктів, для системи моделювання Scilab 6.0.0.

Апробація результатів досліджень. Результати досліджень були оприлюднені на Всеукраїнській науково-практичній інтернет-конференції.

Публікації. Результати роботи викладені у 1 публікації, а саме: 1 матеріалах Всеукраїнської науково-практичної інтернет-конференції.

Ключові слова: трудомісткість agile тестування, веб проєкт, моделювання, нелінійна регресійна модель, програмне забезпечення.

ABSTRACT

Sorokina Yevgenia Volodimirivna

«A mathematical model for simulating the agile testing efforts for small Web projects and developing the software for its implementation»

The qualification work in obtaining a master's degree in specialty 121 – "Software Engineering". Admiral Makarov National University of Shipbuilding. Mykolayiv, 2022.

The qualification work is presented on the 107 pages of typewritten text, contains 8 tables, 9 figures, 5 appendices and 23 references.

Relevance of the topic of the work: the need to increase the confidence of simulating the agile testing efforts for small Web projects.

The goal and objectives of the study. The goal of the study is to increase the confidence of simulating the agile testing efforts for small Web projects by nonlinear regression model. The objectives of the study: to analyze existing mathematical models for simulating the agile testing efforts for small Web projects; to improve the regression model for simulating the agile testing efforts for small Web projects; to develop the software for simulating the agile testing efforts for small Web projects.

The object of the study is the process of simulating the agile testing efforts for small Web projects.

The subject of the study is the regression models for simulating the agile testing efforts for small Web projects.

The methods of the study. The methods of probability theory, mathematical statistics, regression analysis are used in solving the problems.

The scientific novelty of obtained results. The nonlinear regression model for simulating the agile testing efforts for small Web projects is improved based on the normalizing transformation in the decimal logarithm form which allows to increase confidence of simulating the agile testing efforts for small Web projects in comparison with existing nonlinear regression models.

The practical significance of obtained results is that the software realizing the constructed model is developed for the simulation system of Scilab 6.0.0.

Approbation of study results. The study results were presented at the All-Ukrainian scientific Internet-conference.

Publications. The study results were published in one publication: the materials of the All-Ukrainian scientific Internet-conference.

Keywords: agile testing efforts, Web project, simulation, nonlinear regression model, software.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ AGILE ТЕСТУВАННЯ НЕВЕЛИКИХ ВЕБ ПРОЄКТІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ЗА ОБРАНОЮ ТЕМОЮ	12
1.1 Аналіз існуючих моделей для моделювання трудомісткості agile тестування невеликих веб проєктів	12
1.2 Обґрунтування необхідності проведення досліджень за обраною темою	16
1.3 Висновки за розділом 1	17
2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ AGILE ТЕСТУВАННЯ НЕВЕЛИКИХ ВЕБ ПРОЄКТІВ	19
2.1 Перевірка даних для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів на наявність викидів	19
2.2 Побудова нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів	23
2.3 Висновки за розділом 2	30

3 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ AGILE ТЕСТУВАННЯ НЕВЕЛИКИХ ВЕБ ПРОЄКТІВ	32
3.1 Постановка задачі на розробку проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів	32
3.2 Розробка ескізного проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів	34
3.3 Розробка технічного проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів	41
3.4 Розробка робочого проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів	44
3.5 Висновки за розділом 3	54
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМИ	55
4.1 Розрахунок витрат на створення й експлуатацію програми	55
4.2 Економічна ефективність розробки і впровадження програми	58
5 ОХОРОНА ПРАЦІ	61
5.1 Аналіз існуючої проблеми охорони праці	61
5.2 Аналіз шкідливих та небезпечних факторів у офісному приміщенні з персональними комп'ютерами	62
5.3 Розрахунок системи пожежогасіння офісного приміщення	67
5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів	69
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	72
6.1 Аналіз існуючої проблеми необхідності охорони навколишнього середовища	72

6.2 Утилізація відходів комп'ютерної техніки	75
ВИСНОВКИ	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	82
ДОДАТОК А Технічне завдання на розробку програми для моделювання трудомісткості agile тестування невеликих веб проєктів	87
ДОДАТОК Б Текст програми	91
ДОДАТОК В Опис програми	99
ДОДАТОК Г Інструкція користувача	100
ДОДАТОК Д Програма та методика випробувань	102

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

ПЗ – програмне забезпечення;

LB – lower bound;

MD – the Mahalanobis distance;

MRE – a magnitude of relative error;

MMRE – a mean magnitude of relative error;

PRED – percentage of prediction;

UB – upper bound.

ВСТУП

Актуальність теми. Прогнозування трудомісткості тестування програмного забезпечення (ПЗ) є однією з важливих задач розробки ПЗ. Agile тестування — це процес тестування ПЗ, який відповідає принципам гнучкої розробки ПЗ [1-5]. У порівнянні з каскадним тестуванням, це відносно новий підхід до тестування ПЗ, який призводить до зменшення трудомісткості тестування. Agile тестування добре підходить для невеликих програмних проєктів, включаючи невеликі веб проєкти.

Життєвий цикл agile тестування складається з 5 етапів [5], другим з яких є планування agile тестування, яке включає прогнозування трудомісткості тестування. У свою чергу прогнозування трудомісткості тестування є складною задачею, для вирішення якої застосовуються різноманітні математичні моделі, включаючи і регресійні [6, 7]. Зазначимо, що регресійні моделі містять у своєму складі випадкові складові, що дозволяє здійснювати моделювання трудомісткості тестування програмних проєктів як випадкової величини. В [7] було побудовано декілька нелінійних регресійних моделей для оцінювання трудомісткості agile тестування невеликих веб проєктів, але не були наведені значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки. А це не дозволяє здійснювати моделювання трудомісткості agile тестування. Тому виникає потреба в удосконаленні зазначених моделей для моделювання трудомісткості agile тестування невеликих веб проєктів саме як випадкової величини, що може бути зроблено за допомогою методу, який наведений у [8, 9].

Мета і завдання дослідження. Метою роботи є підвищення достовірності моделювання трудомісткості agile тестування невеликих веб проєктів.

Для досягнення поставленої мети потрібно вирішити такі завдання.

- Проаналізувати існуючі нелінійні регресійні моделі для моделювання трудомісткості agile тестування невеликих веб проєктів;
- Здійснити підбор сукупності даних, за якими буде будуватися нелінійна регресійна модель для моделювання трудомісткості agile тестування невеликих веб проєктів та перевірити ці дані на наявність двовимірних викидів;
- Удосконалити нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів на основі нормалізуючого перетворення;
- На підставі запропонованої нелінійної регресійної моделі розробити програму для моделювання трудомісткості agile тестування невеликих веб проєктів.

Об'єктом дослідження є процес моделювання трудомісткості agile тестування невеликих веб проєктів.

Предметом дослідження є регресійні моделі для моделювання трудомісткості agile тестування невеликих веб проєктів.

Методи дослідження. У вирішенні поставлених завдань використано методи теорії ймовірності, математичної статистики, регресійного аналізу.

Наукова новизна одержаних результатів. Удосконалено нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів в залежності від кількості функціональних точок на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у регресії. Для цієї нелінійної регресійної моделі у порівнянні з

існуючими регресійними нелінійними моделями визначено оцінку середньо квадратичного відхилення випадкової величини, яка описує похибку, що дає змогу здійснювати моделювання трудомісткості agile тестування невеликих веб проєктів.

Практичне значення одержаних результатів. Розроблено програму для моделювання трудомісткості agile тестування невеликих веб проєктів. Програму створено scі-мовою для системи Scilab. Програму, що розроблено, можна використовувати для прогнозування трудомісткості agile тестування невеликих веб проєктів на ранніх стадіях.

Особистий внесок здобувача. У публікації [10], що зроблена у співавторстві, автору належить нелінійна регресійна модель для моделювання трудомісткості agile тестування невеликих веб проєктів на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у регресії.

Апробація результатів досліджень. Результати досліджень, викладені у роботі, були оприлюднені на одній конференції: III-ій Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2022)» (Миколаїв, 26-28 жовтня 2022 р.).

Публікації. За результатами кваліфікаційної роботи зроблено одну публікацію, а саме: матеріали Всеукраїнської науково-практичної інтернет-конференції.

1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ AGILE ТЕСТУВАННЯ НЕВЕЛИКИХ ВЕБ ПРОЄКТІВ, ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ЗА ОБРАНОЮ ТЕМОЮ

1.1 Аналіз існуючих моделей для моделювання трудомісткості agile тестування невеликих веб проєктів

Життєвий цикл agile тестування складається з 5 етапів [5], другим з яких є планування agile тестування, яке включає прогнозування трудомісткості тестування. У свою чергу прогнозування трудомісткості тестування є складною задачею, для вирішення якої застосовуються різноманітні математичні моделі, включаючи і регресійні [6, 7]. В [6] було побудовано низка регресійних рівнянь, як лінійних так і нелінійних, для оцінювання трудомісткості тестування програмного забезпечення в залежності від різної кількості факторів: від одного до чотирьох. Але регресійні рівняння, на відміну від регресійних моделей не містять у своєму складі випадкові складові, що не дозволяє здійснювати моделювання трудомісткості тестування програмних проєктів як випадкової величини. В [7] було побудовано декілька нелінійних регресійних моделей для оцінювання трудомісткості agile тестування невеликих веб проєктів, але не були наведені значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки. А це не дозволяє здійснювати моделювання трудомісткості agile тестування.

Розглянемо запропоновані в [7] нелінійні регресійні моделі. Всі ці моделі дозволяють визначити трудомісткість agile тестування (у годинах), яку

позначено як Y , в залежності від трьох факторів: кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

В [7] було побудовано дві нелінійні регресійні моделі за даними з 40 проєктів. Перша модель в [7] була побудована за перетворенням у вигляді десятичного логарифму

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

де оцінки параметрів є такими: $\hat{b}_0 = -0.4500$, $\hat{b}_1 = 0.78887$, $\hat{b}_2 = 0.00176$, $\hat{b}_3 = 0.18782$; ε повинна бути випадковою величиною з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2$.

Друга модель в [7] була побудована за перетворенням Джонсона для сімейства S_B

$$Y = \hat{\phi}_Y + \hat{\lambda}_Y / \left[1 + e^{-(\hat{Z}_Y + \varepsilon - \hat{\gamma}_Y) / \hat{\eta}_Y} \right], \quad (1.2)$$

де $Z_j = \gamma_j + \eta_j \ln \frac{X_j - \phi_j}{\phi_j + \lambda_j - X_j}$, $\phi_j < X_j < \phi_j + \lambda_j$, $j = 1, 2, 3$, $\hat{Z}_Y$ – оцінка

середньої нормалізованої трудомісткості за лінійним регресійним рівнянням

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \hat{b}_2 Z_2 + \hat{b}_3 Z_3 + \varepsilon, \quad (1.3)$$

де ε повинна бути випадковою величиною з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2$.

Моделі (1.2) та (1.3) були побудовані для двох варіантів перетворення Джонсона сімейства S_B : одновимірного та чотиривимірного. Для одновимірного перетворення Джонсона сімейства S_B оцінки параметрів для моделей (1.2) та (1.3) є такими: $\hat{\gamma}_Y = 1.72684$, $\hat{\gamma}_1 = 1.78876$, $\hat{\gamma}_2 = 1.2346$, $\hat{\gamma}_3 = 1.1127$, $\hat{\eta}_Y = 0.73211$, $\hat{\eta}_1 = 0.73090$, $\hat{\eta}_2 = 0.72269$, $\hat{\eta}_3 = 0.53444$, $\hat{\phi}_Y = 1.1292$, $\hat{\phi}_1 = 3.3512$, $\hat{\phi}_2 = 1.1717$, $\hat{\phi}_3 = 2.90$, $\hat{\lambda}_Y = 64.039$, $\hat{\lambda}_1 = 237.48$, $\hat{\lambda}_2 = 142.95$, $\hat{\lambda}_3 = 92.745$, $\hat{b}_0 = 0$, $\hat{b}_1 = 0.86376$, $\hat{b}_2 = 0.00240$ та $\hat{b}_3 = 0.16633$.

Для чотиривимірного перетворення Джонсона сімейства S_B оцінки параметрів для моделей (1.2) та (1.3) є такими: $\hat{\gamma}_Y = 3.8484$, $\hat{\gamma}_1 = 5.4050$, $\hat{\gamma}_2 = 1.0397$, $\hat{\gamma}_3 = 1.3214$, $\hat{\eta}_Y = 0.9990$, $\hat{\eta}_1 = 0.96416$, $\hat{\eta}_2 = 0.68334$, $\hat{\eta}_3 = 0.61537$, $\hat{\phi}_Y = 0.52944$, $\hat{\phi}_1 = 2.0172$, $\hat{\phi}_2 = 1.0107$, $\hat{\phi}_3 = 2.5590$, $\hat{\lambda}_Y = 298.41$, $\hat{\lambda}_1 = 5439.41$, $\hat{\lambda}_2 = 128.66$, $\hat{\lambda}_3 = 103.668$, $\hat{b}_0 = 0$, $\hat{b}_1 = 0.81961$, $\hat{b}_2 = 0.01428$, $\hat{b}_3 = 0.20730$.

Значення показників R^2 , MMRE та PRED(0.25) дорівнюють відповідно 0.9810, 0.0503 та 1.0 для моделі (1.1), та дорівнюють відповідно 0.9828, 0.0478 та 1.0 для моделі (1.2) з оцінками параметрів для одновимірного перетворення Джонсона сімейства S_B , та дорівнюють відповідно 0.9818, 0.0443 та 1.0 для моделі (1.2) з оцінками параметрів для чотиривимірного перетворення Джонсона сімейства S_B . Значення MMRE краще для моделі (1.2) з оцінками параметрів для чотиривимірного перетворення Джонсона сімейства S_B у порівнянні з іншими попередніми нелінійними регресійними моделями.

Нагадаємо, що середня величина відносної похибки MMRE та відсоток прогнозування на рівні величини відносної помилки 0,25, PRED(0,25) зазвичай застосовуються для оцінювання якості прогнозування за допомогою регресійних моделей і в інженерії програмного забезпечення [11, 12]. Допустимі

значення $MMRE$ і $PRED(0,25)$ складають не більше 0,25 і не менше 0,75 відповідно.

Значення величини відносної похибки MRE обчислюється як

$$MRE_i = \left| \frac{(Y_i - \hat{Y}_i)}{Y_i} \right|.$$

Середня величина відносної похибки $MMRE$ визначається як

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i. \quad (1.4)$$

Зазвичай $MMRE \leq 0,25$ вважається прийнятною точністю моделі.

Значення $PRED(0,25)$ обчислюється як

$$PRED(0,25) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1 & \text{if } MRE_i \leq 0,25 \\ 0 & \text{otherwise.} \end{cases} \quad (1.5)$$

Зазвичай $PRED(0,25) \geq 0,75$ вважається прийнятною точністю прогнозування за допомогою регресійних моделей.

Вибірковий коефіцієнт детермінації R^2 визначається як

$$R^2 = 1 - \frac{SS_E}{SS_T}, \quad (1.6)$$

де SS_E – сума квадратів залишків (the residual sum of squares) або сума квадратів

помилки (the error sum of squares), $SS_E = \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$; SS_T – загальна сума

квадратів (the total sum of squares), $SS_T = \sum_{i=1}^N (Y_i - \bar{Y})^2$. Відомо, що

$SS_T = SS_R + SS_E$. Тут SS_R – сума квадратів регресії (the regression sum of squares),

$$SS_R = \sum_{i=1}^N (\hat{Y}_i - \bar{Y})^2.$$

Коефіцієнт детермінації R^2 інтерпретується як частка мінливості у спостережуваній залежній змінній Y , що пояснюється моделлю лінійної регресії. Значення R^2 вказує наскільки емпіричні дані підтверджують математичну модель або, чи краща наша модель за вибіркове середнє. З формули для R^2 , $0 \leq R^2 \leq 1$, значення R^2 вважається прийнятним, якщо воно більше за 0,75. Велике значення R^2 свідчить про те, що модель успішно пояснює мінливість Y . Коли значення R^2 є малим, то це може свідчити про те, що потрібно знайти альтернативну модель, яка може врахувати більшу мінливість Y .

1.2 Обґрунтування необхідності проведення досліджень за обраною темою

Хоча в [7] було побудовано декілька нелінійних регресійних моделей для оцінювання трудомісткості agile тестування невеликих веб проєктів з добрими показниками якості прогнозування, але не були наведені значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки у цих моделях. А це не дозволяє здійснювати моделювання трудомісткості agile тестування невеликих веб проєктів. Тому виникає потреба в побудові відповідних моделей для моделювання трудомісткості agile тестування невеликих веб проєктів.

Побудову нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів ми будемо за методом [9], що заснований на багатовимірних нормалізуючих перетвореннях та інтервалах прогнозування. Суть цього методу [9] є такою. На першому етапі початкові дані перевіряються на наявність викидів і, якщо останні знайдено, то вони

відкидаються. Для цього використовується критерій на основі квадрату відстані Махаланобіса для нормалізованих даних із 0,005 рівнем значущості. На другому етапі будується нелінійна регресійна модель із використанням відповідного методу на основі нормалізуючих перетворень [13]. На третьому етапі визначаються границі інтервалу прогнозування нелінійної регресії для рівня значущості, що дорівнює 0,05, за відповідним методом [13]. На останньому четвертому етапі перевіряють, чи є серед даних, за якими будувалася нелінійна регресійна модель такі, що виходять за границі інтервалу прогнозування. Та, якщо відповідні дані знайдено, то вони відкидаються, і ми повторюємо знову всі етапи, починаючи з першого, для нових даних. Якщо таких викидів не було, то повторення етапів завершується, відповідна нелінійна регресійна модель побудована.

1.3 Висновки за розділом 1

У даному розділі наведено огляд існуючих математичних моделей для моделювання трудомісткості agile тестування невеликих веб проєктів, здійснено обґрунтування необхідності проведення досліджень за обраною темою.

1) Зараз для моделювання трудомісткості agile тестування невеликих веб проєктів в першу чергу використовуються методи нелінійного регресійного аналізу.

