

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 розробки програмних проектів та створення програми для її реалізації

Виконав: студент групи 6151м
_____ Машкін О.О.
(підпис, ПІБ)

Керівник роботи:
Доцент кафедри ПЗАС, к.т.н.
(посада, науковий ступень вчене звання)
_____ Пухалевич А.В.
(підпис, ПІБ)

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 Машкіну Олексію Олександровичу
 (Прізвище, ім'я, по батькові)

1. Тема роботи: Математична модель для моделювання трудомісткості розробки програмних проектів та створення програми 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: 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 розробки програмних проектів. Нелінійна регресійна модель для моделювання трудомісткості розробки програмних проектів. Постановка задачі на розробку проекту програми. Діаграма варіантів використання. Діаграма діяльності. Графічне вікно програми із інтерфейсними компонентами. Емпіричні дані. Результати випробувань програми. Висновки.

РЕФЕРАТ

Машкін Олексій Олександрович

«Математична модель для моделювання трудомісткості розробки програмних проектів та створення програми 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., 6 табл., 11 рис., 32 використаних джерела, 5 додатків.

Актуальність теми роботи: необхідність підвищення достовірності моделювання трудомісткості розробки програмних проектів.

Мета та завдання дослідження. Метою є підвищення достовірності моделювання трудомісткості розробки програмних проектів. Завдання дослідження: проаналізувати існуючі регресійні моделі для моделювання трудомісткості розробки програмних проектів; удосконалити регресійну модель для моделювання трудомісткості розробки програмних проектів; розробити програму для моделювання трудомісткості розробки програмних проектів.

Об'єктом дослідження є процес моделювання трудомісткості розробки програмних проектів.

Предметом дослідження є регресійні моделі для моделювання трудомісткості розробки програмних проектів.

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

Наукова новизна одержаних результатів. Удосконалено регресійну модель для моделювання трудомісткості розробки програмних проектів, на основі нормалізуючого перетворення у формі десяткового логарифму, яка дозволяє підвищити достовірність моделювання трудомісткості розробки програмних проектів, в порівнянні з існуючими регресійними моделями.

Практичне значення одержаних результатів. Розроблено програму для моделювання трудомісткості розробки програмних проектів, 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, розробка програмних проектів, моделювання, нелінійна регресійна модель, програмне забезпечення.

ABSTRACT

Mashkin Oleksii Oleksandrovyh

«A mathematical model for simulating the efforts of software project development and creating 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 6 tables, 11 figures, 5 appendices and 32 references.

Relevance of the topic of the work: the need to increase the confidence of simulating the efforts of software project development.

The goal and objectives of the study. The goal of the study is to increase the confidence of simulating the efforts of software project development by nonlinear regression model. The objectives of the study: to analyze existing mathematical models for simulating the efforts of software project development; to improve the regression model for simulating the efforts of software project development; to create the software for simulating the efforts of software project development.

The object of the study is the process of simulating the efforts of software project development.

The subject of the study is the regression models for simulating the efforts of software project development.

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 efforts of software project development is improved based on the normalizing transformation in the decimal logarithm form which allows to increase confidence of simulating the efforts of software project development 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: efforts, software project development, simulation, nonlinear regression model, software.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНИХ ПРОЕКТІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ЗА ОБРАНОЮ ТЕМОЮ	12
1.1 Аналіз існуючих моделей для моделювання трудомісткості розробки програмних проектів	12
1.2 Обґрунтування необхідності проведення досліджень за обраною темою	15
1.3 Висновки за розділом 1	16
2 НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНИХ ПРОЕКТІВ	18
2.1 Перевірка даних для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів на наявність викидів	18
2.2 Побудова нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів	21
2.3 Висновки за розділом 2	27
3 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНИХ ПРОЕКТІВ	29

3.1	Постановка задачі на розробку проекту програми для моделювання трудомісткості розробки програмних проектів	29
3.2	Розробка ескізного проекту програми для моделювання трудомісткості розробки програмних проектів	31
3.3	Розробка технічного проекту програми для моделювання трудомісткості розробки програмних проектів	38
3.4	Розробка робочого проекту програми для моделювання трудомісткості розробки програмних проектів	41
3.5	Висновки за розділом 3	50
4	РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМИ	52
4.1	Розрахунок витрат на створення й експлуатацію програми	52
4.2	Економічна ефективність розробки і впровадження програми	55
5	ОХОРОНА ПРАЦІ	58
5.1	Аналіз існуючої проблеми охорони праці	58
5.2	Аналіз шкідливих та небезпечних факторів у офісному приміщенні з персональними комп'ютерами	59
5.3	Розрахунок системи штучного освітлення у офісному приміщенні з персональними комп'ютерами	64
5.4	Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів	67
6	ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	70
6.1	Аналіз існуючої проблеми необхідності охорони навколишнього середовища	70
6.2	Забруднення навколишнього середовища підприємством з розробки та супроводження програмного забезпечення	73

6.3 Розробка заходів щодо зменшення забруднення	75
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТОК А Технічне завдання на розробку програми для моделювання трудомісткості розробки програмних проектів	87
ДОДАТОК Б Текст програми	91
ДОДАТОК В Опис програми	98
ДОДАТОК Г Інструкція користувача	99
ДОДАТОК Д Програма та методика випробувань	101

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

ПЗ – програмне забезпечення;

COCOMO – the COnstructive COst MOdel;

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. У наші дні розробка програмного забезпечення (ПЗ) залишається бути складним процесом, який не завжди є успішним. Так, за дослідженнями Standish Group, які виконувалися з 2016 по 2020 р., лише 31% програмних проектів були успішними, 50% завершилися лише з частковим успіхом, а 19% не були завершені взагалі [1], що у порівнянні з попередніми результатами, які були представлені у 2015 р. [2], практично не змінилися.

Однією із причин цього є не достовірне прогнозування трудомісткості розробки програмних проектів в певних випадках. Тому достовірне прогнозування трудомісткості розробки програмних проектів залишається важливою задачею.

На сьогоднішній день при прогнозуванні трудомісткості розробки програмних проектів часто використовуються нелінійні регресійні рівняння трудомісткості їх виконання в залежності від певних метрик ПЗ, наприклад, таких як кількість строк коду, функціональних точок, сторінок проектної документації, дефектів, тощо. Але зазначені рівняння, зокрема СОСОМО II [3] та інші [4-8], не містять у своєму складі випадкових складових, що не дозволяє здійснювати моделювання трудомісткості розробки програмних проектів як випадкової величини. А ті моделі, що містять у своєму складі випадкові складові, як правило мають не дуже високі показники прогнозування трудомісткості [9-13]. Тому все ще виникає потреба у побудові відповідних моделей для моделювання трудомісткості розробки програмних проектів саме як випадкової величини із врахуванням можливих викидів у даних.

Мета і завдання дослідження. Метою роботи є підвищення достовірності прогнозування трудомісткості розробки програмних проєктів.

Для досягнення поставленої мети потрібно вирішити такі завдання.

- Проаналізувати існуючі нелінійні регресійні моделі для моделювання трудомісткості розробки програмних проєктів;

- Здійснити підбір сукупності даних, за якими буде будуватися нелінійна регресійна модель для моделювання трудомісткості розробки програмних проєктів та перевірити ці дані на наявність двовимірних викидів;

- Удосконалити нелінійну регресійну модель для моделювання трудомісткості розробки програмних проєктів на основі нормалізуючого перетворення;

- На підставі запропонованої нелінійної регресійної моделі розробити програму для моделювання трудомісткості розробки програмних проєктів.

Об'єктом дослідження є процес моделювання трудомісткості розробки програмних проєктів.

Предметом дослідження є регресійні моделі для моделювання трудомісткості розробки програмних проєктів.

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

Наукова новизна одержаних результатів. Удосконалено нелінійну регресійну модель для моделювання трудомісткості розробки програмних 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 розробки програмних проєктів та підвищити його достовірність.

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

Особистий внесок здобувача. У публікації [14], що зроблена у співавторстві, автору належить нелінійна регресійна модель для моделювання трудомісткості розробки програмних 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 РОЗРОБКИ ПРОГРАМНИХ ПРОЄКТІВ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ЗА ОБРАНОЮ ТЕМОЮ

1.1 Аналіз існуючих моделей для моделювання трудомісткості розробки програмних проєктів

На сьогоднішній день при прогнозуванні трудомісткості розробки програмних проєктів часто використовуються нелінійні регресійні рівняння трудомісткості їх виконання в залежності від певних метрик ПЗ, наприклад, таких як кількість строк коду, функціональних точок, сторінок проектної документації, дефектів, тощо. Але зазначені рівняння, зокрема СОСОМО II [3] та інші [4-8], не містять у своєму складі випадкових складових, що не дозволяє здійснювати моделювання трудомісткості розробки програмних проєктів як випадкової величини. А ті моделі, що містять у своєму складі випадкові складові, як правило мають не дуже високі показники прогнозування трудомісткості [9-13]. Тому все ще виникає потреба у побудові відповідних моделей для моделювання трудомісткості розробки програмних проєктів саме як випадкової величини із врахуванням можливих викидів у даних.

В [4, 6, 7] було побудовано декілька нелінійних регресійних рівнянь для оцінювання трудомісткості розробки програмних проєктів в залежності від кількості функціональних точок із використанням різних нормалізуючих перетворень: перетворення у формі натурального логарифму [4], перетворення у формі десяткового логарифму [6] та перетворення Джонсона сім'ї S_B [6, 7]. Але

регресійні рівняння, на відміну від регресійних моделей не містять у своєму складі випадкові складові, що не дозволяє здійснювати моделювання трудомісткості розробки програмних проєктів як випадкової величини. В [13] було побудовано нелінійну регресійну модель для оцінювання трудомісткості розробки програмних проєктів, але значення оцінки середньоквадратичного відхилення випадкової складової, яка описує залишки, було наведено з помилкою. А це не дозволяє скористатися нею для здійснення моделювання трудомісткості розробки програмних проєктів саме як випадкової величини.

В [13] за перетворенням Джонсона для сімейства S_B було побудовано нелінійну регресійну модель для моделювання трудомісткості виконання програмних проєктів Y у годинах в залежності від кількості функціональних точок X_1 , яка має вигляд

$$Y = \hat{\phi}_Y + \hat{\lambda}_Y / \left[1 + e^{-(\hat{Z}_Y + \varepsilon - \hat{\gamma}_Y) / \hat{\eta}_Y} \right], \quad (1.1)$$

де $\hat{Z}_Y$ – оцінка середньої нормалізованої трудомісткості за лінійним регресійним рівнянням, $\hat{Z}_Y = \hat{b}_0 + \hat{b}_1 Z_1$; $Z_1 = \gamma_1 + \eta_1 \ln \frac{X_1 - \phi_1}{\phi_1 + \lambda_1 - X_1}$, $\phi_1 < X_1 < \phi_1 + \lambda_1$, $\hat{\gamma}_Y = 1,97416$, $\hat{\gamma}_1 = 3,61583$, $\hat{\eta}_Y = 0,860386$, $\hat{\eta}_1 = 0,925973$, $\hat{\phi}_Y = 17,4111$, $\hat{\phi}_1 = 118,627$, $\hat{\lambda}_Y = 2812,481$, $\hat{\lambda}_1 = 75516,537$, ε повинна бути випадковою величиною з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2$.

Для моделі (1.1) значення показників якості прогнозування R^2 , MMRE та PRED(0.25) дорівнюють відповідно 0.8086, 0.2862 та 0.5273. 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 [15, 16]. Прийнятні 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

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i. \quad (1.2)$$

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

Значення $PRED(0,25)$ обчислюється як

$$PRED(0,25) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1 & \text{if } MRE_i \leq 0,25 \\ 0 & \text{otherwise.} \end{cases} \quad (1.3)$$

Вибірковий коефіцієнт детермінації R^2 визначається як

$$R^2 = 1 - \frac{SS_E}{SS_T}, \quad (1.4)$$

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 [13] було побудовано нелінійну регресійну модель для оцінювання трудомісткості розробки програмних проєктів, але значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки, було наведено з помилкою. А це не дозволяє скористатися ним для здійснення моделювання трудомісткості розробки програмних проєктів саме як випадкової величини. Крім того значення показників MMRE та PRED(0.25) не дотягують до прийнятних значень. Тому виникає потреба в побудові відповідних моделей для моделювання трудомісткості розробки програмних проєктів.

Зазначимо, що побудову нелінійної регресійної моделі (1.1) для моделювання трудомісткості розробки програмних проєктів було здійснено за методом [17], що заснований на багатовимірних нормалізуючих перетвореннях та інтервалах прогнозування. Суть цього методу [17] є такою. На першому етапі початкові дані перевіряються на наявність багатовимірних 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 [19, 20]. На другому етапі будується нелінійна регресійна модель із використанням відповідного методу на основі нормалізуючих перетворень [18]. На третьому етапі визначаються границі інтервалу прогнозування нелінійної регресії для рівня значущості, що дорівнює 0,05, за відповідним методом [18]. 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.