2) Для моделювання трудомісткості agile тестування невеликих веб проєктів існують нелінійні регресійні моделі, які побудовані в залежності від трьох факторів: кількість тестових випадків, кількість сторінок програмно-технічної документації та кількість дефектів. Але у зазначених моделях не були

наведені значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки. А це не дозволяє здійснювати моделювання трудомісткості agile тестування. Тому виникає потреба в побудові відповідних моделей для моделювання трудомісткості agile тестування невеликих веб проєктів.

3) Побудову нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів, краще здійснювати за методом, що заснований на нормалізуючих перетвореннях та інтервалах прогнозування. Тому удосконалювати нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів доцільно саме на основі зазначеного методу.

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ AGILE ТЕСТУВАННЯ НЕВЕЛИКИХ ВЕБ ПРОЄКТІВ

2.1 Перевірка даних для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів, на наявність викидів

Для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів використовується вибірка чотиривимірних негаусових даних за такими метриками: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 . Ці дані з 41 невеликих веб проєктів наведені в табл.2.1 [7].

Ми перевірили чотиривимірні дані метрик з табл.2.1 на нормальність за допомогою тесту Мардіа (Mardia's test) [14], якій базується на багатовимірних асиметрії та ексцесі. Тест показав, що розподіл даних не є гаусівським. Тому для визначення наявності викидів у даних ми використовували тест на основі квадрату відстані Махаланобіса (Mahalanobis Distance) для нормалізованих за десятковим логарифмом даних як це рекомендується робити у роботах [15, 16]. При цьому, як рекомендується [15-17], ми брали рівень значимості 0,005.

Таблиця 2.1

Набір даних та квадрат відстані Махаланобіса (MD)

№	Y	X_1	X_2	X_3	квадрат MD	
					без перетворення	з перетворенням
1	2	3	4	5	6	7
1	1,33	4	11	4	1,02	4,42
2	3,51	11	2	7	1,45	6,70
3	3,17	10	4	8	1,19	3,31
4	1,53	5	14	4	0,92	4,15
5	2,35	9	12	3	1,09	3,02
6	3,13	10	8	9	0,91	2,31
7	2,03	6	16	7	0,73	3,62
8	3,22	11	11	6	0,91	1,17
9	2,73	10	16	5	0,80	2,18
10	4,65	16	13	7	0,85	0,95
11	5,52	21	12	4	1,47	6,96
12	2,75	7	23	12	0,63	4,21
13	6,93	25	6	13	1,30	2,88
14	3,73	14	29	4	0,69	2,44
15	5,08	18	22	7	0,55	0,81
16	7,12	25	11	11	1,05	1,15
17	3,24	10	31	9	0,64	2,14
18	4,05	12	35	6	0,96	3,94
19	4,72	16	32	7	0,55	1,04
20	3,49	13	41	4	1,10	2,70
21	6,03	18	24	18	0,25	0,79
22	4,13	12	40	13	1,03	2,52
23	7,07	11	13	51	6,44	20,26
24	10,18	33	15	23	0,56	0,54
25	9,95	36	29	12	0,73	1,45

Продовження табл.2.1

1	2	3	4	5	6	7
26	8,67	24	36	32	1,31	2,23
27	16,53	51	3	45	3,05	7,75
28	12,45	44	37	18	0,59	1,00
29	15,56	56	25	19	2,35	2,32
30	17,47	57	7	40	2,76	3,95
31	11,23	29	48	47	4,66	3,98
32	8,21	29	79	10	4,55	2,92
33	19,95	50	6	90	14,66	7,01
34	16,16	44	53	61	6,83	3,76
35	8,83	33	110	8	11,14	5,42
36	12,97	47	94	16	6,12	3,11
37	21,32	94	48	37	13,19	6,62
38	20,97	83	61	62	12,43	9,48
39	38,22	127	25	78	13,37	4,27
40	26,48	111	71	54	14,99	7,33
41	48,2	173	120	60	24,17	7,18

Значення квадрату відстані Махаланобіса, яку далі позначаємо як MD, для кожної багатовимірної точки даних i , $i = 1, 2, \dots, N$, визначається як

$$d_i^2 = (\mathbf{Z}_i - \bar{\mathbf{Z}})^T S_N^{-1} (\mathbf{Z}_i - \bar{\mathbf{Z}}), \quad (2.1)$$

де $\bar{\mathbf{Z}}$ – вектор вибірових середніх; S_N – вибіркова коваріаційна матриця

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

Тут $\mathbf{Z}_i$ – гаусівський випадковий вектор $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_m\}^T$ для кожної багатовимірної точки даних i , $i = 1, 2, \dots, N$; m – кількість компонентів вектору $\mathbf{Z}$. В нашому випадку m дорівнює 4.

Вектор $\mathbf{Z}$ отримують шляхом перетворення негаусівського випадкового вектору $\mathbf{X} = \{X_1, X_2, \dots, X_m\}^T$ за допомогою багатовимірного нормалізуючого перетворення

$$\mathbf{Z} = \psi(\mathbf{X}). \quad (2.3)$$

В нашому випадку у якості перетворення (2.3) ми використовуємо перетворення у формі десяткового логарифму.

Відомо [17], якщо дані мають багатовимірний нормальний розподіл, то розподіл квадрату MD поводить ся як розподіл χ^2 . Для великих $N - m$ (принаймні більше 25), відстань d_i^2 повинна вести себе приблизно як незалежні випадкові величини $\chi_{m,\alpha}^2$. Тут $\chi_{m,\alpha}^2$ – це квантиль розподілу χ^2 з рівнем значимості α . Значення даних, для яких величина квадрату MD більша, ніж квантиль розподілу χ^2 , розглядаються як викиди, і ці значення відкидаються. Після відкидання зменшене число багатомірних точок даних знову нормалізується, використовуючи перетворення (2.3), доки всі значення квадрату MD (2.1) будуть меншими чи дорівнювати квантилю розподілу χ^2 .

Для нормалізованих за десятковим логарифмом даних (з перетворенням) є один викид (рядок 23) тому, що значення квадрата відстані Махаланобіса у ньому більше за значення квантилю розподілу χ^2 , який дорівнює 14,86 для 4

ступенів свободи та рівня значимості 0,005. При цьому за тестом Мардіа багатовимірний розподіл нормалізованих даних є гаусівським. Зазначимо, що для даних метрик з табл.2.1 без нормалізації або перетворення є два рядки даних (рядки 40 та 41), для яких значення квадрата відстані Махаланобіса є більшими за 14,86. Але дані цих рядків не можна прийняти як викиди тому, що розподіл даних метрик з табл.2.1 не є гаусівським. А як відомо [17], методи визначення викидів у багатовимірних даних на основі квадрату відстані Махаланобіса можуть застосовуватися лише у разі нормальності багатовимірного розподілу цих даних. Тому остаточно ми приймаємо, що є один чотиривимірний викид (рядок 23) в даних метрик з табл.2.1. Значення квадрата відстані Махаланобіса, які перевищують 14,86, в табл.2.1 виділено напівжирним.

В подальшому 40 рядків чотиривимірних негаусових даних, що наведені у табл.2.1 (за виключенням рядка 23), використовуються для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів за методом [9].

2.2 Побудова нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів

Для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів, ми використовуємо метод, що заснований на нормалізуючих перетвореннях та інтервалах прогнозування [9, 13], а також модифікований метод [18], який ще додатково враховує можливу наявність викидів у нелінійній регресійній моделі [19]. В

загальному випадку процес побудови нелінійної регресійної моделі за цим методом є ітераційним. За цим методом спочатку чотиривимірні негаусівські дані з табл.2.1 ми нормалізуємо за допомогою нормалізуючого перетворення, у якості якого ми використовуємо десятковий логарифм. Потім дані перевіряємо на наявність чотиривимірних викидів. Викиди відсутні тому, що значення квадрата відстані Махаланобіса (у табл.2.2 позначено як SMD_2 на другій ітерації) для всіх 40 рядків нормалізованих даних менші за значення квантилю розподілу χ^2 , який дорівнює 14,86 для 4 ступенів свободи та рівня значимості 0,005.

Далі для нормалізованих даних 40 рядків будуємо лінійну регресійну модель (1.3), в якій оцінки параметрів є такими: $\hat{b}_0 = -0.45001$, $\hat{b}_1 = 0.78887$, $\hat{b}_2 = 0.00176$, $\hat{b}_3 = 0.18782$, а оцінка середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$ випадкової величини ε дорівнює 0,0294. Параметри моделі (1.3) оцінювалися за методом найменших квадратів.

Після цього будуємо нелінійну регресійну модель (1.1) за лінійною моделлю (1.3) та перетворенням у вигляді десятичного логарифму.

Далі згідно з [9], на третьому етапі ми визначаємо границі інтервалу прогнозування нелінійної регресії для рівня значущості, що дорівнює 0,05. Згідно з [13], границі інтервалу прогнозування нелінійної регресії визначаються як

$$\psi_Y^{-1} \left(\hat{Z}_Y \pm t_{\alpha/2, v} S_{Z_Y} \left\{ 1 + \frac{1}{N} + (\mathbf{z}_X^+)^T \left[(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ \right]^{-1} (\mathbf{z}_X^+) \right\}^{1/2} \right), \quad (2.4)$$

де ψ_Y^{-1} – зворотне до нормалізуючого перетворення ψ_Y для залежної змінної Y ;
 $\hat{Z}_Y$ – результат прогнозування лінійного регресійного рівняння для значень компонент вектору $\mathbf{z}_X = \{Z_1, Z_2, \dots, Z_k\}$, що отриманий за нормалізованими даними; k – кількість незалежних змінних (у нашому випадку $k=3$); $\mathbf{Z}_X^+$ – матриця центрованих нормалізованих незалежних змінних, яка містить значення $Z_{1i} - \bar{Z}_1, Z_{2i} - \bar{Z}_2, \dots, Z_{ki} - \bar{Z}_k$; $t_{\alpha/2, v}$ – квантиль t -розподілу Стюдента в залежності від рівня значимості $\alpha/2$ та кількості ступенів свободи v ;
 $S_{Z_Y}^2 = \frac{1}{v} \sum_{i=1}^N (Z_{Yi} - \hat{Z}_{Yi})^2$, $v = N - k - 1$; $(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = k \times k$ матриця

$$(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = \begin{pmatrix} S_{Z_1 Z_1} & S_{Z_1 Z_2} & \dots & S_{Z_1 Z_k} \\ S_{Z_1 Z_2} & S_{Z_2 Z_2} & \dots & S_{Z_2 Z_k} \\ \dots & \dots & \dots & \dots \\ S_{Z_1 Z_k} & S_{Z_2 Z_k} & \dots & S_{Z_k Z_k} \end{pmatrix}, \quad (2.5)$$

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

В (2.4) знаки мінус та плюс відповідають нижній та верхній границям інтервалу прогнозування нелінійної регресії, що побудована на основі нормалізуючого перетворення.

Матриця (2.5) для нормалізованих за перетворенням у формі десяткового логарифму 40 рядків даних з табл.2.1 (за виключенням рядка 23) є такою

$$(\mathbf{Z}_X^+)^T \mathbf{Z}_X^+ = \begin{pmatrix} 6,2336 & 2,5442 & 5,4182 \\ 2,5442 & 6,9509 & 1,5112 \\ 5,4182 & 1,5112 & 7,0006 \end{pmatrix}.$$

Нижня та верхня границі інтервалу прогнозування нелінійної регресії, що отримані за допомогою перетворення у формі десяткового логарифму на другій ітерації, наведені у табл.2.2 та позначені як LB_2 та UB_2 відповідно.

Таблиця 2.2 – Границі інтервалу прогнозування нелінійної регресії, що отримані за допомогою перетворення у формі десяткового логарифму

№	Y	SMD_2	LB_2	UB_2	$\hat{Y}$	MRE
1	2	3	4	5	6	7
1	1,33	4,31	1,186	1,606	1,380	0,0375
2	3,51	6,62	2,906	3,966	3,394	0,0329
3	3,17	3,21	2,782	3,756	3,233	0,0197
4	1,53	4,12	1,417	1,912	1,646	0,0760
5	2,35	3,54	2,134	2,879	2,479	0,0549
6	3,13	2,24	2,855	3,834	3,309	0,0571
7	2,03	3,58	1,817	2,454	2,112	0,0404
8	3,22	1,16	2,857	3,829	3,308	0,0272
9	2,73	2,65	2,562	3,435	2,967	0,0867
10	4,65	0,90	3,955	5,297	4,577	0,0157
11	5,52	6,77	4,377	5,955	5,105	0,0751
12	2,75	5,70	2,264	3,080	2,641	0,0397
13	6,93	3,05	6,291	8,473	7,301	0,0535
14	3,73	2,48	3,200	4,310	3,714	0,0044
15	5,08	0,78	4,344	5,818	5,027	0,0104

Продовження табл. 2.2

1	2	3	4	5	6	7
16	7,12	1,11	6,117	8,201	7,083	0,0052
17	3,24	2,13	2,860	3,847	3,317	0,0237
18	4,05	5,23	3,064	4,112	3,550	0,1235
19	4,72	1,02	3,961	5,306	4,584	0,0288
20	3,49	2,74	3,018	4,070	3,505	0,0042
21	6,03	1,04	5,186	6,951	6,004	0,0043
22	4,13	2,86	3,535	4,768	4,106	0,0059
23	7,07	-	-	-	-	-
24	10,18	0,56	8,759	11,722	10,133	0,0046
25	9,95	1,39	8,304	11,135	9,616	0,0336
26	8,67	3,16	7,231	9,756	8,399	0,0312
27	16,53	7,75	13,802	18,916	16,158	0,0225
28	12,45	0,95	10,508	14,075	12,162	0,0232
29	15,56	2,25	12,810	17,214	14,850	0,0456
30	17,47	3,89	14,853	20,103	17,280	0,0109
31	11,23	6,04	8,997	12,224	10,487	0,0662
32	8,21	2,86	6,763	9,108	7,849	0,0440
33	19,95	9,55	15,500	21,234	18,142	0,0906
34	16,16	5,18	13,141	17,825	15,305	0,0529
35	8,83	5,26	7,160	9,712	8,339	0,0556
36	12,97	3,02	10,809	14,575	12,551	0,0323
37	21,32	9,32	21,839	29,433	25,353	0,1892
38	20,97	11,06	21,804	29,432	25,333	0,2081
39	38,22	4,30	31,740	42,984	36,936	0,0336
40	26,48	9,10	26,713	36,101	31,054	0,1727
41	48,2	7,12	38,533	52,537	44,994	0,0665

Для моделі (1.1) з оцінками параметрів, що отримані для 40 рядків даних з табл.2.1 (за виключенням рядка 23), виявилось, що значення Y для всіх рядків даних не виходять за визначені межі інтервалу прогнозування. Границі інтервалу прогнозування на другій ітерації з табл.2.2 свідчать про те, що викиди у регресії відсутні. Тому на цьому побудову нелінійної регресійної моделі (1.1) завершено для 40 рядків даних з табл.2.1 (за виключенням рядка 23).

Остаточна нелінійна регресійна модель (1.1) має наступні оцінки параметрів: $\hat{b}_0 = -0.45001$, $\hat{b}_1 = 0.78887$, $\hat{b}_2 = 0.00176$, $\hat{b}_3 = 0.18782$, а оцінка середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$ випадкової величини ε дорівнює 0,0294.

Побудована нелінійна регресійна модель (1.1) була перевірена за множинним коефіцієнтом детермінації R^2 , середньою величиною відносної помилки MMRE і відсотком прогнозованих результатів, для яких величини відносної помилки MRE менші за 0,25, PRED(0,25). Значення R^2 , MMRE і PRED(0,25) для створеної моделі дорівнюють відповідно 0.9810, 0.0503 та 1.0. Це свідчить про задовільну якість отриманої моделі. Зазначимо, що показники R^2 , MMRE і PRED(0,25) обчислювалися за формулами (1.6), (1.4) та (1.5) відповідно. Одиничне значення PRED(0,25) вказує на те, що всі оцінки трудомісткості тестування $\hat{Y}$, які наведені в табл.2.2, для даних з табл.2.1 (за виключенням рядка 23) мають значення відносної похибки MRE менші за 0,25.

Для моделювання трудомісткості agile тестування невеликих веб проєктів за допомогою нелінійної регресійної моделі (1.1) треба здійснювати моделювання гаусівської випадкової величини ε з нульовим математичним сподіванням та оцінкою середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$, що дорівнює 0,0294. Для моделювання випадкової величини ε з розподілом Гауса у цій кваліфікаційній роботі використовується запропонований в [20, 21] метод на

основі бієктивного нормалізуючого перетворення Джонсона із сім'ї S_U . Вибір зазначеного методу був обумовлений тим, що він не має проблеми зменшення фактичної кількості значень псевдовипадкових чисел з рівномірним розподілом в межах періоду генератора, яку можна використовувати до їх повторення, при моделюванні випадкової величини з розподілом Гауса.

За зазначеним методом [21] значення гаусівської випадкової величини з математичним сподіванням нуль і дисперсією одиниця визначається за перетворенням Джонсона сім'ї S_U за формулою:

$$z = \eta \operatorname{Arsh}(x/\lambda), \quad (2.6)$$

де η та λ – це параметри перетворення Джонсона сім'ї S_U ; x – це випадкова величина, значення якої знаходяться за наступною функцією:

$$x = \operatorname{tg}(c_1 U + c_3 U^3), \quad U \in [-1, 1]. \quad (2.7)$$

В (2.7) U – це випадкова величина з рівномірним розподілом на інтервалі $[-1, +1]$; c_1 і c_3 – це певні константи, за допомогою яких можна регулювати розмах значень величини x , а через неї і випадкової величини z .

Для моделювання випадкової величини z з розподілом Гауса на інтервалі $[-1, +1]$ за формулами (2.6) та (2.7) ми обрали такі значення оцінок параметрів де η , λ , c_1 та c_3 : 0,48779, 0,22719, 0,57 та 1 відповідно, що наведені в [21]. Зазначимо, що за формулами (2.6) та (2.7) ми отримуємо значення випадкової величини z з математичним сподіванням нуль і дисперсією одиниця. А нам треба здійснювати моделювання гаусівської випадкової величини ε з нульовим

математичним сподіванням та оцінкою середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$, що дорівнює 0,0294. Для цього потрібно помножити значення z на оцінку середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$.

2.3 Висновки за розділом 2

У даному розділі виконано перевірку даних для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів на наявність викидів, побудовано нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів.

1) Перевірку даних для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів на наявність викидів виконано за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса. З'ясовано, що за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса, чотиривимірні дані містять один чотиривимірний викид. Застосування критерія на основі квадрата відстані Махаланобіса до даних без їх нормалізації показало, що дані містять два чотиривимірних викиди. Такий результат пояснюється тим, що розподіл даних не є гаусівським.

2) Удосконалено нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів на основі нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у регресії. Для цієї нелінійної регресійної моделі у порівнянні з існуючими регресійними нелінійними моделями визначено оцінку середньо

квадратичного відхилення випадкової величини, яка описує похибку, що дає змогу здійснювати моделювання трудомісткості agile тестування невеликих веб проєктів.

3) Отримані результати свідчать про те, що для побудови нелінійних регресійних моделей для моделювання трудомісткості agile тестування невеликих веб проєктів потрібно використовувати нормалізуючі перетворення з урахуванням викидів у даних.

РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ AGILE ТЕСТУВАННЯ НЕВЕЛИКИХ ВЕБ ПРОЄКТІВ

3.1 Постановка задачі на розробку проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів

Виходячи з аналізу існуючих моделей для моделювання трудомісткості agile тестування невеликих веб проєктів, сформулюємо постановку задачі. Потрібно створити програму для моделювання трудомісткості agile тестування невеликих веб проєктів, яка би задовольняла таким вимогам.

Вимоги до програми.

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

1) Вводити з текстового файлу емпіричні дані для моделювання трудомісткості agile тестування невеликих веб проєктів: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

2) Вводити з інтерфейсу програми довірчу ймовірність, три метрики за якими потрібно змодельовати трудомісткість agile тестування невеликого веб проєкту (у годинах): кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

3) Виконувати нормалізацію емпіричних даних за нормалізуючим перетворенням у формі десятичного логарифму.

4) Знаходити значення коефіцієнтів лінійного рівняння регресії для нормалізованих емпіричних даних.

5) Здійснювати побудову нелінійної регресійної моделі.

6) Обчислювати значення коефіцієнту детермінації R^2 .

7) Отримувати трудомісткість, нелінійну регресію, її довірчий інтервал та інтервал прогнозування.

8) Виводити на екран трудомісткість, нелінійну регресію, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах), значення коефіцієнту детермінації R^2 .

9) Виводити у текстовий файл результати моделювання трудомісткості agile тестування невеликих веб проєктів у годинах.

Вимоги до складу й параметрів технічних засобів.

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.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з програмою необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0., компілятор C (Visual Studio 2010 або Visual Express 2010) для компіляції зовнішніх модулів C або C++ та використання моделі в Xcos.

На підставі цієї постановки задачі розроблено технічне завдання, яке наведено у додатку А.

3.2 Розробка ескізного проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів

В ескізному проєкті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для моделювання трудомісткості agile тестування невеликих веб проєктів та стани цієї програми при виконанні.

Даний проєкт передбачає наявність лише одного актора – користувача програми, який буде повноцінно використовувати всі функції програми. Виявлені варіанти використання знаходяться в таблиці 3.1.

Таблиця 3.1 Виявлені варіанти використання

Основні актори	Найменування	Формулювання
1	2	3
Користувач	Ввести дані метрик веб проєктів	Вводить дані про фактичну трудомісткість agile тестування (у годинах), кількість тестових випадків, кількість сторінок програмно-технічної документації та кількість дефектів

Продовження табл. 3.1.

1	2	3
Користувач	Ввести три метрики для моделювання трудомісткості тестування	Вводить три метрики, за якими потрібно змодельовати трудомісткість agile тестування невеликого веб проєкту у годинах: кількість тестових випадків, кількість сторінок програмно-технічної документації та кількість дефектів
Користувач	Ввести довірчу ймовірність	Вводить довірчу ймовірність, для якої потрібно оцінити довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту у годинах
Користувач	Нормалізувати дані	Виконує нормалізацію внесених даних
Користувач	Побудувати нелінійну регресійну модель	Будує нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликого веб проєкту у годинах
Користувач	Отримати трудомісткість, її довірчий інтервал та інтервал прогнозування	Отримує із заданою ймовірністю довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості, випадкове значення трудомісткості та її середнє

Діаграма варіантів використання програми для моделювання трудомісткості agile тестування невеликих веб проєктів зображена на рисунку 3.1.

Сценарії варіантів використання зображені у вигляді діаграми діяльності на рисунку 3.2.

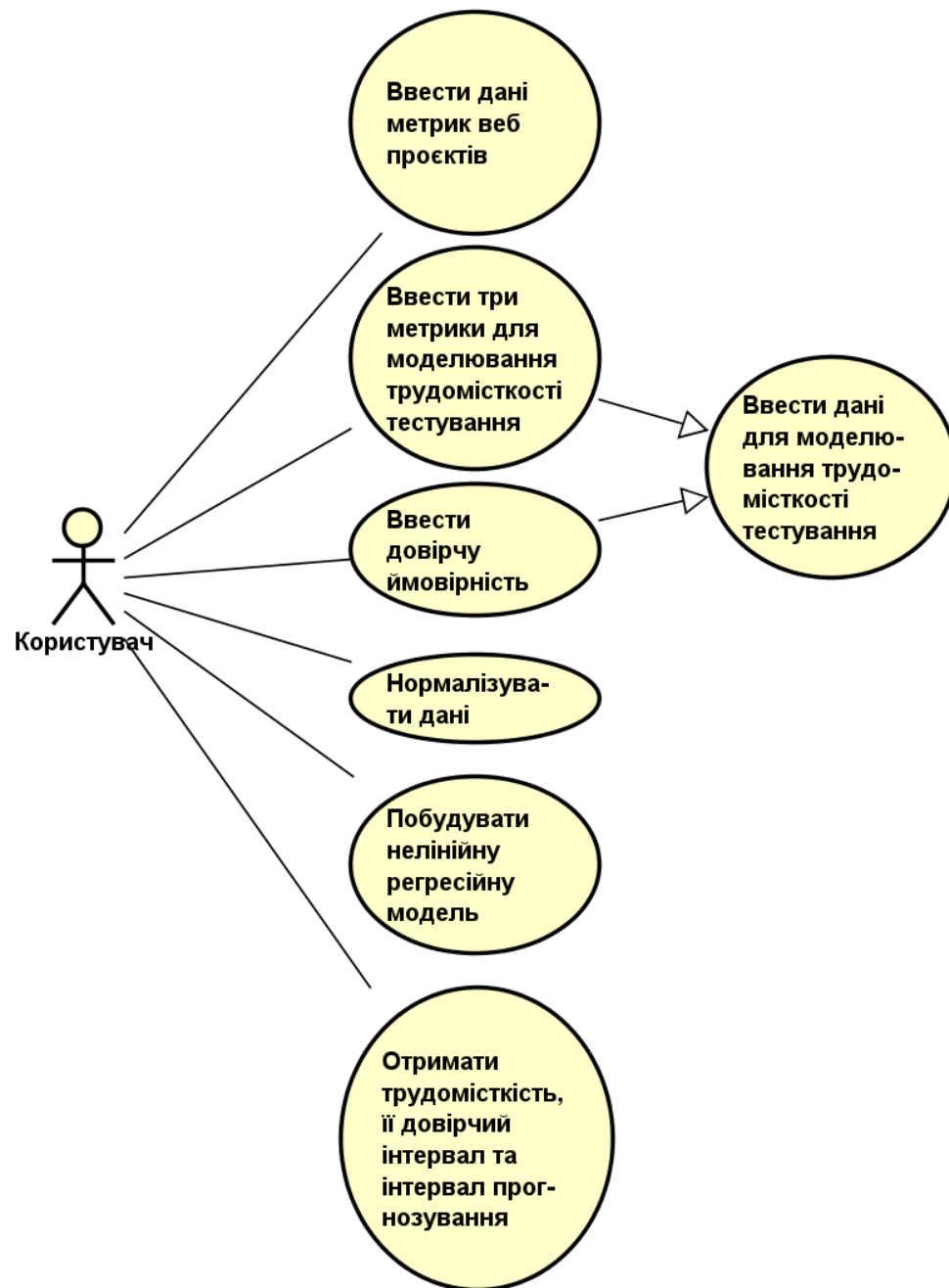


Рисунок 3.1 – Діаграма варіантів використання програми для моделювання трудомісткості agile тестування невеликих веб проєктів

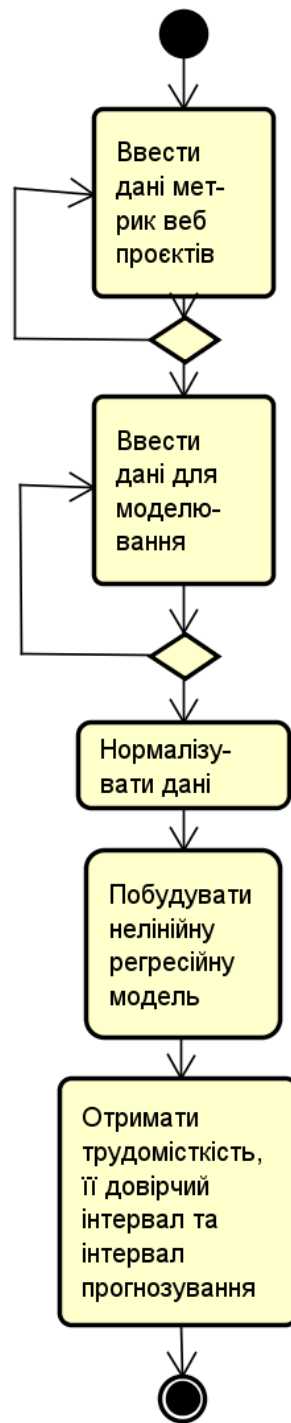


Рисунок 3.2 – Діаграма діяльності ПЗ для моделювання трудомісткості agile тестування невеликих веб проектів

Зовнішня специфікація програми.

Назва програми: WebTestEffortSim.

Функції, які виконує програма:

1) Вводити з текстового файлу емпіричні дані для моделювання трудомісткості agile тестування невеликих веб проєктів: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

2) Вводити з інтерфейсу програми довірчу ймовірність, три метрики, за якими потрібно змодельовати трудомісткість agile тестування невеликого веб проєкту (у годинах): кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

3) Виконувати нормалізацію емпіричних даних за нормалізуючим перетворенням у вигляді десяткового логарифму.

4) Знаходити значення коефіцієнтів лінійної регресійної моделі за нормалізованими емпіричними даними.

5) Здійснювати побудову нелінійної регресійної моделі.

6) Обчислювати значення коефіцієнту детермінації R^2 .

7) Отримувати трудомісткість, нелінійну регресію, її довірчий інтервал та інтервал прогнозування.

8) Виводити на екран трудомісткість, нелінійну регресію, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах), значення коефіцієнту детермінації R^2 .

9) Виводити у текстовий файл результати моделювання трудомісткості agile тестування невеликих веб проєктів у годинах.

Вхідні дані:

- емпіричні дані для моделювання трудомісткості agile тестування невеликих веб проєктів: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 ;

- значення довірчої ймовірності, трьох метрик, за якими потрібно змоделювати трудомісткість agile тестування невеликого веб проєкту (у годинах): кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

Вихідні дані:

- значення трудомісткості, нелінійної регресії, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах);

- значення коефіцієнту детермінації.

Зовнішні ефекти: поява графічного вікна для вводу даних, управління виводом даних, графічного вікна для виводу даних, значення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах), вихід з програми (закриття всіх графічних вікон).

Для проектування інтерфейсу програми для моделювання трудомісткості agile тестування невеликих веб проєктів (у годинах), представимо стани програми при виконанні у вигляді відповідної діаграми станів. Діаграма станів програми для моделювання трудомісткості agile тестування невеликих веб проєктів (у годинах), наведена на рисунку 3.3.

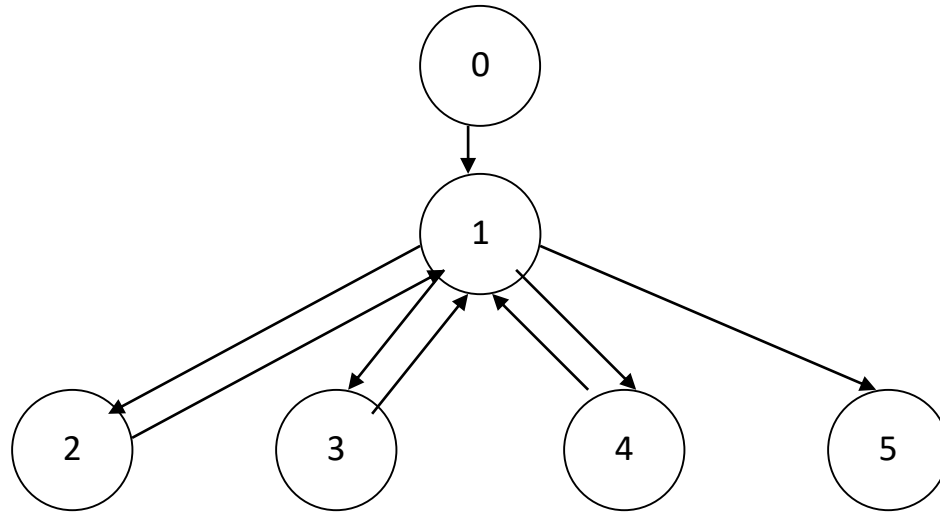


Рисунок 3.3 – Діаграма станів програми для моделювання трудомісткості agile тестування невеликих веб проєктів

На діаграмі станів програми введені такі позначення станів: 0 – початковий стан, 1 – графічне вікно головного модуля, 2 – графічне вікно візуалізації введених емпіричних даних, 3 – графічне вікно із результатами знаходження нелінійної регресії, 4 – графічне вікно з результатами трудомісткості, нелінійної регресії, довірчим інтервалом та інтервалом прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах), 5 – кінцевий стан.

Далі виконуємо розробку технічного проєкту програми для оцінювання трудомісткості agile тестування невеликого веб проєкту (у годинах)

3.3 Розробка технічного проєкту програми для моделювання трудомісткості agile тестування невеликих веб проєктів

В технічному проєкті було спроектовано програму для оцінювання трудомісткості agile тестування невеликого веб проєкту (у годинах), на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

Програма складається з таких модулів: головного модуля (WebTestEffortSim), модуля для введення емпіричних даних з файлу та їх візуалізації (DataInput), модуля для визначення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (WebTestEffortIntEst), модуля для закриття графічних вікон (W_Close).

Специфікація головного модуля.

Назва модуля: WebTestEffortSim.

Функції, які виконує програма:

- 1) Відкриття графічного вікна головного модуля.
- 2) Введення імені файлу з емпіричними даними.
- 3) Виклик функції для введення емпіричних даних з файлу та їх візуалізації у графічному вікні.
- 4) Вибір наявності фактичної трудомісткості веб проєкту та введення її значення у годинах.
- 5) Виклик функції для визначення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах).
- 6) Виклик функції для закриття графічних вікон.

Вхідні дані:

- емпіричні дані для моделювання трудомісткості agile тестування невеликих веб проєктів: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 ;

- значення довірчої ймовірності, трьох метрик, за якими потрібно змоделювати трудомісткість agile тестування невеликого веб проєкту (у годинах): кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

Вихідні дані:

- значення трудомісткості, нелінійної регресії, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах);

- значення коефіцієнту детермінації.

Зовнішні ефекти: поява графічного вікна для вводу даних, управління виводом даних, графічного вікна для виводу даних, значення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах), вихід з програми (закриття всіх графічних вікон).

Специфікація модуля для введення емпіричних даних з файлу та їх візуалізації.

Назва модуля: DataInput.

Функції, які виконує модуль:

- 1) Введення емпіричних даних (значень випадкових величини) з файлу.
- 2) Формування одномірних та двомірних масивів значень випадкових величини (емпіричних даних).

3) Відкриття графічного вікна для візуалізації введених емпіричних даних.

4) Візуалізацію введених емпіричних даних.

Вхідні дані:

- значення випадкових величини (емпіричні дані).

Вихідні дані:

- одномірні та двомірні масиви значення випадкових величини (емпіричні дані);

- кількість значень випадкових величин (емпіричних даних);

- ідентифікатор графічного вікна.

Зовнішні ефекти: Поява графічних вікон для візуалізації введених емпіричних даних.

Специфікація модуля для визначення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах).

Назва модуля: WebTestEffortIntEst.

Функції, які виконує модуль:

1) Визначає трудомісткість, нелінійну регресію, довірчий інтервал та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах).

2) Обчислює значення коефіцієнту детермінації для нелінійної регресії.

Вхідні дані:

- двомірні масиви значень випадкових величини (емпіричних даних);

- кількість значень випадкових величин (емпіричних даних).

Вихідні дані:

- значення трудомісткості, нелінійної регресії, нижньої, верхньої границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах);

- значення коефіцієнту детермінації для нелінійної регресії.

Зовнішні ефекти: з'являються результати трудомісткості, нелінійної регресії, нижньої та верхньої границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах) та виводиться значення коефіцієнту детермінації для нелінійної регресії.

Назва модуля: W_Close.

Функції, які виконує модуль:

Закриття графічних вікон.

Вхідні дані:

- ідентифікатори графічних вікон.

Вихідні дані: відсутні.

Зовнішні ефекти: Закриття всіх графічних вікон.

В подальшому були розроблені алгоритми зазначених модулів.

3.4 Розробка робочого проєкту програми для моделювання трудомісткості agile тестування невеликих веб проєктів

В робочому проєкті було розроблено програму для моделювання трудомісткості agile тестування невеликих веб проєктів, здійснено тестування програми, розроблено програмну документацію, здійснено випробування програми.