Остаточно модель (1.1) було побудовано за 110 точками даних. Тобто було 35 викидів з первинного набору 145 точок даних з [4]. Але при побудові моделі (1.1) не враховувалася можливість появи викидів, що по'язані з похибкою ε [21]. Тому в подальшому ми будемо використовувати модифікований метод для побудови нелінійних регресійних моделей на основі багатовимірних нормалізуючих перетворень [22].

1.3 Висновки за розділом 1

У даному розділі наведено огляд існуючих математичних моделей для моделювання трудомісткості розробки програмних проектів, здійснено обґрунтування необхідності проведення досліджень за обраною темою.

1) Зараз для моделювання трудомісткості розробки програмних проєктів в першу чергу використовуються методи нелінійного регресійного аналізу.

2) Для моделювання трудомісткості розробки програмних проєктів існують нелінійні регресійні рівняння, які побудовані в залежності від кількості функціональних точок із використанням різних нормалізуючих перетворень: перетворення у формі натурального логарифму, перетворення у формі десяткового логарифму та перетворення Джонсона сім'ї S_B . Але регресійні рівняння, на відміну від регресійних моделей не містять у своєму складі випадкові складові, що не дозволяє здійснювати моделювання трудомісткості розробки програмних проєктів як випадкової величини. Також відома нелінійна регресійна модель для оцінювання трудомісткості розробки програмних проєктів, яка побудована на основі перетворення Джонсона сім'ї S_B , але значення оцінки середньоквадратичного відхилення випадкової складової, що описує залишки, було наведено для неї з помилкою. А це не дозволяє скористатися нею для здійснення моделювання трудомісткості розробки програмних проєктів саме як випадкової величини. Тому виникає потреба в побудові відповідних моделей для моделювання трудомісткості розробки програмних проєктів.

3) Побудову нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проєктів, краще здійснювати за відповідним модифікованим методом, що заснований на нормалізуючих перетвореннях та визначені викидів. Тому удосконалювати нелінійну регресійну модель для моделювання трудомісткості розробки програмних проєктів доцільно саме на основі зазначеного методу.

НЕЛІНІЙНА РЕГРЕСІЙНА МОДЕЛЬ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНИХ ПРОЕКТІВ

2.1 Перевірка даних для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проєктів, на наявність викидів

Для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проєктів використовується вибірка двовимірних даних за такими метриками: трудомісткість виконання програмного проєкту (у годинах) Y та кількість функціональних точок X_1 . Ці дані з 133 програмних проєктів [6] наведені на рис.2.1.

Ми перевірили двовимірні дані метрик з рис.2.1 на нормальність за допомогою тесту Мардіа (Mardia's test) [23], 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 відповідно з [20, 21]. При цьому, як рекомендується в [20, 21, 24], ми брали рівень значимості 0,005.

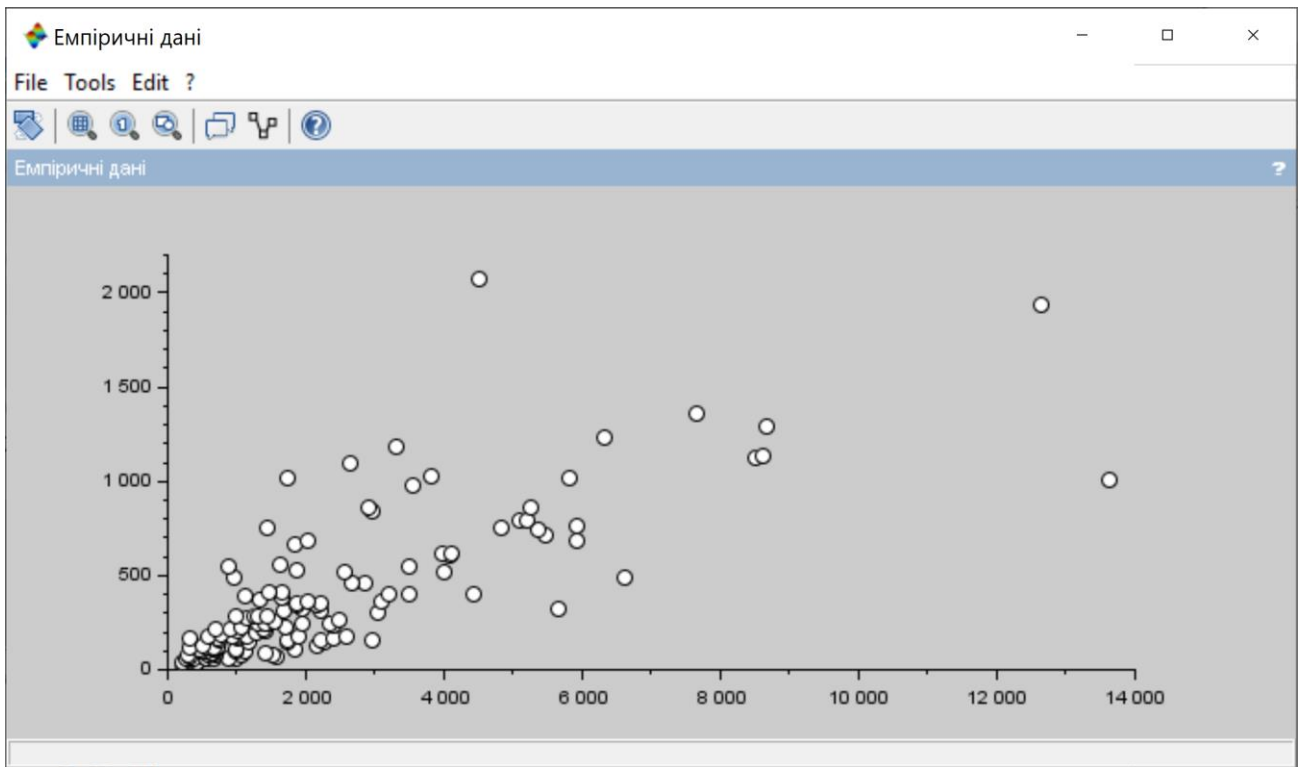


Рисунок 2.1 – Емпіричні дані з трудомісткості розробки 133 програмних проєктів в залежності від кількості функціональних точок

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 2.

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 [24], 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 рис.2.1 двовимірні викиди відсутні тому, що значення квадрата відстані Махаланобіса для всіх 133 точок даних менші за значення квантилю розподілу χ^2 , який дорівнює 10,60 для 2 ступенів свободи та рівня значимості 0,005. При цьому за тестом Мардіа багатовимірний розподіл нормалізованих даних є гаусівським. Зазначимо, що для даних з рис.2.1 без нормалізації є двовимірний викид ($X_1=4494$, $Y=2075,8$), для якого значення квадрата відстані Махаланобіса дорівнює 36,06 і є більшими за 10,60. Але цю точку даних ($X_1=4494$, $Y=2075,8$) не можна прийняти як двовимірний викид тому, що розподіл даних метрик з рис.2.1 не є гаусівським. А як відомо [24], 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 відсутні двовимірні викиди у даних з рис.2.1.

В подальшому 133 точок двовимірних негаусових даних, що наведені на рис.2.1, використовуються для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів за модифікованим методом [22].

2.2 Побудова нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів

Для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів, 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 на першому кроці двовимірні негаусівськи дані з рис.2.1 ми нормалізуємо за допомогою нормалізуючого перетворення, у якості якого ми використовуємо десятковий логарифм. Потім на другому кроці дані перевіряємо на наявність двовимірних викидів. Викиди відсутні тому, що значення квадрата відстані Махаланобіса для всіх 133 точок нормалізованих даних менші за значення квантилю розподілу χ^2 , який дорівнює 10,60 для 2 ступенів свободи та рівня значимості 0,005.

Далі згідно з [22], на третьому кроці для 133 точок нормалізованих даних будемо лінійну регресійну модель

$$Z_Y = \hat{Z}_Y + \varepsilon = \hat{b}_0 + \hat{b}_1 Z_1 + \varepsilon, \quad (2.4)$$

де $Z_Y = \lg Y$; $Z_1 = \lg X_1$; ε повинна бути випадковою величиною з розподілом Гаусу, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, з оцінкою дисперсії $\hat{\sigma}_\varepsilon^2$.

Оцінки параметрів моделі (2.4) для 133 точок нормалізованих даних є такими: $\hat{b}_0 = -0,39454$, $\hat{b}_1 = 0,87458$, а оцінка середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$ випадкової величини ε дорівнює 0,2325. Параметри моделі (2.4) оцінювалися за методом найменших квадратів.

Далі згідно з [22], на четвертому кроці перевіряємо гіпотезу про нормальність розподілу випадкової величини ε за критерієм Пірсона (критерієм

χ^2). За цим критерієм розподіл випадкової величини ε є гаусівським тому, що значення χ^2 , яке дорівнює 4,03, менше за квантиль розподілу χ^2 , що дорівнює 9,49 для 4 ступенів свободи та рівня значимості 0,05.

Після цього на п'ятому кроці за лінійною моделлю (2.4) та перетворенням у вигляді десяткового логарифму будуюмо нелінійну регресійну модель

$$Y = 10^{\varepsilon + \hat{b}_0} X_1^{\hat{b}_1}. \quad (2.5)$$

Далі згідно з [22], на шостому кроці ми визначаємо границі інтервалу прогнозування нелінійної регресії для рівня значущості, що дорівнює 0,05. Згідно з [25], у разі однофакторної нелінійної регресії інтервал прогнозування можна визначити як

$$\psi_Y^{-1} \left(\hat{Z}_{Y_i} \pm t_{\alpha/2, N-2} S_Z \sqrt{1 + \frac{1}{N} + \frac{(Z_{1_i} - \bar{Z}_1)^2}{S_{Z_1 Z_1}}} \right), \quad (2.6)$$

де ψ_Y^{-1} – зворотне до нормалізуючого перетворення ψ_Y для залежної змінної Y ;

$t_{\alpha/2, N-2}$ – квантиль t -розподілу Стюдента з $N-2$ ступенями свободи та $\alpha/2$

рівнем значущості; $S_Z^2 = \frac{1}{N-2} \sum_{i=1}^N (Z_i - \hat{Z}_i)^2$; $\bar{Z}_1 = \frac{1}{N} \sum_{i=1}^N Z_{1_i}$; $S_{Z_1 Z_1} = \sum_{i=1}^N (Z_{1_i} - \bar{Z}_1)^2$. у

нашому випадку $\psi_Y^{-1} = 10^{Z_Y}$.

В (2.6) знаки мінус та плюс відповідають нижній та верхній границям інтервалу прогнозування нелінійної регресії, що побудована на основі нормалізуючого перетворення.

Для моделі (2.5) з оцінками параметрів, що отримані для 133 точок даних з рис.2.1, виявилось, що значення Y для восьми точок даних виходять за визначені межі інтервалу прогнозування. Тому згідно з [22] ці вісім точок даних були видалені і на цьому перша ітерація була завершена. Ми почали виконувати другу ітерацію але вже для 125 точок даних, які зображені на рис.2.2.