Спочатку sci-мовою системи моделювання Scilab було виконано кодування алгоритмів, що були розроблені на стадії технічного проекту. Текст програми наведений в додатку Б. Приклад кодування модуля WebTestEffortIntEst, що здійснює моделювання трудомісткості agile тестування невеликих веб проєктів, sci-мовою в системі моделювання Scilab наведено нижче.

```
//Функція визначення трудомісткості agile тестування невеликого веб
//проекту
function WebTestEffortIntEst(x, yx, n)
//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
x1i=eval(get(edit_text4,'string'));
x2i=eval(get(edit_text5,'string'));
x3i=eval(get(edit_text6,'string'));
Pconf=eval(get(edit_text7,'string'));

x1A=0; x2A=0; x3A=0; yA=0;
for i=1:n
    x1A=x1A+x(i,1); x2A=x2A+x(i,2); x3A=x3A+x(i,3); yA=yA+yx(i,1);
end;
x1A=x1A/n; x2A=x2A/n; x3A=x3A/n; yA=yA/n;

Sx1x1=0; Sx2x2=0; Sx3x3=0;
Sx1x2=0; Sx1x3=0; Sx2x3=0; SyA=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    Sx1x2=Sx1x2+(x(i,1)-x1A)*(x(i,2)-x2A);
    Sx1x3=Sx1x3+(x(i,1)-x1A)*(x(i,3)-x3A);
    Sx2x3=Sx2x3+(x(i,2)-x2A)*(x(i,3)-x3A);
    Sx2x2=Sx2x2+(x(i,2)-x2A)*(x(i,2)-x2A);
    Sx3x3=Sx3x3+(x(i,3)-x3A)*(x(i,3)-x3A);
    SyA=SyA+(yx(i,1)-yA)*(yx(i,1)-yA);
end;
Sxx(1,1)=Sx1x1; Sxx(1,2)=Sx1x2; Sxx(1,3)=Sx1x3;
Sxx(2,1)=Sx1x2; Sxx(2,2)=Sx2x2; Sxx(2,3)=Sx2x3;
Sxx(3,1)=Sx1x3; Sxx(3,2)=Sx2x3; Sxx(3,3)=Sx3x3;

Sxx_1=1/Sxx;

z1A=0; z2A=0; z3A=0;
for i=1:n
    z1A=z1A+log10(x(i,1)); z2A=z2A+log10(x(i,2)); z3A=z3A+log10(x(i,3));
end;
z1A=z1A/n; z2A=z2A/n; z3A=z3A/n;
```

```

Sz1z1=0; Sz2z2=0; Sz3z3=0;
Sz1z2=0; Sz1z3=0; Sz2z3=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
    Sz1z2=Sz1z2+(log10(x(i,1))-z1A)*(log10(x(i,2))-z2A);
    Sz1z3=Sz1z3+(log10(x(i,1))-z1A)*(log10(x(i,3))-z3A);
    Sz2z3=Sz2z3+(log10(x(i,2))-z2A)*(log10(x(i,3))-z3A);
    Sz2z2=Sz2z2+(log10(x(i,2))-z2A)*(log10(x(i,2))-z2A);
    Sz3z3=Sz3z3+(log10(x(i,3))-z3A)*(log10(x(i,3))-z3A);
end;
Szz(1,1)=Sz1z1; Szz(1,2)=Sz1z2; Szz(1,3)=Sz1z3;
Szz(2,1)=Sz1z2; Szz(2,2)=Sz2z2; Szz(2,3)=Sz2z3;
Szz(3,1)=Sz1z3; Szz(3,2)=Sz2z3; Szz(3,3)=Sz3z3;

Szz_1=1/Szz;

zi(1)=log10(x1i)-z1A; zi(2)=log10(x2i)-z2A; zi(3)=log10(x3i)-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

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

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

b0=-0.45001;
b1=0.7888688;
b2=0.001760;
b3=0.1878210;

Szyzyn4=0; Syy=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))+b2*log10(x(i,2))+b3*log10(x(i,3))-
        log10(yx(i,1)))^2;

    Syy=Syy+(yx(i,1)-10^b0*x(i,1)^b1*x(i,2)^b2*x(i,3)^b3)^2;
end;
Szyzyn4=Szyzyn4/(n-4)
Delta_z=T*(Szyzyn4*(1/n+result))^0.5;
Delta_z_pr=T*(Szyzyn4*(1+1/n+result))^0.5;

r_2=1-Syy/SyA;

zy_result=b0+b1*log10(x1i)+b2*log10(x2i)+b3*log10(x3i);

```

```

zyLBC=zy_result-Delta_z; zyUBC=zy_result+Delta_z;
zyLBPr=zy_result-Delta_z_pr; zyUBPr=zy_result+Delta_z_pr;

y_result=10^zy_result;
yLBC=10^zyLBC; yUBC=10^zyUBC;
yLBPr=10^zyLBPr; yUBPr=10^zyUBPr;

    r = grand(1, 1, "unf", -1, 1);
    xtan=tan(0.57*r);
    zd=0.532275*asinh(3.874868*xtan);
    zd=0.0294*zd;

zy_random=zy_result+zd;
y_random=10^zy_random;

file_PC5=mopen('D:\Data_F\WebTestEffort_40_data5.txt','w+');
//for i=1:3
    fprintf(file_PC5, "%5.6f ", yUBC);
//end
fclose(file_PC5);

file_PC4=mopen('D:\Data_F\WebTestEffort_40_data4.txt','w+');
for i = 1:3
    for j = 1:3 fprintf(file_PC4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_PC4, "\n");
end;
fclose(file_PC4);

set(textresult,'string', sprintf("=%1.4f",y_result));
set(textresult2,'string', sprintf("=%1.4f",yLBC));
set(textresult3,'string', sprintf("=%1.4f",yUBC));
set(textresult4,'string',..
    sprintf("Коефіцієнт детермінації R^2=%1.4f",r_2));
set(textresult5,'string', sprintf("=%1.4f",yLBPr));
set(textresult6,'string', sprintf("=%1.4f",yUBPr));
set(textresult7,'string', sprintf("=%1.4f",y_random));

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

else
    yfact=eval(get(edit_text12,'string'));
    MRE=abs((yfact-y_result)/yfact);
    textresult5=uicontrol(f,'style','text','string','', 'position',...
        [500,120,160,20]);
    set(textresult5,'string',..
        sprintf("Відносна похибка MRE=%1.4f",MRE));
end;

endfunction

```

Далі в середовищі Scilab 6.0.0 було створено sce-файл розміром 14 КБ. Зазначимо, що в програмі використовувався динамічний спосіб створення інтерфейсних компонентів. Він полягає у тому, що при виконанні програми можуть створюватися і знищуватися ті чи інші елементи управління (командні кнопки, радіокнопки, мітки та таке інше), а їх властивостям присвоюються відповідні значення. В програмі були використані такі інтерфейсні компоненти: вікно редагування, командні кнопки та радіокнопки. Графічне вікно програми із зазначеними інтерфейсними компонентами наведено на рис.3.4.

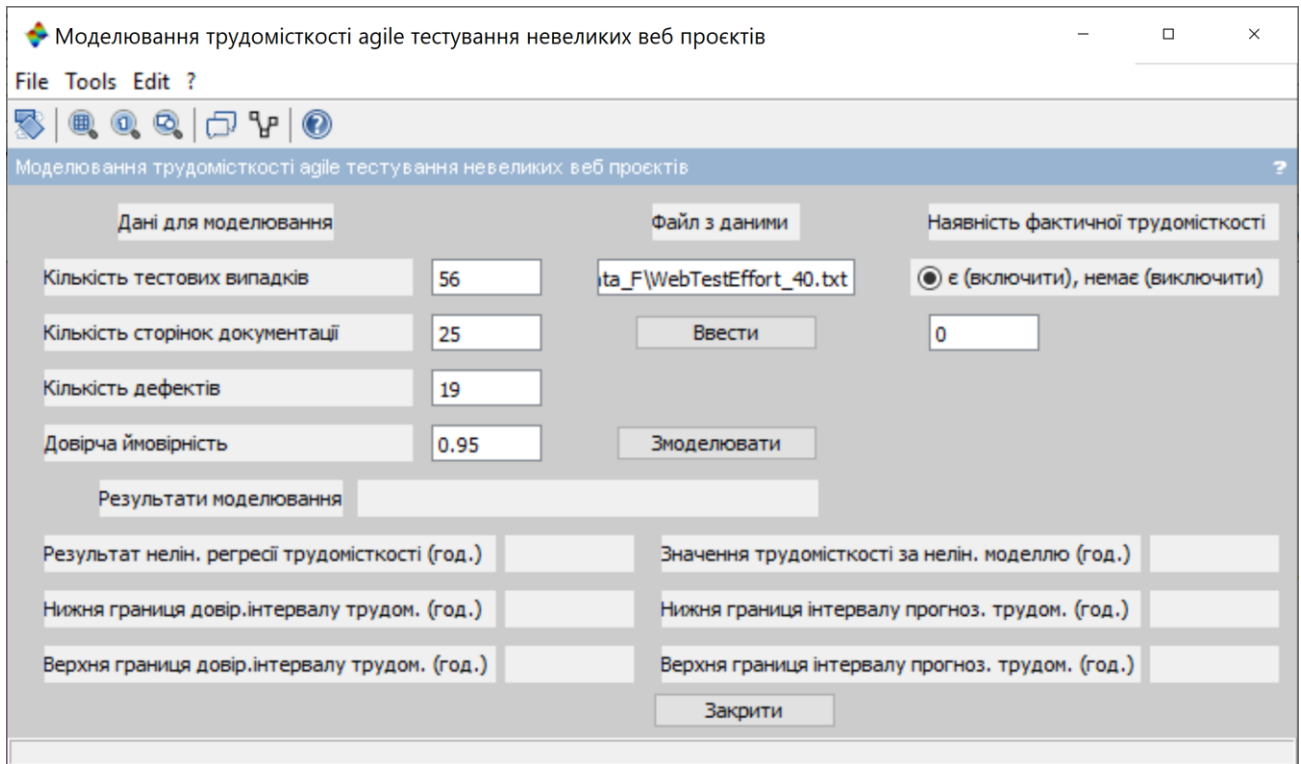


Рисунок 3.4. – Графічне вікно програми із інтерфейсними компонентами

Було проведено тестування окремих модулів. У якості прикладу наведемо результати тестування для модуля для моделювання трудомісткості agile тестування невеликих веб проєктів для емпіричних даних – трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості тестових випадків, кількості сторінок програмно-технічної документації та кількості дефектів. Було побудовано нелінійну регресійну модель на основі нормалізуючого перетворення у формі десяткового логарифму. Результати візуалізації емпіричних даних з трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості тестових випадків наведені на рис.3.5.

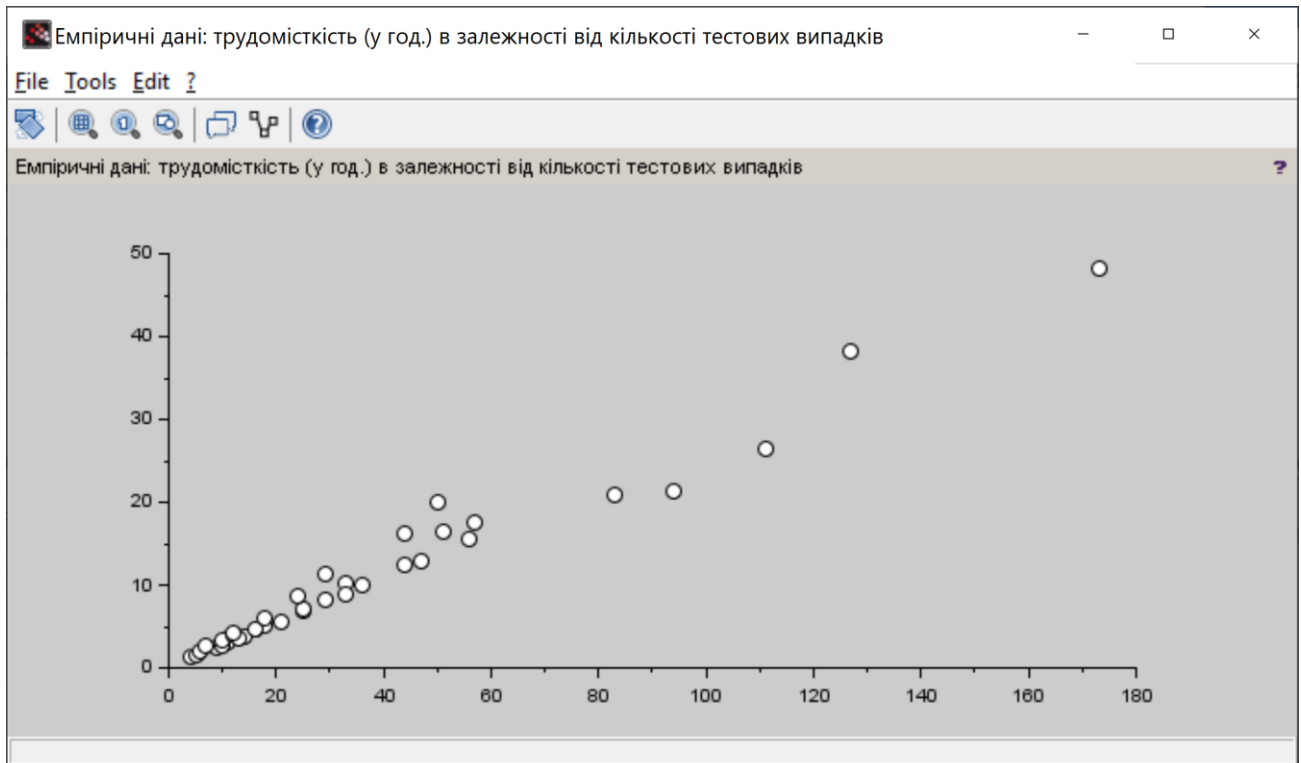


Рисунок 3.5. – Емпіричні дані з трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості тестових випадків

Результати візуалізації емпіричних даних з трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості сторінок програмно-технічної документації наведені на рис.3.6.



Рисунок 3.6. – Емпіричні дані з трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості сторінок програмно-технічної документації

Результати візуалізації емпіричних даних з трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості дефектів наведені на рис.3.7.

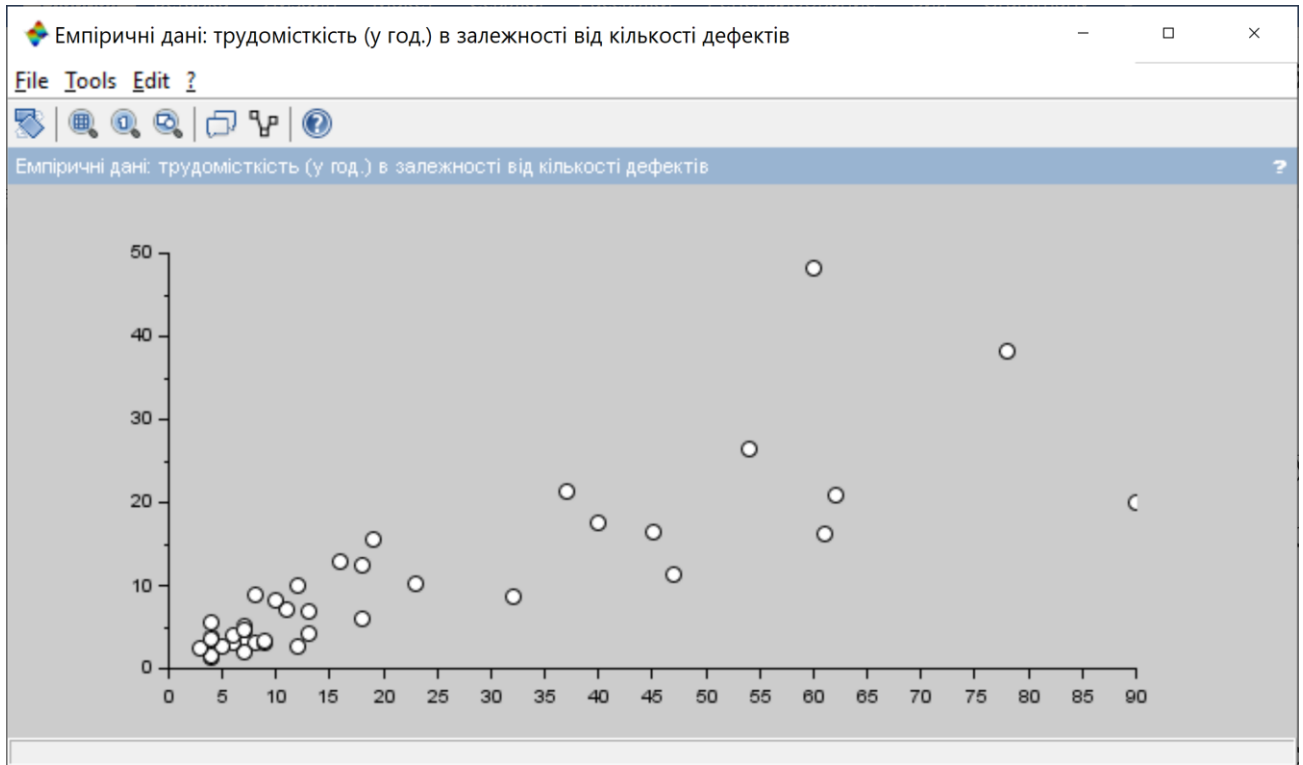


Рисунок 3.7. – Емпіричні дані з трудомісткості agile тестування (у годинах) 40 невеликих веб проєктів в залежності від кількості дефектів

Результати моделювання трудомісткості agile тестування (у годинах), значень нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії трудомісткості agile тестування (у годинах) веб проєкту та результати обчислення коефіцієнту детермінації R^2 наведені на рис.3.8. Також у разі відомого фактичної трудомісткості agile тестування веб проєкту може бути обчислено значення відносної похибки (MRE) результату визначення

значень нелінійної регресії в порівнянні з фактичною трудомісткістю agile тестування веб проєкту для зазначених метрик.

Моделювання трудомісткості agile тестування невеликих веб проєктів

File Tools Edit ?

Моделювання трудомісткості agile тестування невеликих веб проєктів

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості
Кількість тестових випадків	56	ta_F\WebTestEffort_40.txt	☐ є (включити), немає (виключити)
Кількість сторінок документації	25	Ввести	0
Кількість дефектів	19		
Довірча ймовірність	0.95	Змоделювати	

Результати моделювання Коефіцієнт детермінації $R^2=0.9810$

Результат нелін. регресії трудомісткості (год.)	=14.8497	Значення трудомісткості за нелін. моделлю (год.)	=14.0027
Нижня границя довір.інтервалу трудом. (год.)	=14.3026	Нижня границя інтервалу прогноз. трудом. (год.)	=12.8100
Верхня границя довір.інтервалу трудом. (год.)	=15.4177	Верхня границя інтервалу прогноз. трудом. (год.)	=17.2142

Закрити

Рисунок 3.8. – Результати визначення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах)

На рис.3.9 наведено значення відносної похибки (MRE) результату визначення значень нелінійної регресії в порівнянні з фактичною трудомісткістю agile тестування веб проєкту для зазначених метрик.