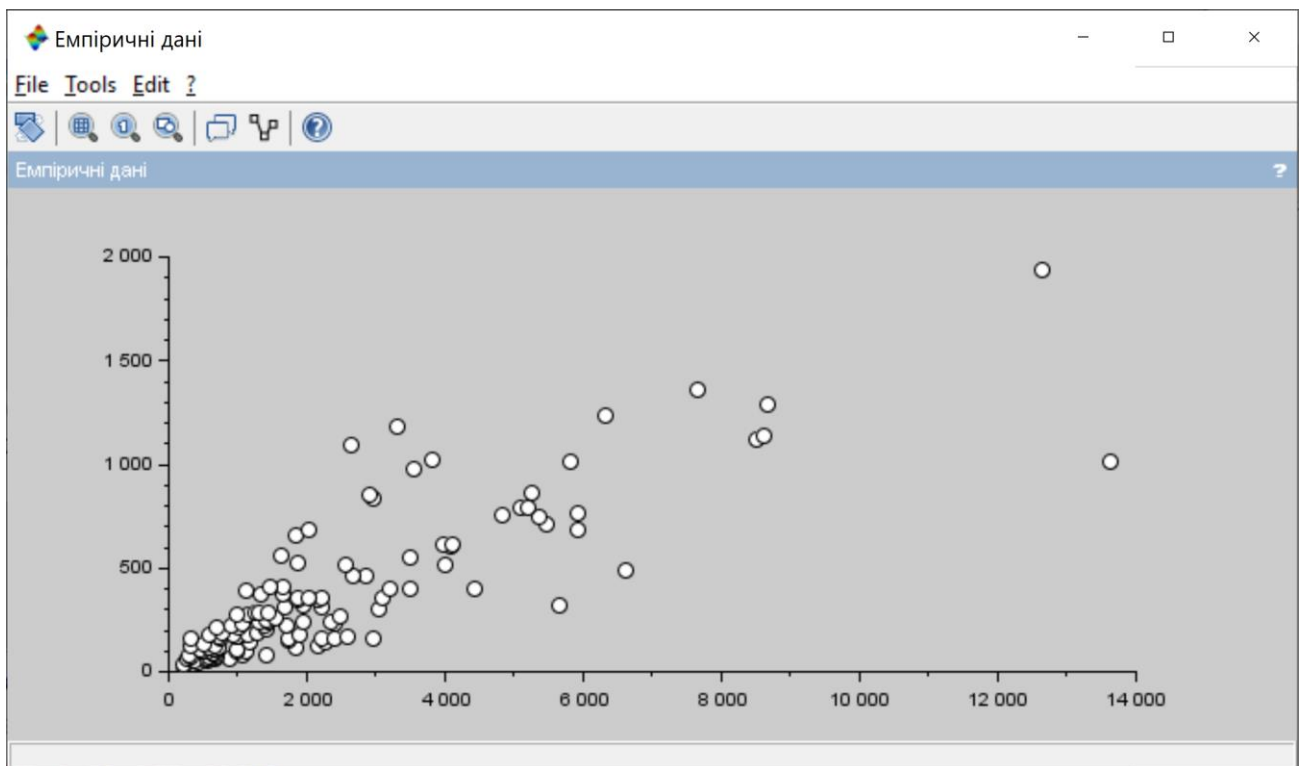


Рисунок 2.2 – Емпіричні дані з трудомісткості розробки 125 програмних проєктів в залежності від кількості функціональних точок

Всього було 17 ітерацій при побудові нелінійної регресійної моделі за модифікованим методом [22]. Остаточна нелінійна регресійна модель (2.5) була побудована за 95 точками даних, які наведені на рис.2.3.

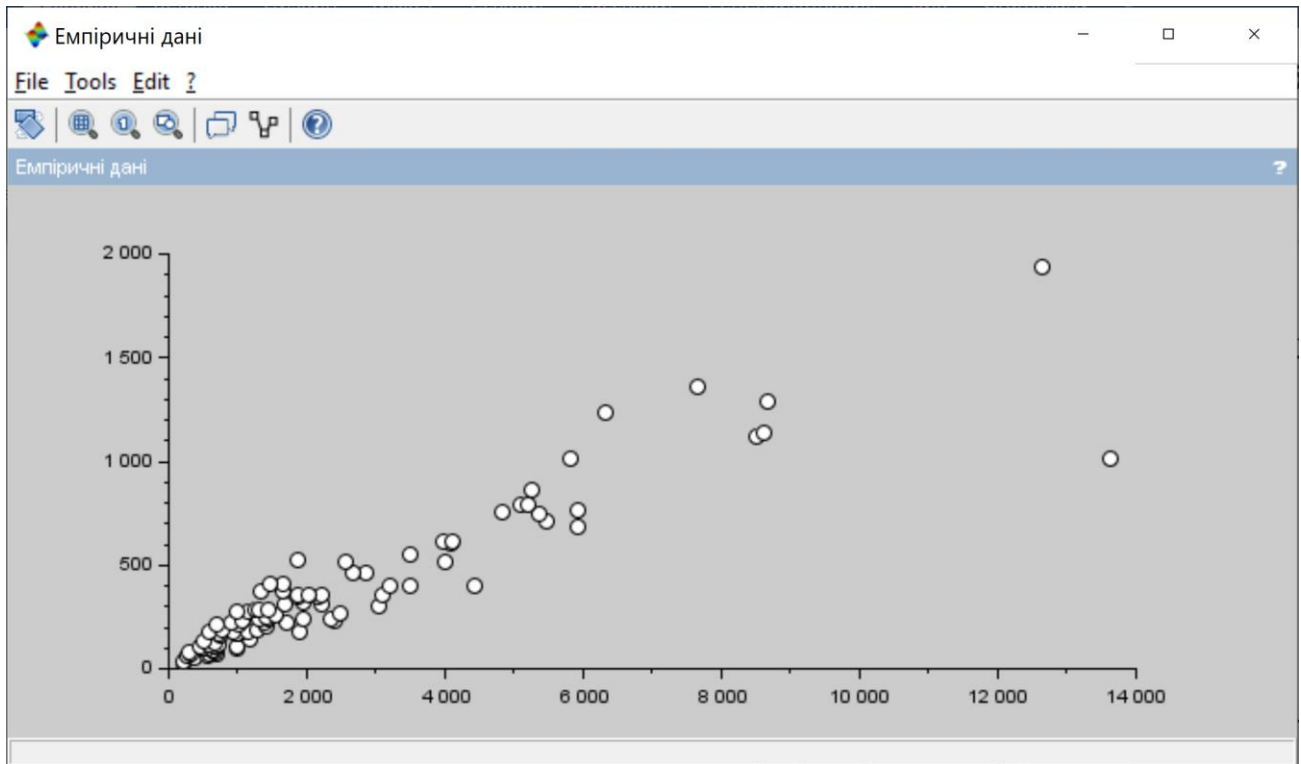


Рисунок 2.3 – Емпіричні дані з трудомісткості розробки 95 програмних проєктів в залежності від кількості функціональних точок

Остаточна нелінійна регресійна модель (2.5) має наступні оцінки параметрів: $\hat{b}_0 = -0,38926$, $\hat{b}_1 = 0,87634$, а оцінка середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$ випадкової величини ε дорівнює 0,1208.

Побудована нелінійна регресійна модель (2.5) 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 0,8745, 0,2298 та 0,6211. Це свідчить про задовільну якість отриманої моделі. Зазначимо, що показники R^2 , MMRE і PRED(0,25) обчислювалися за формулами (1.4), (1.2) та (1.3) відповідно.

Для моделювання трудомісткості розробки програмних проєктів за допомогою нелінійної регресійної моделі (2.5) ми використовуємо відповідний метод, що наведений в [26-28]. Для цього треба здійснювати моделювання гаусівської випадкової величини ε з нульовим математичним сподіванням та оцінкою середньо квадратичного відхилення $\hat{\sigma}_\varepsilon$, що дорівнює 0,1208. Для моделювання випадкової величини ε з розподілом Гауса у цій кваліфікаційній роботі використовується запропонований в [29, 30] 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 [30] значення гаусівської випадкової величини з математичним сподіванням нуль і дисперсією одиниця визначається за перетворенням Джонсона сім'ї S_U за формулою:

$$z = \eta \operatorname{Arsh}(x/\lambda), \quad (2.7)$$

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

$$x = tg(c_1 U + c_3 U^3), U \in [-1, 1]. \quad (2.8)$$

В (2.8) 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 (2.7) та (2.8) 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 [30]. Зазначимо, що за формулами (2.7) та (2.8) 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 0,1208. 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 розробки програмних

проектів на наявність викидів, побудовано нелінійну регресійну модель для моделювання трудомісткості розробки програмних проектів.