Моделювання трудомісткості agile тестування невеликих веб проєктів

File Tools Edit ?

Моделювання трудомісткості agile тестування невеликих веб проєктів

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості
Кількість тестових випадків	56	ta_F\WebTestEffort_40.txt	☒ є (включити), немає (виключити)
Кількість сторінок документації	25	Ввести	15.56
Кількість дефектів	19		
Довірча ймовірність	0.95	Змоделювати	
Результати моделювання		Коефіцієнт детермінації $R^2=0.9810$	Відносна похибка MRE=0.0456
Результат нелін. регресії трудомісткості (год.)	=14.8497	Значення трудомісткості за нелін. моделлю (год.)	=14.6689
Нижня границя довір.інтервалу трудом. (год.)	=14.3026	Нижня границя інтервалу прогноз. трудом. (год.)	=12.8100
Верхня границя довір.інтервалу трудом. (год.)	=15.4177	Верхня границя інтервалу прогноз. трудом. (год.)	=17.2142
Закрити			

Рисунок 3.9. – Результати визначення відносної похибки оцінки трудомісткості в порівнянні з фактичною трудомісткістю agile тестування веб проєкту для зазначених метрик

Зверніть увагу, що результати моделювання трудомісткості agile тестування веб проєкту для зазначених метрик за нелінійною регресійною моделлю (1.1) на рис. 3.8 та 3.9 наведені у полі «Значення трудомісткості за нелін. моделлю (год.)». Вони змінюються як значення випадкової величини Y у (1.1).

Отримані результати практично співпадають з результатами, що наведені у табл.2.2 та оцінками інших авторів [7]. Це свідчить про працездатність розробленої програми.

3.5 Висновки за розділом 3

У даному розділі виконано постановку задачі на розробку проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів, розроблено ескізний, технічний та робочий проекту програми для моделювання трудомісткості agile тестування невеликих веб проєктів.

1) В ескізному проєкті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для моделювання трудомісткості agile тестування невеликих веб проєктів та стани цієї програми при виконанні.

2) В технічному проєкті було спроектовано програму для моделювання трудомісткості agile тестування невеликих веб проєктів на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

3) В робочому проєкті було розроблено програму для моделювання трудомісткості agile тестування невеликих веб проєктів, здійснено тестування програми, розроблено програмну документацію, здійснено випробування програми. Результати випробування програми практично співпадають з результатами, що отримані іншими авторами. Це свідчить про працездатність розробленої програми.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМИ

Найбільш важливим моментом при проектуванні комп'ютерної програми для розробника, з економічної точки зору, є процес формування її вартості. Вочевидь, що комп'ютерна програма являє собою дуже специфічний товар з безліччю особливостей.

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

4.1 Розрахунок витрат на створення й експлуатацію програми

Витрати на розробку програми складаються з витрат на зарплату розробника, на амортизацію комп'ютера, на якому виконується розробка, на експлуатацію цього комп'ютера, на засоби розробки та витрат на матеріали і комплектуючі.

Розробка програми виконується програмістом, місячний оклад якого складає 30000 грн. Додаткова заробітна плата складає 20% від основної. Виходячи з цього, основна і додаткова заробітна плата програміста

36000 грн/мес, а вартість сучасного ноутбука складає 26076 грн. (приведена вартість ноутбука Lenovo IdeaPad 3 15ALC6 (82KU00NERA) Arctic Grey: Екран 15.6" TN (1920x1080) Full HD, матовий / AMD Ryzen 3 5300U (2.6 - 3.8 ГГц) / RAM 8 ГБ / SSD 128 ГБ / AMD Radeon Graphics / без ОД / Wi-Fi / Bluetooth / веб-камера / Windows 11 Професійна на 1ПК (ESD - електронна ліцензія в конверті, всі мови) (FQC-10572)). При вартості кіловат-години електроенергії у 1,7166 грн. (II клас напруги до 27,5 кВт, разом з ПДВ), розраховується вартість розробки ПЗ. Витрати на допоміжні матеріали приведені в табл. 4.1

Таблиця 4.1 – Витрати на допоміжні матеріали

Пункти витрат	Сума
Папір	180,00
Заправлення картриджа до принтера	180,00
CD-диск	12,00
Непередбачені витрати	300,00
Разом матеріали і комплектуючі	672,00

Вартість програми розраховуємо за формулою:

$$\text{Спр} = (\text{Ззп} + \text{Зсз} + \text{Ззг} + \text{Зе}) * \text{T} + \text{Зм}, \quad (4.1)$$

де Т – тривалість розробки, міс.; Ззп – основна і додаткова заробітна плата обслуговуючого персоналу, грн.; Зсз – відрахування на соціальні заходи або єдиний соціальний внесок (36,76% від основної і додаткової заробітної плати), грн.; Ззг – загальногосподарські витрати (10% від основної заробітної плати), грн.; Зе – витрати на електроенергію; Зм – витрати на основні і допоміжні матеріали. Зе при споживанні потужності 0,5 кВт, тривалості роботи за місяць, рівної $22 * 8 = 176$ годин, вартості кіловат-години електроенергії 1,7166 грн. складає

$$З_е = 176 * 1,7166 * 0,5 = 151,06 \text{ грн.}$$

Таблиця 4.2 – Витрати на розробку ПЗ

Найменування витрат	Одиниця	Кількість
Тривалість розробки	Мес	1,5
Основна і додаткова ЗП	Грн.	36000
Відрахування на соціальні заходи	Грн.	13233,6
Загальногосподарські витрати	Грн.	500
Витрати на матеріали	Грн.	672,00
Витрати на електроенергію	Грн.	151,06

Відповідно до формули (4.1) вартість програми:

$$С_{пр} = (36000 + 13233,6 + 500 + 151,06) * 1,5 + 672,00 = 75498,99 \text{ грн.}$$

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

$$А_{об} = 26076 * 0,6 = 15645,6 \text{ грн.}$$

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

$$В_м = 16800 * 0,05 = 1303,8 \text{ грн.}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$Ф_м = 264,5 * Т_з, \quad (4.2)$$

де $Т_з$ – це середнє місячне завантаження устаткування (близько 4 годин); 264,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПК складе:

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

Витрати на електроенергію Z_e при 1013 годинах роботи устаткування в рік складуть

$$Z_e = 1058 \cdot 0,5 \cdot 1,7166 = 908,08 \text{ грн.}$$

Експлуатаційні витрати для ПК за рік складуть:

$$Z_p = 15645,6 + 1303,8 + 908,08 = 17857,48 \text{ грн.}$$

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

$$Z_{se} = 75498,99 + 17857,48 = 93356,47 \text{ грн.}$$

4.2 Економічна ефективність розробки і впровадження програми

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

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

- високі витрати на складування паперових документів (сейфи, шафи, папки й ін.);
- підвищені витрати на канцтовари;

– витрати, пов'язані з роботою виявлення раніше допущених помилок (людський фактор).

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

- підвищення продуктивності праці;
- вивільнення робочого часу.

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

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

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

Зарплата 0,6 працівника в рік складає:

$$(30000 \cdot 12) \cdot 0,6 = 216000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$E_p = \Delta C_n - E_n \cdot k,$$

де ΔC_n – вивільнені кошти після впровадження програми (216000 грн.) мінус експлуатаційні витрати (17857,48 грн.); E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації – 0,6); k – одноразові витрати на впровадження програми (75498,99 грн.).

Річний економічний ефект буде

$$E_p = 198142,52 - 45299,39 = 152843,13 \text{ грн.}$$

Строк окупності програми розраховується по формулі:

$$T = \frac{k}{\Delta C_n} = \frac{75498,99}{198142,52} = 0,38 \text{ (року).}$$

Отже строк окупності програми складає приблизно 4,6 місяця.

5 ОХОРОНА ПРАЦІ

5.1 Аналіз існуючої проблеми охорони праці

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людей в процесі праці (ст. 1 Закону «Про охорону праці» із змінами [22]). Як правовий інститут охорона праці – це чималий комплекс правових норм. Можна визначити охорону праці також як систему забезпечення безпеки, збереження здоров'я і працездатності людини в процесі праці, засновану на сукупності законодавчих актів і відповідних їм соціально-економічних, технічних, гігієнічних і організаційних заходів.

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.

Власник або уповноважений ним орган повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизму, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників.

На власника або уповноважений ним орган покладається систематичне проведення інструктажу (навчання) працівників з питань охорони праці, протипожежної охорони.

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

5.2 Аналіз шкідливих та небезпечних факторів у офісному приміщенні з персональними комп'ютерами

Небезпечним називається виробничий фактор, вплив якого на працюючу людину в певних умовах приведе до травми або іншому раптовому різкому погіршенню здоров'я. Якщо ж виробничий фактор приведе до захворювання або зниження працездатності, то його вважають шкідливим. Залежно від рівня й тривалості впливу шкідливий виробничий фактор може стати небезпечним.

На офісного працівника при роботі впливають дві основні групи шкідливих факторів:

- 1) Фізичні, до яких відносять фактори, що породжуються безпосереднім впливом оточуючого середовища (електромагнітне випромінювання, освітленість робочого місця, температура повітря, шум).

2) Психофізіологічні, до яких відносять фізичне та нервово-психічне перевантаження організму (зорове напруження, статичні навантаження, нервово-психічне навантаження).

Освітлення – отримання, розподіл та використання світлової енергії для забезпечення нормальних умов праці. Світло, крім забезпечення фізичної можливості розрізняти предмети праці, також впливає на психічний стан людини та перебіг фізіологічних процесів в організмі. Раціонально влаштоване освітлення виробничих приміщень позитивно впливає на працівників, підвищує ефективність та безпеку праці, знижує втому та травматизм, забезпечує високу працездатність.

За вимогами санітарних норм освітлення повинно бути достатньо яскравим, рівномірним, не засліплювати очі та не створювати відблисків на робочій поверхні. За спектральним складом освітлення має наближатися до сонячного світла. Гігієнічними нормами вимагається максимально використовувати природне освітлення, оскільки денне світло краще сприймається органами зору. Штучне освітлення у виробничих та адміністративно-громадських приміщеннях має здійснюватись системою загального рівномірного освітлення, допускають застосування системи комбінованого освітлення. Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500лк. Як джерела світла для штучного освітлення мають застосовувати переважно люмінесцентні лампи типу ЛБ.

Одним з найнесприятливіших факторів, що зменшує працездатність та уважність робітників, створює передумови до травматизму та захворювань, є шум. Інтенсивний шум та вібрації негативно впливають не лише на слух, а і на

нервову, серцево-судинну, та більшість інших систем організму, викликаючи зниження слуху, гіпертонію, психічні розлади тощо.

Гігієнічними нормативами (СН 3222-85, ГОСТ 12.1.003-85, ГР 2411-81) встановлені допустимі рівні звуку різних частот для робітників певних професій. У таблиці 5.1 наведені нормативи для людей, що працюють з ПК.

Таблиця 5.1 – Допустимі рівні шуму

Вид трудової діяльності	Рівні звукового тиску в дБ октавних смугах із середньо геометричними частотами, ГЦ									
	31,5	63	125	250	500	1000	2000	4000	8000	Рівні звуку, еквівалентні рівні звуку, дБа/дБАекв
Програмісти комп'ютерів	86	71	61	54	49	45	42	40	38	50
Оператори в залах обробки інформації на комп'ютері та оператори комп'ютерного набору	96	83	74	68	63	60	57	55	54	65
В приміщеннях для розташування шумних агрегатів ЕОМ	103	91	83	77	73	70	68	66	64	75

Виходячи з даних норм, при проведенні комплексу заходів зі зменшення шуму усувається не будь-яка віброакустична активність обладнання, а тільки та, яка перевищує встановлений граничний рівень та шкідливо впливає на організм людини.

Усунення промислового шуму в першу чергу відбувається шляхом вдосконалення технологічних процесів та обладнання. При цьому в першу чергу усуваються найбільш сильні джерела шуму. Також необхідно знешумлювати усе обладнання, яке цього потребує, оскільки знешумлення окремих одиниць при наявності великої кількості обладнання, що продовжує випромінювати шум, недоцільно.

Самопочуття та працездатність людини також сильно залежать від теплового стану організму, на який впливають такі фізичні фактори виробничого середовища, як температура повітря, вологість, швидкість руху повітря, атмосферний тиск, доля кисню у повітрі тощо. Наприклад, якщо кількість кисню в повітрі зменшиться до 12 % то утруднюється дихання. В таких умовах людина напружує дихальний апарат, дихає частіше; такий стан людина витримує до 0,5 години. Перегрівання організму відбувається за умов надлишкового конвективного випромінювання тепла нагрітих поверхонь. При перегріванні вступають в активну реакцію пристосувальні функції організму. При цьому активізується робота серцево-судинної та дихальної систем, відбувається інтенсивне потовиділення, котре сягає 5 л за зміну. З потом втрачається велика кількість мінеральних солей та вітамінів. Вологість повітря суттєво впливає на терморегуляцію людського організму. Підвищення відносної вологості повітря у виробничому приміщенні ускладнює терморегуляцію, зменшення тепловиділення організмом. Фізіологічно оптимальною є відносна вологість в межах 40..60 %.

Сукупність описаних показників стану навколишнього середовища називають мікрокліматом. Для приміщень з комп'ютерами встановлені норми мікроклімату приведені у таблиці 5.2.

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека ураження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватися правил електробезпеки.

Таблиця 5.2 – Норми мікроклімату для приміщень з комп'ютерами

Пора року	Категорія робіт	Температура повітря, С, не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	Легка – 1а	22-24	40-60	0,1
	Легка – 1б	21-23	40-60	0,1
Тепла	Легка – 1а	23-25	40-60	0,1
	Легка – 1б	22-24	40-60	0,2

Небезпека електричного струму на відміну від інших небезпек збільшується тим, що людина не в змозі без спеціальних приладів виявити напругу дистанційно, а також швидкоплинністю поразки – небезпека виявляється, коли людина вже уражена. Ураження струмом приводить до різних порушень в організмі, викликаючи як місцеву поразку тканин і органів, так і загальне ураження організму.

Людина відчуває дратівну дію змінного струму промислової частоти силою 0,6 -1,5 мА і постійного струму 5-7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Для перемінного струму промислової частоти сила невідпускаючого струму знаходиться в межах 6-20 мА. Постійний струм не викликає невідпускаючого ефекту, але приводить до сильних болючих відчуттів, сила такого струму 15-80 мА і більше.

5.3 Розрахунок системи пожежогасіння офісного приміщення

Група приміщення: 1

Тип зрошувача: вода

Розміри приміщення: довжина $a = 7$ м, ширина $b = 6$ м.

1. Знайдемо групу приміщення (табл. 653) згідно з пожежним навантаженням, які забезпечуються автоматичними установками пожежогасіння (ДВН В.2.5-13-98).

За таблицею 5.3 приміщення офісу належить до 1 групи.

2. Параметри для розрахунку спринклерної установки залежно від групи приміщення (1 група) є наступними:

- Інтенсивність зрошування водою $L = 0,08$ л/(см²).
- Площа, яка захищається одним зрошувачем $S_{зр} = 12$ м².
- Тривалість роботи установки водного пожежогасіння $T = 30$ хвилин
- Відстань між зрошувачами $D = 4$ м.

3. Знаходимо площу приміщення:

$$S_{\text{прим}} = a * b = 7 * 6 = 42 \text{ м}^2.$$

4. Знаходимо необхідну кількість зрошувачів:

$$N = S_{\text{прим}}/S_{\text{зр}} = 42 / 12 = 3,5.$$

5. Розміщуємо зрошувачі на плані приміщення.

6. Знаходимо необхідну інтенсивність води в трубопроводі:

$$L_{\text{тр}} = L * S_{\text{прим}} = 0,08 * 42 = 3,36 \text{ л/с}.$$

7. Знаходимо інтенсивність води крізь один спринклер:

$$L_{\text{др}} = L_{\text{тр}} / N = 3,36 / 3,5 = 0,97 \text{ л/с}.$$

8. Знаходимо необхідний об'єм води:

$$Q_{\text{н}} = L_{\text{тр}} * T = 3,36 * 1800 \text{ с} = 6048 \text{ л}.$$

Таблиця 5.3 – Групи приміщень

Група	Приміщення	Пожежне навантаження
1	Приміщення книгосховищ, з комп'ютерами, магазинів, будівель управлінь, готелів, лікарень	до 200 Мдж/м ²
2	Приміщення з використанням рідин, які легко спалахують (ЛСР) і горять (ГР); деревообробні, швейні та інші приміщення; підприємства з обслуговування автомобілів	200...2000 Мдж/м ²
3	Приміщення гумотехнічного виробництва	200...2000 Мдж/м ²
4	Приміщення для виробництва, обробки, переробки різних матеріалів з використанням ЛСР і ГР	>2000 Мдж/м ²
5	Склади негорючих матеріалів в упаковці, що горить	>2000 Мдж/м ²
6	Склади твердих матеріалів, що горять	>2000 Мдж/м ²
7	Склади лаків, фарб, ЛСР, ГР, пластмас та ін.	>2000 Мдж/м ²

5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів

Сьогодні фахівці в області ергономіки вже зрозуміли, що не можна знайти ідеальне положення, у якому можна перебувати і працювати протягом усього робочого дня. Для більшості людей комфортабельним робочим місцем повинне бути таке, котре можна пристосувати не менш чим для двох позицій, при цьому положення крісла, монітора повинні щораз відповідати виконуваній роботі і звичкам. Багато хто вважають, що для роботи на комп'ютері більше підходять вертикальне і злегка похиле положення. Можливо, щоб крісло було злегка нахилене вперед.

Схеми розміщення робочих місць із персональним комп'ютером повинні ураховувати відстані між робочими столами з відеомоніторами (в напрямку тилу поверхні одного відеомонітора й екрана іншого відеомонітора), які повинні бути не менш 2 м, а відстань між бічними поверхнями відеомоніторів – не менш 1,2 м.

Робочий стіл повинен мати простір для ніг висотою не менш 600 мм, шириною не менш 500 мм, глибиною на рівні колін не менш 450 мм і на рівні витягнутих ніг не менш 650 мм.

Дисплей. Положення тіла звичайно відповідає напрямку погляду; дисплеї, розташовані занадто низько чи під неправильним кутом, є основними причинами сутулості. Відстань від дисплея до очей може варіюватися в залежності від характеру виконуваної роботи – 40-70 см. При роботі з текстом відстань від екрана до очей повинна лише небагато перевищувати відстань між

книгою й очима і складати 40-45 см. Необхідно давати відпочинок очам і час від часу їх просто закривати. Не можна допускати, щоб очі увесь час були сфокусовані на одній відстані, працюючи з текстом, рекомендується установлювати великий шрифт. Дуже важливо, щоб екран монітора не мерехтів. Частота регенерації зображення повинна бути не менш 72 Гц, тому що при більш низькій частоті помітне мерехтіння, хоча люди з особливо чутливим зором можуть помітити його і при більш високій частоті. Їм доведеться підібрати графічну плату і монітор з частотою регенерації 85 Гц і вище.

Необхідно уникати того, щоб монітор був звернений убік вікна, оскільки інтенсивна освітленість області зору може затопити потоками світла очі і розмити зображення на сітківці. Якщо приходить сидіти поруч з вікном, то треба розташуватися під прямим кутом до нього, причому екран дисплея повинний бути перпендикулярний шибці – цим виключаються відблиски на екрані.

Для зниження впливу низькочастотного електромагнітного випромінювання монітора на ЕЛТ рекомендується вибирати монітор, що задовольняє стандарту MPR II, чи TCO 92-95. У протилежному випадку необхідно установити на екран захисний фільтр, найбільш сучасні моделі, затримують до 99 % електромагнітного випромінювання.

Форма спинки крісла повинна повторювати форму спини працюючого. Крісло необхідно установити на такій висоті, щоб не почувався тиск на куприк (крісло розташоване занадто низько) на стегна (крісло розташоване занадто високо). Фахівці з ергономіки, вважали, що кут між стегнами і хребтом повинний складати 90°, однак недавно проведені дослідження показали, що більшість людей переважно сидять трохи відкинувшись.

Клавіатуру необхідно установити так, щоб не треба було до неї тягтися і нахилитися вперед. При зміні положення тіла (наприклад, з вертикального на похиле) обов'язково треба перемінити положення клавіатури і дисплея. Може виявитися корисною регульована підставка клавіатури, завдяки якій можна без усякої напруги працювати з маніпулятором "миша". Можна поставити клавіатуру і на коліна. Руки при роботі повинні бути прямими в зап'ястях і зігнуті в ліктях приблизно під прямим кутом. Пальці також повинні бути злегка зігнуті. Удари по клавішах не повинні бути занадто сильними.

Зручна висота столу особливо важлива в тому випадку, коли на ньому розташовується клавіатура. Якщо в клавіатури немає підставки, а висоту столу не можна змінити (і він занадто високий), то треба вище підняти сидіння крісла, а під ноги підставити лавочку: чи що-небудь інше. Якщо стіл занадто низький, необхідно підкласти що-небудь під його ніжки.

Щогодини необхідно робити перерву в роботі. У цей час рекомендується робити вправи для зап'ясть. Нижче описаний можливий комплекс вправ.

- Покласти руку на край столу долонею вниз. Взявшись, за пальці, іншою рукою відвести кисть назад і утримувати протягом 5 секунд. Повторити для іншої руки.

- Злегка упертися рукою в стіл і на 5 секунд напружити пальці в зап'ястя. Повторити іншою рукою.

- Сильно зжати пальці, а потім розпрямити їх.

Виконання приведених рекомендацій дозволяє скоротити вплив шкідливих виробничих факторів на організм людини.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Аналіз існуючої проблеми необхідності охорони навколишнього середовища

Охорона навколишнього природного середовища, раціональне використання природних ресурсів, забезпечення екологічної безпеки життєдіяльності людини – невід'ємна умова сталого економічного та соціального розвитку України.

З цією метою Україна здійснює на своїй території екологічну політику, спрямовану на збереження безпечного для існування живої і неживої природи навколишнього середовища, захисту життя і здоров'я населення від негативного впливу, зумовленого забрудненням навколишнього природного середовища, досягнення гармонійної взаємодії суспільства і природи, охорону, раціональне використання і відтворення природних ресурсів.

Відносини у галузі охорони навколишнього природного середовища в Україні регулюються Законом України № 1268-ХІІ від 26.06.91 р. «Про охорону навколишнього природного середовища» із змінами [23], а також розроблюваними відповідно до нього земельним, водним, лісовим законодавством, законодавством про надра, про охорону атмосферного повітря, про охорону і використання рослинного і тваринного світу та іншим спеціальним законодавством.

Цей Закон визначає правові, економічні та соціальні основи організації охорони навколишнього природного середовища в інтересах нинішнього і майбутніх поколінь.

Завданням законодавства про охорону навколишнього природного середовища є регулювання відносин у галузі охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідації негативного впливу господарської та іншої діяльності на навколишнє природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною.

Охорона навколишнього природного середовища є системою державних і суспільних заходів, направлених на збереження, відтворювання і раціональне використання природних ресурсів і поліпшення полягання природного середовища, і є частиною прикладної екології.

В міру прискорення темпів науково-технічного прогресу дія людей на природу стає все більш могутньою. І в даний час воно вже сумирно із дією природних чинників, що приводить до якісної зміни співвідношення сил між суспільством і природою. На сучасному етапі людство поставлено перед чинником виникнення в природі незворотних процесів, нових шляхів переміщення і перетворення енергії і речовини. В природу втручається все більше і більше нових речовин, чужих їй, часом сильно токсичних для організмів. Частина з них не включається в природний круговорот і нагромаджується в біосфері, що приводить до небажаних екологічних наслідків.

Накопичення промислових відходів сприяє підвищенню захворюваності людей і тварин, прискоренню корозії машин і металевого устаткування,

зниженню врожайності сільськогосподарських культур і продуктивності тваринництва, прискореному і нераціональному використуванню природних ресурсів і енергії, погіршенню багатьох властивостей екологічних систем, загибелі деяких унікальних природних територіальних комплексів, зникненню окремих видів тварин і рослин.

Основну загрозу для навколишнього середовища в межах галузі розробки ПЗ становлять комплекси електронно-обчислювальних машин (надалі ЕОМ), а саме електромагнітні випромінювання від них.

Дослідження останніх років показали, що електромагнітні випромінювання вищезгаданих електронних пристроїв містять торсіонову компоненту, котра несе інформацію про процеси, що відбуваються в тому чи іншому електронному пристрої.

Торсіонові поля мають високу проникаючу здатність і їх неможливо заекранувати.

Останніми роками при проектуванні (конструюванні), виготовленні (будівництві) і експлуатації технічних і систем управління в різних сферах діяльності надзвичайно широко застосовуються персональні комп'ютери і всілякі комп'ютерні програми.

Робота з комп'ютерами програмістів, операторів і інших користувачів пов'язана з додатковими шкідливими і небезпечними чинниками, що негативно впливають на організм людини. Наприклад, шкідлива дія на зір надає не при тримання стандартних візуальних ергономічних параметрів екрану, розмір мінімального елементу відображення, мерехтіння зображення, відбивна здатність (відблиски) і ін. Працюючий комп'ютер створює електромагнітне поле, що шкідливо діє на організм людини. Це поле може викликати радіоперешкоди, тобто заважати роботі радіо і телеапаратури, що призводить до

зниження надійності технічної системи або системи управління, до збільшення ризику виникнення аварійної ситуації у виробничому середовищі. Для забезпечення безпеки роботи з комп'ютером розроблені і повинні повсюдно застосовуватися стандарти на монітори, вимоги до приміщень для експлуатації комп'ютерів і до організації і устаткування робочих місць.

6.2 Утилізація відходів комп'ютерної техніки

Використання комп'ютерів вимагає вирішення таких важливих питань, як утилізація відходів (мікросхеми з вмістом кольорових металів, плати, дискети).

При утилізації старих комп'ютерів відбувається їх розділ на сім фракцій: метали, пластмаси, штекери, дроти, батареї, скло. Жодна деталь не йде для повторного використання, оскільки не можна гарантувати їх надійність, але у формі вторинної сировини вони йдуть на виготовлення нових комп'ютерів або інших пристроїв.

Детально розглянемо декілька прикладів переробки відходів обчислювальної техніки.

Гадолінієво-галлеві гранати (ГГГ) використовуються у виробництві компонентів пристроїв, що запам'ятовують. В ході обробки біля 80 % вихідного матеріалу перетворюється на відходи або відбраковується. ГГГ мають високу вартість і їх виділення з відходів представляє інтерес з економічної точки зору.

При здобутті досить чистих продуктів можливі повторне їх використання як вихідний матеріал. При цьому значно підвищується економічність виробництва заготовок з ГГГ. Під терміном відходи маються на

увазі кристалічні залишки (залишки середовища для зростання кристалів, частини кристалів, виробництва, що утворюються на різних стадіях), а також дрібний порошок, що виходить при різанні, шліфовці, поліровці кристалів граната або подібних матеріалів.

Переробка цих відходів протягом ряду останніх років викликає труднощі і не вирішена до цих пір. Всі попередні спроби були безуспішними із-за низької розчинності цих складних оксидів.

Вдосконалений процес, розроблений Е. Гуссетом (патент США 4-198231 від 15 квітня 1980 року, фірма «Свісс Алюмініум Лтд.», Швейцарія), призначений для виділення галію і гадолінія з відходів, що містять обидва ці елементи у вигляді оксидів або з'єднань, що перекладаються в оксиди. Відходи дрібно подрібнюються і потім розчиняються в сильних мінеральних кислотах. Гадоліній осідає з очищених розчинів у вигляді оксалата, галій виділяється в металевому вигляді електролітично. Електролітичне виділення галію може проводитися до виділення гадолінія у вигляді оксалата з розчину.

Розглянемо приклад проведення такого процесу. Відходи піддаються переробці, є залишки завантаження в пристрої для зростання кристалів, розколені частини кристалів зі всіх стадій переробки, дрібнозернисті порошки і пудри після операції різання, шліфовки і поліровки гранатів $GdGaO$. Змельчений порошок після обробки кристалів ГГГ висушується при 1200 С і потім нагрівається при 6000 С протягом декількох годин для розкладання летких забруднень.

Дрібнозернисті відходи в кількості 1000 г розміром менше 40 мкм., що містять 34 % галію і 46 % гадолінія кип'ятяться із зворотним холодильником протягом двох годин в 2100 мл 35 %-ной соляної кислоти. Це відповідає 99 % завантаження. Після кип'ятіння частина відходів, що не розчинилася,

фільтрується і промивається. Після висушування залишок важить 20 г. Залишки такого типу об'єднуються і піддаються повторній обробці кип'яченням. Фільтрат, що включає промивні води, об'ємом 2300 мл, обробляється 4000 г металевого галію при 500 С в течії 45 хвилин. Метал має максимально диспергувати.

В ході цієї операції благородніші елементи, присутні в розчині, виділяються і частково розчиняються в галії до його насичення і далі охолоджуються у вигляді інтерметалевих включень або в елементарній формі. Висадження можна проводити з меншою кількістю галію. Метал може періодично замінюватися на нову порцію до досягнення необхідної міри очищення. В результаті процесу очищення виходить розчин з вмістом галію 140 г/л і гадолінія 190 г/л.

Встановлюється величина $ph=1$ шляхом додавання 900 мл 4%-ного розчину перекису водню для окислення домішок заліза. Осадження гадолінія проводиться при 500 С шляхом додавання 1500 г кристалічної технічної щавлевої кислоти $COH*2HO$; суспензія акуратно перемішується 12 годин для повторного осадження у вигляді оксалата гадолінія.

Оксалат гадолінія $Gd(CO)*10HO$ відділяється центрифугуванням, промивається 20 мл розбавленої щавлевої кислоти (6 г/л) і висушується при 1300 С; перетворення на оксид гадолінія досягається прожаренням при 8000 С. Подальше очищення може проводитися розчиненням в кислоті і осадженням у вигляді оксалата гадолінія. До рідини після центрифугування і промивань осаду (3300 мл) додають 2300 г КОН при інтенсивному перемішуванні до $ph = 12$, 6000 мл розчину з вмістом галію 55 г/л і гадолінія 1 г/л піддається електролізу при 600 Із з використанням катода з нержавіючої сталі і графітового анода при щільності струму близько 0,1 А/кв.см. Після 48 годин осідає 325 г

галію і залишається розчин з вмістом 0,4 г/л, що не піддається подальшій переробці. Обложений метал має чистоту 99,99 % і може бути безпосередньо перетворений в оксид.

При експлуатації ПК витрачаються наступні ресурси:

- електроенергія;
- папір для принтера;
- картріджи з фарбувальною стрічкою.

Для раціонального використання електроенергії не слід залишати включеним комп'ютер і принтер, якщо вони не потрібні в даний час. Друкувати можна з двох сторін. Витрати на папір навряд чи вдасться скоротити удвічі, проте економія буде вельми істотною. Проблему з утилізацією паперових відходів може вирішити вторинна переробка.

Для повторного використання картріджів з фарбувальною стрічкою необхідно купити пристрій для просочення стрічок і тоді картріджи можна буде використовувати 20-40 раз.

Сучасна технологія виготовлення елементів засобів обчислювальної техніки (СВТ) дозволяє досягти дуже низького рівня відмов елементів під час експлуатації. У зв'язку з цим відпадає необхідність проведення ремонтних робіт на місці експлуатації сучасних засобів обчислювальної техніки і як наслідок не утворюються відходи (несправні мікросхеми), що містять дорогоцінні і рідкоземельні метали. Природно, в сервісних центрах, що спеціалізуються на ремонті і технічному обслуговуванні СВТ, має бути організований збір і облік матеріалів, що містять коштовні метали, з подальшою обробкою цих матеріалів на спеціалізованих заводах з метою їх вилучення. У зв'язку з тим, що вітчизняне виробництво сучасних компонентів інформаційних технологій знаходиться в

сьогоднішні дні лише в зачатковому стані, СВТ складаються з парку імпорتنих машин і устаткування. Із-за відсутності інформації про вміст дорогоцінних металів в елементах устаткування, строгий облік не представляється можливим і має бути покладений на фахівців експортних фірм. В умовах ринкової економіки підприємства мають бути самі зацікавлені у вторинній переробці, що містять дорогоцінні метали вузлів і елементів за умови неможливості їх використання.

Сумарна маса дорогоцінних металів в ПК, використовуваному в розробці даної дипломної роботи приблизно складає: золото - 0,23 г; срібло - 5,09 г.

Технологічний процес витягання дорогоцінних металів з друкарських плат здійснюється за наступною схемою. Друкарські плати сортуються по переважанню в них кількості дорогоцінних металів, дробляться і подрібнюються, обпалюються і плавляться. В процесі випалення перолітичеському розкладанню піддається пластмасова основа, а основа дорогоцінних металів у вигляді металевих залишків відновлюється до оксидів. Металевий залишок подрібнюється, гранулюється, проходить магнітну сепарацію і відбувається відділення магнітних від немагнітних часток. Отриманий таким чином порошок, розділений по видах дорогоцінних металів, у вигляді гранул розплавляється в індукційних плавильних печах з подальшим розділенням кожного металу окремо.

ВИСНОВКИ

В кваліфікаційній роботі вирішене наукове завдання з підвищення достовірності моделювання трудомісткості agile тестування невеликих веб проєктів за допомогою нелінійної регресійної моделі.

1) Для моделювання трудомісткості agile тестування невеликих веб проєктів існують нелінійні регресійні моделі, які побудовані в залежності від трьох факторів: кількість тестових випадків, кількість сторінок програмно-технічної документації та кількість дефектів. Але у зазначених моделях не були наведені значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки. А це не дозволяє здійснювати моделювання трудомісткості agile тестування. Тому виникає потреба в побудові відповідних моделей для моделювання трудомісткості agile тестування невеликих веб проєктів.

2) Перевірку даних для побудови нелінійної регресійної моделі для моделювання трудомісткості agile тестування невеликих веб проєктів на наявність викидів виконано за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса. З'ясовано, що за методом, який оснований на нормалізуючих перетвореннях і квадраті відстані Махаланобіса, чотиривимірні дані містять один чотиривимірний викид. Застосування критерія на основі квадрата відстані Махаланобіса до даних без їх нормалізації показало, що дані два чотиривимірних викиди. Такий результат пояснюється тим, що розподіл даних не є гаусівським.

3) Удосконалено нелінійну регресійну модель для моделювання трудомісткості agile тестування невеликих веб проєктів на основі

нормалізуючого перетворення у формі десяткового логарифму із врахуванням викидів у регресії. Для цієї нелінійної регресійної моделі у порівнянні з існуючими регресійними нелінійними моделями визначено оцінку середньо квадратичного відхилення випадкової величини, яка описує похибку, що дає змогу здійснювати моделювання трудомісткості agile тестування невеликих веб проєктів.