1) Перевірку даних для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів 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, двовимірні дані не містять викидів. Застосування критерія на основі квадрата відстані Махаланобіса до даних без їх нормалізації показало, що дані містять двовимірний викид. Але цю точку даних не можна прийняти як двовимірний викид тому, що розподіл даних не є гаусівським.

2) Удосконалено нелінійну регресійну модель для моделювання трудомісткості розробки програмних проектів 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 розробки програмних проектів.

3) Отримані результати свідчать про те, що для побудови нелінійних регресійних моделей для моделювання трудомісткості розробки програмних проектів потрібно використовувати нормалізуючі перетворення з урахуванням викидів у даних.

РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ МОДЕЛЮВАННЯ ТРУДОМІСТКОСТІ РОЗРОБКИ ПРОГРАМНИХ ПРОЕКТІВ

3.1 Постановка задачі на розробку проекту програми для моделювання трудомісткості розробки програмних проектів

Виходячи з аналізу існуючих моделей для моделювання трудомісткості розробки програмних проектів, сформулюємо постановку задачі. Потрібно створити програму для моделювання трудомісткості розробки програмних проектів, 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 розробки програмних проектів: трудомісткість розробки програмних проектів (у годинах) Y , кількість функціональних точок X_1 .

2) Вводити з інтерфейсу програми довірчу ймовірність та кількість функціональних точок за якою потрібно змоделювати трудомісткість розробки програмного проекту (у годинах).

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 та відносної похибки.

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 розробки програмного проєкту (у годинах), значення коефіцієнту детермінації R^2 та відносної похибки.

9) Виводити у текстовий файл результати моделювання трудомісткості розробки програмних 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 розробки програмних проектів

В ескізному проекті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для моделювання трудомісткості розробки програмних проектів та стани цієї програми при виконанні. Даний проект передбачає наявність лише одного актора – користувача програми, який буде повноцінно використовувати всі функції програми. Виявлені варіанти використання наведені в таблиці 3.1.

Таблиця 3.1 Виявлені варіанти використання

Основні актори	Найменування	Формулювання
1	2	3
Користувач	Ввести дані метрик програмних проектів	Вводить дані про трудомісткості (у годинах) розробки програмних проектів в залежності від їх розміру (у функціональних точках)
Користувач	Ввести розмір програми для моделювання трудомісткості її розробки	Вводить розмір програми у функціональних точках, в залежності від якої потрібно змоделювати трудомісткість розробки програмного проекту у годинах
Користувач	Ввести довірчу ймовірність	Вводить довірчу ймовірність, для якої потрібно оцінити довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості розробки програмного проекту у годинах

Продовження табл. 3.1.

1	2	3
Користувач	Нормалізувати дані	Виконує нормалізацію введених даних
Користувач	Побудувати нелінійну регресійну модель	Будує нелінійну регресійну модель для моделювання трудомісткості розробки програмних проєктів 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 розробки програмних проєктів зображена на рисунку 3.1.

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

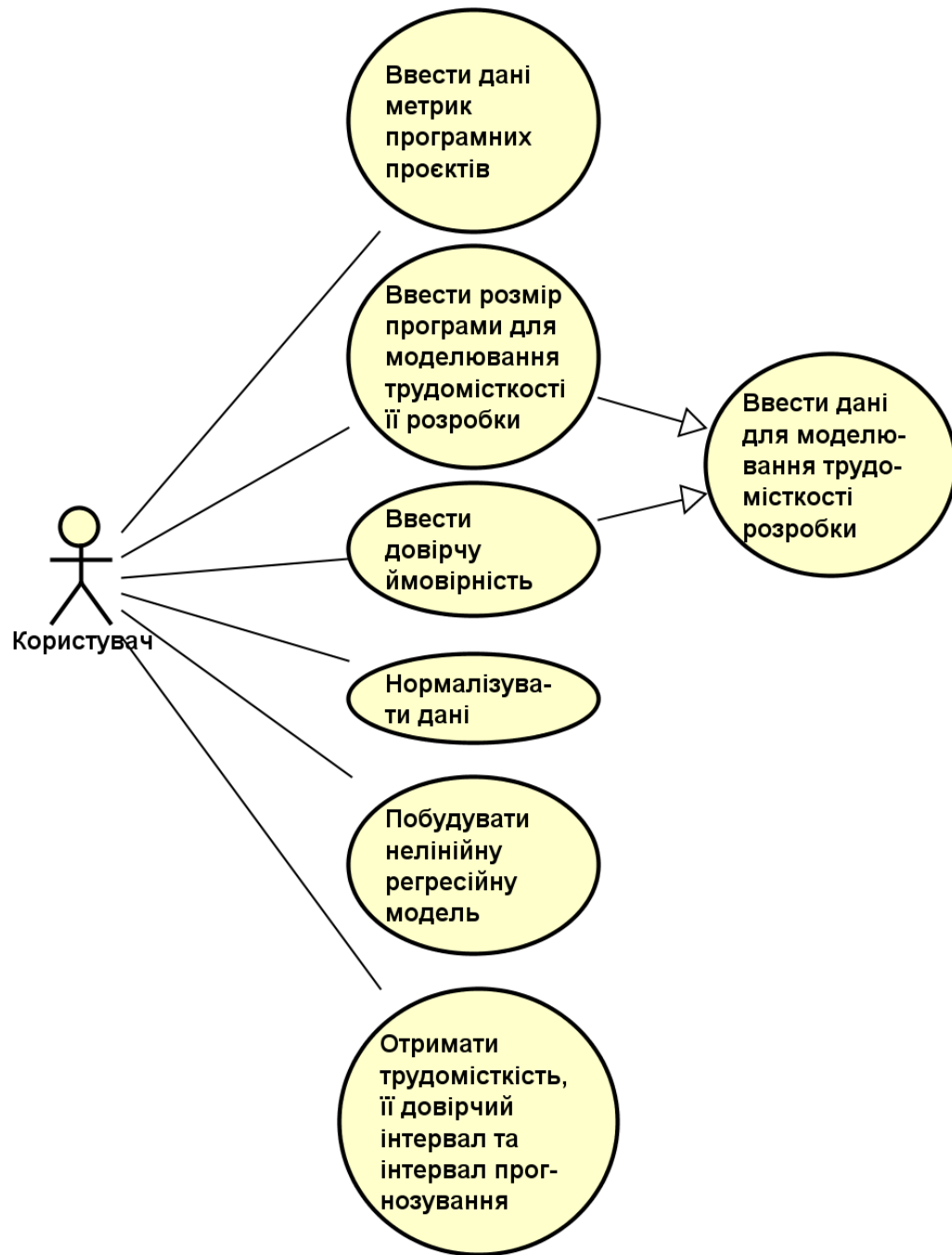


Рисунок 3.1 – Діаграма варіантів використання програми для моделювання трудомісткості розробки програмних проєктів

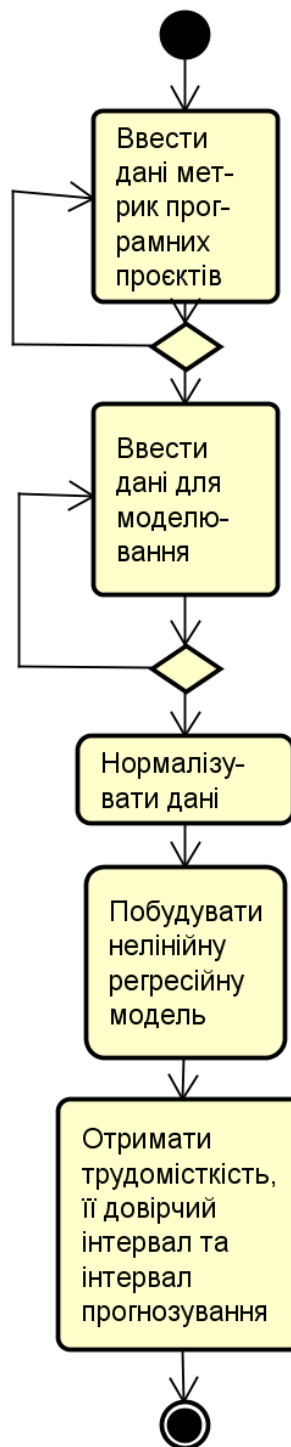


Рисунок 3.2 – Діаграма діяльності для моделювання
трудомісткості розробки програмних проєктів

Зовнішня специфікація програми.

Назва програми: SoftwareDevEffortSim.

Функції, які виконує програма:

1) Вводити з текстового файлу емпіричні дані для моделювання трудомісткості розробки програмних проєктів: трудомісткість розробки програмних проєктів (у годинах) Y , кількість функціональних точок X_1 .

2) Вводити з інтерфейсу програми довірчу ймовірність та кількість функціональних точок, за якою потрібно змодельовати трудомісткість розробки програмного проєкту.

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 та відносної похибки.

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 розробки програмних проєктів невеликого веб проєкту (у годинах), значення коефіцієнту детермінації R^2 та відносної похибки.

9) Виводити у текстовий файл результати моделювання трудомісткості розробки програмних проєктів у годинах.

Вхідні дані:

- емпіричні дані для моделювання трудомісткості розробки програмних проєктів: трудомісткість розробки програмних проєктів (у годинах) Y , кількість функціональних точок X_1 ;

- значення довірчої ймовірності, кількість функціональних точок, за якою потрібно змоделювати трудомісткість розробки програмного проєкту (у годинах).

Вихідні дані:

- значення трудомісткості, нелінійної регресії, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості розробки програмного проєкту (у годинах);

- значення коефіцієнту детермінації та відносної похибки.

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 розробки програмних проєктів (у годинах), вихід з програми (закриття всіх графічних вікон).

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

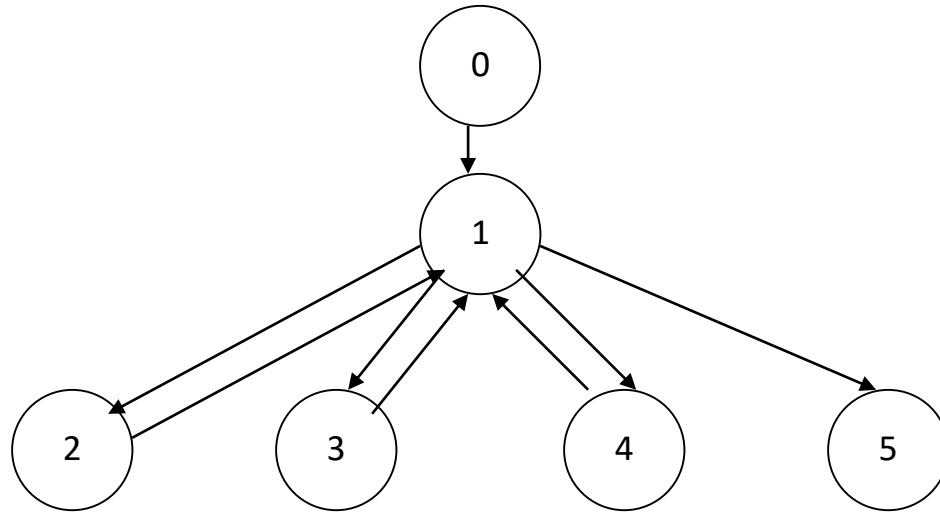


Рисунок 3.3 – Діаграма станів програми для моделювання трудомісткості розробки програмних проєктів

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 розробки програмних проєктів (у годинах), 5 – кінцевий стан.

Далі виконуємо розробку технічного проєкту програми для моделювання трудомісткості розробки програмних проєктів.

3.3 Розробка технічного проекту програми для моделювання трудомісткості розробки програмних проєктів

В технічному проєкті було спроектовано програму для оцінювання трудомісткості розробки програмних проєктів на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

Програма складається з таких модулів: головного модуля (SoftwareDevEffortSim), 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 розробки програмних проєктів (SoftwareDevEffortIntEst), модуля для закриття графічних вікон (W_Close).

Специфікація головного модуля.

Назва модуля: SoftwareDevEffortSim.

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 програмного проєкту та введення її значення у годинах.
- 5) Виклик функції для визначення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості розробки програмного проєкту (у годинах).
- 6) Виклик функції для закриття графічних вікон.

Вхідні дані:

- емпіричні дані для моделювання трудомісткості розробки програмних проєктів: трудомісткість розробки програмних проєктів (у годинах) Y , кількість функціональних точок X_1 ;

- значення довірчої ймовірності, кількість функціональних точок, за якою потрібно змодельовати трудомісткість розробки програмного проєкту (у годинах).

Вихідні дані:

- значення трудомісткості, нелінійної регресії, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості розробки програмного проєкту (у годинах);

- значення коефіцієнту детермінації та відносної похибки.