4) На підставі запропонованої нелінійної регресійної моделі розроблено програму для моделювання трудомісткості agile тестування невеликих веб проєктів для системи моделювання Scilab 6.0.0. 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, що наведена у додатку Д. За результатами випробувань зроблено висновок про працездатність розробленої програми для моделювання трудомісткості agile тестування невеликих веб проєктів для системи моделювання Scilab 6.0.0.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Olausson M. Introduction to Agile Planning, Development, and Testing / M. Olausson, J. Rossberg, J. Ehn, M. Sköld // Pro Team Foundation Service / M. Olausson, J. Rossberg, J. Ehn, M. Sköld. – Berkeley, CA: Apress, 2013. – Chapter 2. – P. 9-19. DOI: https://doi.org/10.1007/978-1-4302-5996-1_2
2. Olausson M. Agile Testing / M. Olausson, J. Rossberg, J. Ehn, M. Sköld // Pro Team Foundation Service / M. Olausson, J. Rossberg, J. Ehn, M. Sköld. – Berkeley, CA: Apress, 2013. – Chapter 19. – P. 19-33. DOI: https://doi.org/10.1007/978-1-4302-5996-1_19
3. Penmetsa J.R. Agile Testing / J. R. Penmetsa // Trends in Software Testing / H. Mohanty, J. Mohanty, A. Balakrishnan (eds). – Singapore: Springer, 2017. – Chapter 2. – P. 19-33. DOI: [10.1007/978-981-10-1415-4_2](https://doi.org/10.1007/978-981-10-1415-4_2)
4. Nader-Rezvani N. Agile Quality Test Strategy / N. Nader-Rezvani // An Executive's Guide to Software Quality in an Agile Organization / N. Nader-Rezvani. – Berkeley, CA: Apress, 2019. – Chapter 7. – P. 121-138. DOI: https://doi.org/10.1007/978-1-4842-3751-9_7
5. Agile Testing – Principles, methods & advantages [Electronic resource]. – Access mode: <https://reqtest.com/testing-blog/agile-testing-principles-methods-advantages/>
6. Lazić L. Challenges in estimating software testing effort / L. Lazić, I. Dokić, S. Milinković // Proceedings of INFOTEH-JAHORINA. – 2014. – Vol. 13. – P. 637-642.
7. Prykhodko N. V. A multiple non-linear regression model to estimate the agile testing efforts for small Web projects / N. V. Prykhodko, S. B. Prykhodko //

Radio Electronics, Computer Science, Control. – No.2 (49) – 2019. – P.158-166. – ISSN: 1607-3274 – E-ISSN: 2313-688X – DOI: <https://doi.org/10.15588/1607-3274-2019-2-17>

8. Приходько С.Б. Комп'ютерне моделювання випадкової залежної змінної за нелінійним рівнянням регресії / С. Б. Приходько // Проблеми моделювання та розроблення інформаційних систем: матеріали науково-практичної інтернет-конференції (Дрогобич, 5-7 квітня 2016 року). – Дрогобич: ДДПУ ім. І. Франка, 2016. – С.151-154.

9. Prykhodko S. Mathematical Modeling of Non-Gaussian Dependent Random Variables by Nonlinear Regression Models Based on the Multivariate Normalizing Transformations / S. Prykhodko, N. Prykhodko // In: Shkarlet S., Morozov A., Palagin A. (eds) Mathematical Modeling and Simulation of Systems (MODS'2020). MODS 2020. Advances in Intelligent Systems and Computing, vol 1265, 2021. – P. 166-174. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-58124-4_16

10. Приходько С.Б. Математичне моделювання трудомісткості agile тестування невеликих веб проєктів за нелінійною регресійною моделлю / С. Б. Приходько, А. В. Пухалевич, **Є. В. Сорокіна** // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2022): Матеріали III-ої Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2022 р.). – Миколаїв: НУК імені адмірала Макарова, 2022. – С. 18-19. Режим доступу: <http://itconf.nuos.edu.ua/2022/wp-content/uploads/sites/5/NUOS-ITMAS-2022-Proceedings.pdf>

11. Foss T. A simulation study of the model evaluation criterion MMRE / T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit // IEEE Transactions on software

engineering. – 2003. – 11(29). – P. 985–995. DOI: <https://doi.org/10.1109/TSE.2003.1245300>

12. Port, D. Comparative studies of the model evaluation criteria MMRE and PRED in software cost estimation research / D. Port, M. Korte // In: Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. – ACM, New York, 2008. – P. 51–60.

13. Prykhodko N.V. Constructing the non-linear regression models on the basis of multivariate normalizing transformations / N. V. Prykhodko, S. B. Prykhodko // Electronic modeling. – 2018. – T.40. – № 6. – P. 99-108. – ISSN 0204–3572. DOI: <https://doi.org/10.15407/emodel.40.06.101>

14. Mardia K.V. Measures of multivariate skewness and kurtosis with applications / K. V. Mardia // Biometrika. – 1970. – Vol. 57. – P. 519–530. DOI: <https://doi.org/10.1093/biomet/57.3.519>

15. Prykhodko S. Detecting Outliers in Multivariate Non-Gaussian Data on the basis of Normalizing Transformations / S. Prykhodko, N. Prykhodko, L. Makarova, K. Pugachenko // Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) «Celebrating 25 Years of IEEE Ukraine Section», Kyiv, Ukraine, May 29 – June 2, 2017. – P. 846-849. DOI: <https://doi.org/10.1109/UKRCON.2017.8100366>

16. Prykhodko S. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data / S. Prykhodko, N. Prykhodko, L. Makarova, A. Pukhalevych // Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, February 20–24, 2018. – P. 962-965. DOI: <https://doi.org/10.1109/TCSET.2018.8336353>

17. Johnson R.A. Applied Multivariate Statistical Analysis / R. A. Johnson, D. W. Wichern. – Pearson Prentice Hall, 2007. – 800 p.

18. Prykhodko S. A Modified Technique for Constructing Nonlinear Regression Models Based on the Multivariate Normalizing Transformations / Sergiy Prykhodko, Natalia Prykhodko // Selected Papers of the VIII International Scientific Conference “Information Technology and Implementation” (IT&I-2021). Workshop Proceedings (IT&I-WS-2021), Kyiv, Ukraine, December 1-3, 2021. CEUR Workshop Proceedings. – 2022. – Vol.3179. – CEUR-WS.org – P. 156-166. ISSN 1613-0073. http://ceur-ws.org/Vol-3179/Paper_15.pdf

19. Prykhodko S. Outlier Detection in Non-Linear Regression Analysis Based on the Normalizing Transformations / Sergiy Prykhodko, Natalia Prykhodko, Lidiia Makarova, Andrii Pukhalevych // Proceedings of the 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), IEEE, Lviv-Slavske, 2020, P. 407-410. <https://doi.org/10.1109/TCSET49122.2020.235464>

20. Приходько С.Б. Генерування випадкових чисел з розподілом Гауса на основі перетворення Джонсона із сім'ї S_U [Електронний ресурс] / С. Б. Приходько // Електронне видання "Вісник Національного університету кораблебудування". – Миколаїв: НУК, 2013. – № 4. – Режим доступу: <http://evn.nuos.edu.ua/article/view/28205/25142> або <http://ev.nuos.edu.ua/content/>

21. Приходько С.Б. Моделювання гаусівських випадкових величин із використанням перетворення Джонсона із сім'ї S_U / С. Б. Приходько // Інформатика та математичні методи в моделюванні. – 2015. – т.5, №.1. – С. 92-97.

22. Закон України «Про охорону праці» (із змінами). Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

23. Законом України № 1268-ХІІ від 26.06.91 р. «Про охорону навколишнього природного середовища» (із змінами). Режим доступу: <https://zakon.rada.gov.ua/laws/show/1264-12#Text>

Додаток А – Технічне завдання на розробку програми для моделювання трудомісткості agile тестування невеликих веб проєктів

Вступ

Назва розробки – “Програма для моделювання трудомісткості agile тестування невеликих веб проєктів”, далі програма.

Підстави для розробки

Завдання на кваліфікаційну роботу.

Призначення розробки:

Програма призначена для моделювання трудомісткості agile тестування невеликих веб проєктів 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 емпіричні дані для моделювання трудомісткості agile тестування невеликих веб проєктів: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

2) Вводити з інтерфейсу програми довірчу ймовірність, три метрики за якими потрібно змодельовати трудомісткість agile тестування невеликого веб проєкту (у годинах): кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

3) Виконувати нормалізацію емпіричних даних за нормалізуючим перетворенням у формі десяткового логарифму.

4) Знаходити значення коефіцієнтів лінійного рівняння регресії для нормалізованих емпіричних даних.

5) Здійснювати побудову нелінійної регресійної моделі.

6) Обчислювати значення коефіцієнту детермінації R^2 .

7) Отримувати трудомісткість, нелінійну регресію, її довірчий інтервал та інтервал прогнозування.

8) Виводити на екран трудомісткість, нелінійну регресію, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах), значення коефіцієнту детермінації R^2 .

9) Виводити у текстовий файл результати моделювання трудомісткості agile тестування невеликих веб проєктів у годинах.

Вимоги до організації вхідних та вихідних даних

Вхідні дані:

- емпіричні дані для моделювання трудомісткості agile тестування невеликих веб проєктів: трудомісткість agile тестування (у годинах) Y , кількість тестових випадків X_1 ; кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 ;

- значення довірчої ймовірності, значення трьох метрик, за якими потрібно змодельовати трудомісткість agile тестування невеликого веб проєкту (у годинах): кількість тестових випадків X_1 , кількість сторінок програмно-технічної документації X_2 та кількість дефектів X_3 .

Вихідні дані:

- значення трудомісткості, нелінійної регресії, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості agile тестування невеликого веб проєкту (у годинах);

- значення коефіцієнту детермінації.

Вимоги до надійності не висуваються.

Вимоги до умов експлуатації не висуваються.

Вимоги до складу й параметрів технічних засобів.

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.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0., компілятор C (Visual Studio 2010 або Visual Express 2010) для компіляції зовнішніх модулів C або C++ та використання моделі в Xcos.

Вимоги до програмної документації.

Програмна документація має включати: 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	11.10.2022
2. Ескізний проект	Розробка ескізного проекту	13.10.2022
	Затвердження ескізного проекту	14.10.2022
3. Технічний проект	Розробка технічного проекту	20.10.2022
	Затвердження технічного проекту	21.10.2022
4. Робочий проект	Розробка програми	04.11.2022
	Розробка програмної документації	10.11.2022
	Випробування програми	11.11.2022

Порядок контролю і приймання

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

Для приймання програми надається програмна документація, яка була розроблена відповідно до технічного завдання. Представники замовника установлюють відповідність програми технічному завданню.

За результатами приймання програми представниками замовника складається акт приймання програми.

Додаток Б – Текст програми

```
//Моделювання трудомісткості agile тестування невеликих веб проєктів
f=figure();//Создание графического окна.
//
set(f,'position',[0,0,700,300])
//
set(f,'figure_name',...
'Моделювання трудомісткості agile тестування невеликих веб проєктів');

//plot2d(x,y,-9);

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\WebTestEffort_40.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 трудомісткості','position',...
[500,270,190,20]);
lab_text3=uicontrol(f,'style','text','string',...
'Дані для моделювання','position',...
[60,270,117,20]);
lab_text4=uicontrol(f,'style','text','string',...
'Кількість тестових випадків','position',...
[20,240,200,20]);
edit_text4=uicontrol(f,'style','edit','string',...
'56','position',[230,240,60,20]);
lab_text5=uicontrol(f,'style','text','string',...
'Кількість сторінок документації','position',...
[20,210,200,20]);
edit_text5=uicontrol(f,'style','edit','string',...
'25','position',[230,210,60,20]);
lab_text6=uicontrol(f,'style','text','string',...
'Кількість дефектів','position',...
[20,180,200,20]);
edit_text6=uicontrol(f,'style','edit','string',...
'19','position',[230,180,60,20]);
lab_text7=uicontrol(f,'style','text','string',...
'Довірча ймовірність','position',...
[20,150,200,20]);
edit_text7=uicontrol(f,'style','edit','string',...
'0.95','position',[230,150,60,20]);
lab_text8=uicontrol(f,'style','text','string',...
'Результати моделювання','position',...
[50,120,132,20]);
lab_text9=uicontrol(f,'style','text','string',...
'Результат нелін. регресії трудомісткості (год.)','position',...
```

```

[20,90,245,20]);
lab_textl0=uicontrol(f,'style','text','string',...
'Нижня границя довір.інтервалу трудом. (год.)','position',...
[20,60,245,20]);
lab_textl1=uicontrol(f,'style','text','string',...
'Верхня границя довір.інтервалу трудом. (год.)','position',...
[20,30,245,20]);
lab_textl2=uicontrol(f,'style','text','string',...
'Нижня границя інтервалу прогноз. трудом. (год.)','position',...
[355,60,260,20]);
lab_textl3=uicontrol(f,'style','text','string',...
'Верхня границя інтервалу прогноз. трудом. (год.)','position',...
[355,30,260,20]);
lab_textl4=uicontrol(f,'style','text','string',...
'Значення трудомісткості за нелін. моделлю (год.)','position',...
[355,90,260,20]);
//
textresult=uicontrol(f,'style','text','string','', 'position',...
[270,90,70,20]);
textresult2=uicontrol(f,'style','text','string','', 'position',...
[270,60,70,20]);
textresult3=uicontrol(f,'style','text','string','', 'position',...
[270,30,70,20]);
textresult4=uicontrol(f,'style','text','string','', 'position',...
[190,120,250,20]);
textresult5=uicontrol(f,'style','text','string','', 'position',...
[620,60,70,20]);
textresult6=uicontrol(f,'style','text','string','', 'position',...
[620,30,70,20]);
textresult7=uicontrol(f,'style','text','string','', 'position',...
[620,90,70,20]);

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_textl2=uicontrol(f,'style','edit','string',..
    '0','position',[500,210,60,20]);
end;

Btnr_pred_int=uicontrol('style','pushbutton','string','Змоделювати','CallBack',.
..
'WebTestEffortIntEst(x,yx,n)', 'position',[330,150,110,20]);

BtClose=uicontrol('style','pushbutton','string','Закрити',...
'CallBack','W_Close()','position',[350,5,100,20]);

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

```

```

file_PC=mopen(dir_name,'r');
file_PC2=mopen('D:\Data_F\WebTestEffort_40_data2.txt','w+');
file_PC3=mopen('D:\Data_F\WebTestEffort_40_data3.txt','w+');
i=1;
//n=52;
while (feof(file_PC))<1
    y(i)=mfscanf(file_PC,'%f'); x1(i)=mfscanf(file_PC,'%f');
    x2(i)=mfscanf(file_PC,'%f'); x3(i)=mfscanf(file_PC,'%f');
    i=i+1;
end;
n=i-1;

fclose(file_PC);

for i = 1:n
    yx(i,1)=y(i); yx(i,2)=x1(i); yx(i,3)=x2(i); yx(i,4)=x3(i);
    x(i,1)=x1(i); x(i,2)=x2(i); x(i,3)=x3(i);
end;
//for i=1:n
    fprintf(file_PC2,'%f\t%f\t%f\t%f\n',y,x1,x2,x3);
//end;

for i = 1:n
    for j = 1:3 fprintf(file_PC3, "%5.3f ", x(i,j)); end;
    fprintf(file_PC3, "\n");
end;

fclose(file_PC2);
fclose(file_PC3);
f1=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f1,'position',[50,50,700,300]);
//Устанавливаем заголовок окна.
set(f1,'figure_name','Емпіричні дані: трудомісткість (у год.) в залежності від
кількості тестових випадків');
plot2d(x1,y,-9);

f2=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f2,'position',[50,50,700,300]);
//Устанавливаем заголовок окна.
set(f2,'figure_name','Емпіричні дані: трудомісткість (у год.) в залежності від
кількості сторінок програмно-технічної документації');
plot2d(x2,y,-9);

f3=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f3,'position',[50,50,700,300]);
//Устанавливаем заголовок окна.
set(f3,'figure_name','Емпіричні дані: трудомісткість (у год.) в залежності від
кількості дефектів');
plot2d(x3,y,-9);

x1A=0; x2A=0; x3A=0;

```

```

for i=1:n
    x1A=x1A+x(i,1); x2A=x2A+x(i,2); x3A=x3A+x(i,3);
end;
x1A=x1A/n; x2A=x2A/n; x3A=x3A/n;
Sx1x1=0; Sx2x2=0; Sx3x3=0;
Sx1x2=0; Sx1x3=0; Sx2x3=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    Sx1x2=Sx1x2+(x(i,1)-x1A)*(x(i,2)-x2A);
    Sx1x3=Sx1x3+(x(i,1)-x1A)*(x(i,3)-x3A);
    Sx2x3=Sx2x3+(x(i,2)-x2A)*(x(i,3)-x3A);
    Sx2x2=Sx2x2+(x(i,2)-x2A)*(x(i,2)-x2A);
    Sx3x3=Sx3x3+(x(i,3)-x3A)*(x(i,3)-x3A);
end;
Sxx(1,1)=Sx1x1; Sxx(1,2)=Sx1x2; Sxx(1,3)=Sx1x3;
Sxx(2,1)=Sx1x2; Sxx(2,2)=Sx2x2; Sxx(2,3)=Sx2x3;
Sxx(3,1)=Sx1x3; Sxx(3,2)=Sx2x3; Sxx(3,3)=Sx3x3;

Sxx_1=1/Sxx;

z1A=0; z2A=0; z3A=0;
for i=1:n
    z1A=z1A+log10(x(i,1)); z2A=z2A+log10(x(i,2)); z3A=z3A+log10(x(i,3));
end;
z1A=z1A/n; z2A=z2A/n; z3A=z3A/n;
Sz1z1=0; Sz2z2=0; Sz3z3=0;
Sz1z2=0; Sz1z3=0; Sz2z3=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
    Sz1z2=Sz1z2+(log10(x(i,1))-z1A)*(log10(x(i,2))-z2A);
    Sz1z3=Sz1z3+(log10(x(i,1))-z1A)*(log10(x(i,3))-z3A);
    Sz2z3=Sz2z3+(log10(x(i,2))-z2A)*(log10(x(i,3))-z3A);
    Sz2z2=Sz2z2+(log10(x(i,2))-z2A)*(log10(x(i,2))-z2A);
    Sz3z3=Sz3z3+(log10(x(i,3))-z3A)*(log10(x(i,3))-z3A);
end;
Szz(1,1)=Sz1z1; Szz(1,2)=Sz1z2; Szz(1,3)=Sz1z3;
Szz(2,1)=Sz1z2; Szz(2,2)=Sz2z2; Szz(2,3)=Sz2z3;
Szz(3,1)=Sz1z3; Szz(3,2)=Sz2z3; Szz(3,3)=Sz3z3;

Szz_1=1/Szz;

zi(1)=log10(x(1,1))-z1A; zi(2)=log10(x(1,2))-z2A; zi(3)=log10(x(1,3))-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

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

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

//      Delta_z=T/n^0.5;

```

```

b0=-0.45001;
b1=0.7888688;
b2=0.001760;
b3=0.1878210;

Szyzyn4=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))+b2*log10(x(i,2))+b3*log10(x(i,3))-log10(yx(i,1)))^2;
end;
Szyzyn4=Szyzyn4/(n-4)
Delta_z=T*(Szyzyn4*(1/n+result))^0.5;

zy_result=b0+b1*log10(x(1,1))+b2*log10(x(1,2))+b3*log10(x(1,3));

zyLBC=zy_result-Delta_z; zyUBC=zy_result+Delta_z;

y_result=10^zy_result;
yLBC=10^zyLBC; yUBC=10^zyUBC;

file_PC5=fopen('D:\Data_F\Web_Django_30_data5.txt','w+');
    fprintf(file_PC5, "%5.6f ", yUBC);

fclose(file_PC5);

file_PC4=fopen('D:\Data_F\WebTestEffort_40_data4.txt','w+');
for i = 1:3
    for j = 1:3 fprintf(file_PC4, "%5.6f ", Szz_1(i,j)); end;
    fprintf(file_PC4, "\n");
end;
fclose(file_PC4);

endfunction

//Функція визначення трудомісткості agile тестування невеликого веб проекту
function WebTestEffortIntEst(x, yx, n)
//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
x1i=eval(get(edit_text4,'string'));
x2i=eval(get(edit_text5,'string'));
x3i=eval(get(edit_text6,'string'));
Pconf=eval(get(edit_text7,'string'));

x1A=0; x2A=0; x3A=0; yA=0;
for i=1:n
    x1A=x1A+x(i,1); x2A=x2A+x(i,2); x3A=x3A+x(i,3); yA=yA+yx(i,1);
end;
x1A=x1A/n; x2A=x2A/n; x3A=x3A/n; yA=yA/n;

Sx1x1=0; Sx2x2=0; Sx3x3=0;
Sx1x2=0; Sx1x3=0; Sx2x3=0; SyA=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    Sx1x2=Sx1x2+(x(i,1)-x1A)*(x(i,2)-x2A);

```

```

    Sx1x3=Sx1x3+(x(i,1)-x1A)*(x(i,3)-x3A);
    Sx2x3=Sx2x3+(x(i,2)-x2A)*(x(i,3)-x3A);
    Sx2x2=Sx2x2+(x(i,2)-x2A)*(x(i,2)-x2A);
    Sx3x3=Sx3x3+(x(i,3)-x3A)*(x(i,3)-x3A);
    SyA=SyA+(yx(i,1)-yA)*(yx(i,1)-yA);
end;
Sxx(1,1)=Sx1x1; Sxx(1,2)=Sx1x2; Sxx(1,3)=Sx1x3;
Sxx(2,1)=Sx1x2; Sxx(2,2)=Sx2x2; Sxx(2,3)=Sx2x3;
Sxx(3,1)=Sx1x3; Sxx(3,2)=Sx2x3; Sxx(3,3)=Sx3x3;

Sxx_1=1/Sxx;

z1A=0; z2A=0; z3A=0;
for i=1:n
    z1A=z1A+log10(x(i,1)); z2A=z2A+log10(x(i,2)); z3A=z3A+log10(x(i,3));
end;
z1A=z1A/n; z2A=z2A/n; z3A=z3A/n;
Sz1z1=0; Sz2z2=0; Sz3z3=0;
Sz1z2=0; Sz1z3=0; Sz2z3=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
    Sz1z2=Sz1z2+(log10(x(i,1))-z1A)*(log10(x(i,2))-z2A);
    Sz1z3=Sz1z3+(log10(x(i,1))-z1A)*(log10(x(i,3))-z3A);
    Sz2z3=Sz2z3+(log10(x(i,2))-z2A)*(log10(x(i,3))-z3A);
    Sz2z2=Sz2z2+(log10(x(i,2))-z2A)*(log10(x(i,2))-z2A);
    Sz3z3=Sz3z3+(log10(x(i,3))-z3A)*(log10(x(i,3))-z3A);
end;
Szz(1,1)=Sz1z1; Szz(1,2)=Sz1z2; Szz(1,3)=Sz1z3;
Szz(2,1)=Sz1z2; Szz(2,2)=Sz2z2; Szz(2,3)=Sz2z3;
Szz(3,1)=Sz1z3; Szz(3,2)=Sz2z3; Szz(3,3)=Sz3z3;

Szz_1=1/Szz;

zi(1)=log10(x1i)-z1A; zi(2)=log10(x2i)-z2A; zi(3)=log10(x3i)-z3A;
ziT=zi';
ziSzz_1=ziT*Szz_1;
result=ziSzz_1*zi;

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

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

b0=-0.45001;
b1=0.7888688;
b2=0.001760;
b3=0.1878210;

Szyzyn4=0; Syy=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))+b2*log10(x(i,2))+b3*log10(x(i,3))-log10(yx(i,1)))^2;

```



```

        Syy=Syy+(yx(i,1)-10^b0*x(i,1)^b1*x(i,2)^b2*x(i,3)^b3)^2;
    end;
    Szyzyn4=Szyzyn4/(n-4)
    Delta_z=T*(Szyzyn4*(1/n+result))^0.5;
    Delta_z_pr=T*(Szyzyn4*(1+1/n+result))^0.5;

    r_2=1-Syy/SyA;

    zy_result=b0+b1*log10(x1i)+b2*log10(x2i)+b3*log10(x3i);

    zyLBC=zy_result-Delta_z; zyUBC=zy_result+Delta_z;

    zyLBPr=zy_result-Delta_z_pr; zyUBPr=zy_result+Delta_z_pr;

    y_result=10^zy_result;
    yLBC=10^zyLBC; yUBC=10^zyUBC;
    yLBPr=10^zyLBPr; yUBPr=10^zyUBPr;

    r = grand(1, 1, "unf", -1, 1);
    xtan=tan(0.57*r);
    zd=0.532275*asinh(3.874868*xtan);
    zd=0.0294*zd;

    zy_random=zy_result+zd;
    y_random=10^zy_random;

    file_PC5=mopen('D:\Data_F\WebTestEffort_40_data5.txt','w+');
    //for i=1:3
        fprintf(file_PC5, "%5.6f ", yUBC);
    //end
    mclose(file_PC5);

    file_PC4=mopen('D:\Data_F\WebTestEffort_40_data4.txt','w+');
    for i = 1:3
        for j = 1:3 fprintf(file_PC4, "%5.6f ", Szz_1(i,j)); end;
        fprintf(file_PC4, "\n");
    end;
    mclose(file_PC4);

    set(textresult,'string', sprintf("=%1.4f",y_result));
    set(textresult2,'string', sprintf("=%1.4f",yLBC));
    set(textresult3,'string', sprintf("=%1.4f",yUBC));
    set(textresult4,'string',...
        sprintf("Коефіцієнт детермінації R^2=%1.4f",r_2));
    set(textresult5,'string', sprintf("=%1.4f",yLBPr));
    set(textresult6,'string', sprintf("=%1.4f",yUBPr));
    set(textresult7,'string', sprintf("=%1.4f",y_random));

    if (get(audio_bikv_x,'value')==0)
    else
        yfact=eval(get(edit_text12,'string'));
        MRE=abs((yfact-y_result)/yfact);
    end

```

```

    textresult5=uicontrol(f,'style','text','string','', 'position',...
    [500,120,160,20]);
    set(textresult5,'string',...
    sprintf("Відносна похибка MRE=%1.4f",MRE));
end;

endfunction

//Функція закриття вікон
function W_Close()
close(f);
close(f1);
close(f2);
close(f3);
endfunction

```

ДОДАТОК В – Опис програми

Загальні відомості

Програма для моделювання трудомісткості agile тестування невеликих веб проєктів, для системи моделювання Scilab 6.0.0. Файл програми WebTestEffortSim.sce має розмір 14 КБ.

Функціональне призначення

Програма WebTestEffortSim призначений для моделювання трудомісткості agile тестування невеликих веб проєктів, для системи моделювання Scilab 6.0.0.

Технічні засоби, що використовуються

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.

Виклик та завантаження

Для запуску та роботи з програмою необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0.

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

ДОДАТОК Г – Інструкція користувача

Загальний опис роботи з програмою

Після завантаження файлу WebTestEffortSim.sce на екрані з'являється форма для відображення початкового стану програми для моделювання трудомісткості agile тестування невеликих веб проєктів (рисунок Г.1).

Дані для моделювання		Файл з даними		Наявність фактичної трудомісткості	
Кількість тестових випадків	56	Файл з даними	...ta_F\WebTestEffort_40.txt	Наявність фактичної трудомісткості	☒ є (включити), немає (виключити)
Кількість сторінок документації	25	Ввести		0	
Кількість дефектів	19	Змодельовати			
Довірча ймовірність	0.95				
Результати моделювання					
Результат нелін. регресії трудомісткості (год.)		Значення трудомісткості за нелін. моделлю (год.)			
Нижня границя довір.інтервалу трудом. (год.)		Нижня границя інтервалу прогноз. трудом. (год.)			
Верхня границя довір.інтервалу трудом. (год.)		Верхня границя інтервалу прогноз. трудом. (год.)			
Закрити					

Рисунок Г.1 – Форма для відображення початкового стану програми

Далі у вікні редагування «Файл з даними» графічного вікна «Моделювання трудомісткості agile тестування невеликих веб проєктів» (рис.Г.1) потрібно відредагувати ім'я текстового файлу з початковими

емпіричними даними, що будуть використовуватися для моделювання трудомісткості agile тестування невеликих веб проєктів.

Після натискання на командну кнопку «Ввести» з текстового файлу вводяться дані.

Потім у графічному вікні «Моделювання трудомісткості agile тестування невеликих веб проєктів» (рис. Г.1) у відповідних вікнах редагування потрібно відредагувати: кількість тестових випадків, кількість сторінок програмно-технічної документації, кількість дефектів, довірчу ймовірність.

Крім того у графічному вікні «Моделювання трудомісткості agile тестування невеликих веб проєктів» (рис. Г.1) потрібно за допомогою радіо кнопки здійснити вибір наявності фактичної трудомісткості agile тестування та ввести його для подальшого порівняння з відповідною оцінкою трудомісткості.

Далі потрібно натиснути на командну кнопку «Змоделювати». Після цього з'являться результати моделювання трудомісткості agile тестування невеликого веб проєкту, метрики для якого були введені.

І на останок у графічному вікні «Моделювання трудомісткості agile тестування невеликих веб проєктів» 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 моделювання трудомісткості agile тестування невеликих веб проєктів WebTestEffortSim.

Ціль випробувань: перевірка відповідності програми WebTestEffortSim 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 WebTestEffortSim. Далі виконується перевірка роботи програми. Виконується перевірка всіх екранних форм програми, а також всіх допустимих для виконання операцій всередині екранних форм згідно до посібника користувача. Якщо програма демонструє адекватне функціонування згідно до вимог ТЗ та представленої програмної документації, то робота з

розробки програми вважається виконаною, і виконується впровадження програми до експлуатації. В протилежному випадку програма має бути відправлено на доробку розробникові.

Використовуються тести за методологією «чорної скриньки» (рис.Д.1-Д.5).

Моделювання трудомісткості agile тестування невеликих веб проєктів

File Tools Edit ?

Моделювання трудомісткості agile тестування невеликих веб проєктів

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості
Кількість тестових випадків	56	ta_F\WebTestEffort_40.txt	☒ є (включити), немає (виключити)
Кількість сторінок документації	25	Ввести	0
Кількість дефектів	19		
Довірча ймовірність	0.95	Змоделювати	
Результати моделювання			
Результат нелін. регресії трудомісткості (год.)		Значення трудомісткості за нелін. моделлю (год.)	
Нижня границя довір.інтервалу трудом. (год.)		Нижня границя інтервалу прогноз. трудом. (год.)	
Верхня границя довір.інтервалу трудом. (год.)		Верхня границя інтервалу прогноз. трудом. (год.)	
Закрити			

Рисунок Д.1 – Графічне вікно головного модуля програми

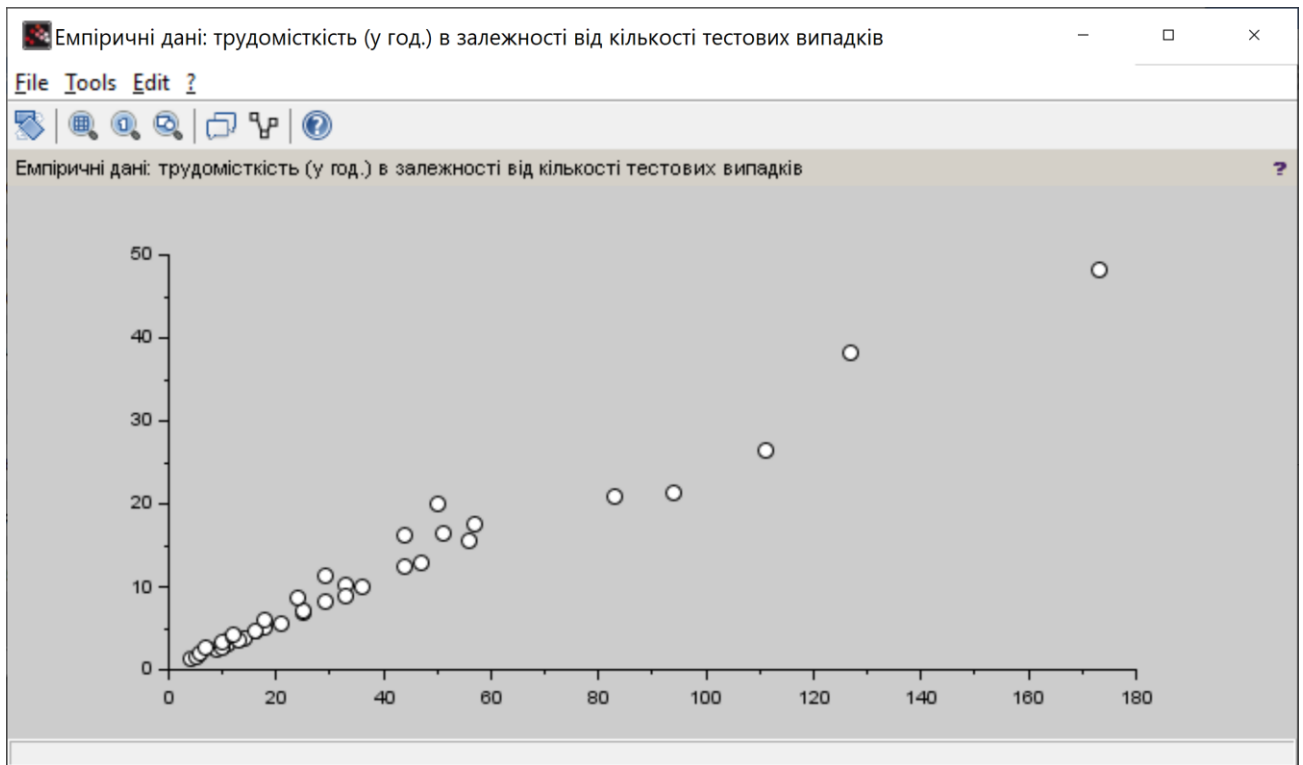


Рисунок Д.2 – Візуалізація тестових емпіричних даних.

Результати роботи програми після натискання на командну кнопку “Змодельовати» у разі, коли радіо кнопка виключена, наведені на рисунку Д.3.

Моделювання трудомісткості agile тестування невеликих веб проєктів

File Tools Edit ?

Моделювання трудомісткості agile тестування невеликих веб проєктів

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості
Кількість тестових випадків	56	ta_F\WebTestEffort_40.txt	☐ є (включити), немає (виключити)
Кількість сторінок документації	25	Ввести	0
Кількість дефектів	19		
Довірча ймовірність	0.95	Змоделювати	

Результати моделювання Коефіцієнт детермінації $R^2=0.9810$

Результат нелін. регресії трудомісткості (год.)	=14.8497	Значення трудомісткості за нелін. моделлю (год.)	=15.4883
Нижня границя довір.інтервалу трудом. (год.)	=14.3026	Нижня границя інтервалу прогноз. трудом. (год.)	=12.8100
Верхня границя довір.інтервалу трудом. (год.)	=15.4177	Верхня границя інтервалу прогноз. трудом. (год.)	=17.2142

Закрити

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

Результати роботи програми після натискання на командну кнопку «Змоделювати» у разі, коли радіо кнопка включена та введена фактична трудомісткість agile тестування невеликого веб проєкту, наведені на рисунку Д.4.

Моделювання трудомісткості agile тестування невеликих веб проєктів

File Tools Edit ?

Моделювання трудомісткості agile тестування невеликих веб проєктів

Дані для моделювання

Кількість тестових випадків: 56

Кількість сторінок документації: 25

Кількість дефектів: 19

Довірча ймовірність: 0.95

Файл з даними

ta_F\WebTestEffort_40.txt

Ввести

Наявність фактичної трудомісткості

☒ є (включити), немає (виключити)

15.56

Змоделювати

Результати моделювання

Коефіцієнт детермінації $R^2=0.9810$

Відносна похибка MRE=0.0456

Результат нелін. регресії трудомісткості (год.)	=14.8497	Значення трудомісткості за нелін. моделлю (год.)	=14.1672
Нижня границя довір.інтервалу трудом. (год.)	=14.3026	Нижня границя інтервалу прогноз. трудом. (год.)	=12.8100
Верхня границя довір.інтервалу трудом. (год.)	=15.4177	Верхня границя інтервалу прогноз. трудом. (год.)	=17.2142

Закрити

Рисунок Д.4 – Результати роботи програми після натискання на командну кнопку “Змоделювати» у разі, коли радіо кнопка включена та введена фактична трудомісткість – 15,56 год.

При випробуванні програми потрібно ввести наступні тестові дані із текстового файлу:

```

1.33 4 11 4
3.51 11 2 7
3.17 10 4 8
1.53 5 14 4
2.35 9 12 3
3.13 10 8 9
2.03 6 16 7
3.22 11 11 6
2.73 10 16 5

```

4.65 16 13 7
5.52 21 12 4
2.75 7 23 12
6.93 25 6 13
3.73 14 29 4
5.08 18 22 7
7.12 25 11 11
3.24 10 31 9
4.05 12 35 6
4.72 16 32 7
3.49 13 41 4
6.03 18 24 18
4.13 12 40 13
10.18 33 15 23
9.95 36 29 12
8.67 24 36 32
16.53 51 3 45
12.45 44 37 18
15.56 56 25 19
17.47 57 7 40
11.23 29 48 47
8.21 29 79 10
19.95 50 6 90
16.16 44 53 61
8.83 33 110 8
12.97 47 94 16
21.32 94 48 37
20.97 83 61 62
38.22 127 25 78
26.48 111 71 54
48.2 173 120 60