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 розробки програмного 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 розробки програмного проєкту (у годинах).

Назва модуля: SoftwareDevEffortIntEst.

Функції, які виконує модуль:

1) Визначає трудомісткість, нелінійну регресію, довірчий інтервал та інтервалу прогнозування нелінійної регресії трудомісткості розробки програмного проєкту (у годинах).

2) Обчислює значення коефіцієнту детермінації для нелінійної регресії та відносну похибку.

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 розробки програмного проєкту (у годинах);

- значення коефіцієнту детермінації для нелінійної регресії та відносної похибки.

Зовнішні ефекти: з'являються результати трудомісткості, нелінійної регресії, нижньої та верхньої границь довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості розробки програмних проєктів невеликого веб проєкту (у годинах) та виводиться значення коефіцієнту детермінації та відносної похибки 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 розробки програмних проєктів

В робочому проєкті було розроблено програму для моделювання трудомісткості розробки програмних 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 SoftwareDevEffortIntEst, що здійснює моделювання трудомісткості розробки програмних проектів, sci-мовою в системі моделювання Scilab наведено нижче.

```
//Функція визначення трудомісткості розробки програмних проектів
function SoftwareDevEffortIntEst(x, yx, n)
//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
xli=eval(get(edit_text4,'string'));
Pconf=eval(get(edit_text7,'string'));

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

Sx1x1=0;
SyA=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    SyA=SyA+(yx(i,1)-yA)*(yx(i,1)-yA);
end;
Sxx(1,1)=Sx1x1;

z1A=0;
for i=1:n
    z1A=z1A+log10(x(i,1));
end;
z1A=z1A/n;
Sz1z1=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
end;
Szz(1,1)=Sz1z1;

zi(1)=log10(xli)-z1A;
```

```

result=zi(1)/Sz1z1;

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

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

b0=-0.38926;
b1=0.87634;

Szyzyn4=0; Syy=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))-log10(yx(i,1)))^2;
    Syy=Syy+(yx(i,1)-10^b0*x(i,1)^b1)^2;
end;
Szyzyn4=Szyzyn4/(n-2)
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(xli);

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=0.1208*zd;

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

file_PC5=mopen('D:\Data_F\Effort95_data5.txt','w+');
//for i=1:3
    fprintf(file_PC5, "%5.6f ", yUBC);
//end
fclose(file_PC5);

```

```

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(radius_bikv_x,'value')==0)

else
    yfact=eval(get(edit_text12,'string'));
    MRE=abs((yfact-y_result)/yfact);
    textresult5=icontrol(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 sse-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.

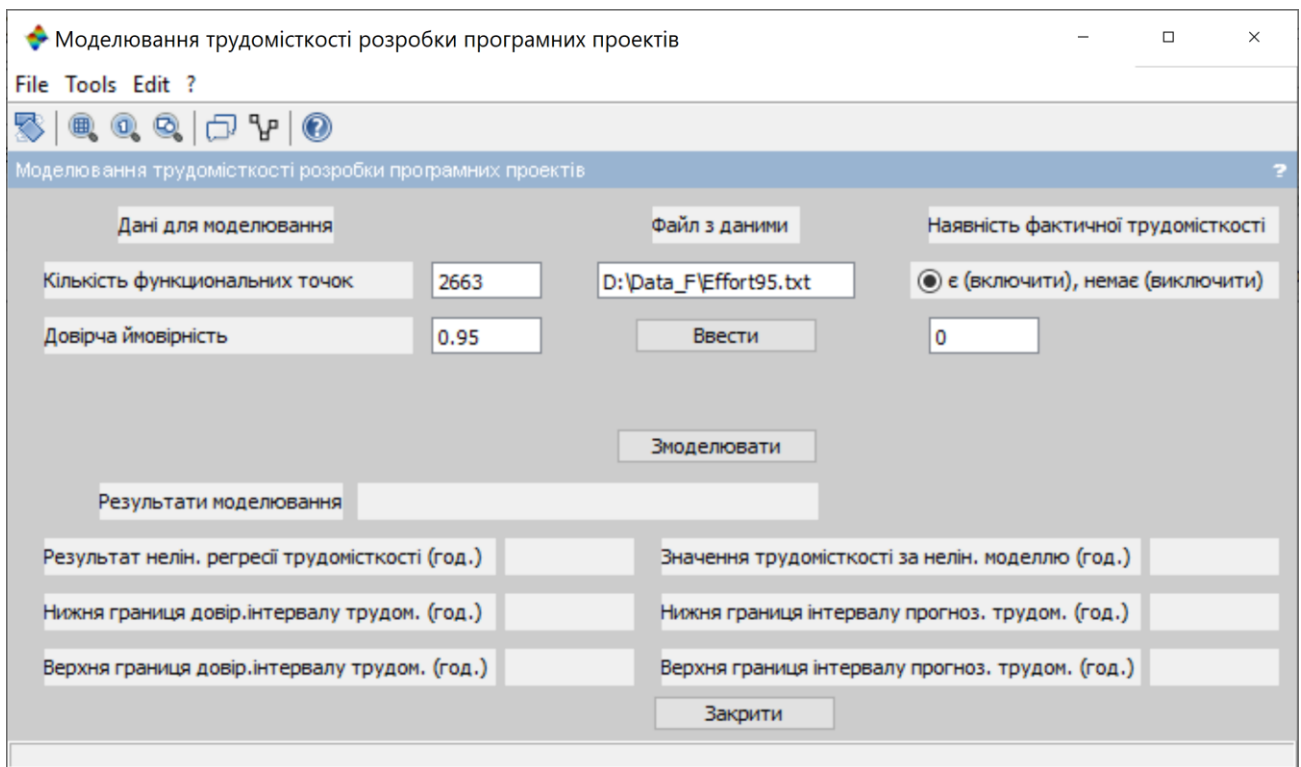


Рисунок 3.4. – Графічне вікно програми із інтерфейсними компонентами

Було проведено тестування окремих модулів. У якості прикладу наведемо результати тестування для модуля для моделювання трудомісткості розробки програмних проєктів для емпіричних даних – трудомісткості розробки програмних проєктів (у годинах) 95 програмних проєктів в залежності від кількості функціональних точок. Було побудовано нелінійну регресійну модель на основі нормалізуючого перетворення у формі десяткового логарифму. Результати візуалізації емпіричних даних з трудомісткості розробки програмних проєктів (у годинах) 95 програмних проєктів в залежності від кількості функціональних точок наведені на рис.3.5.

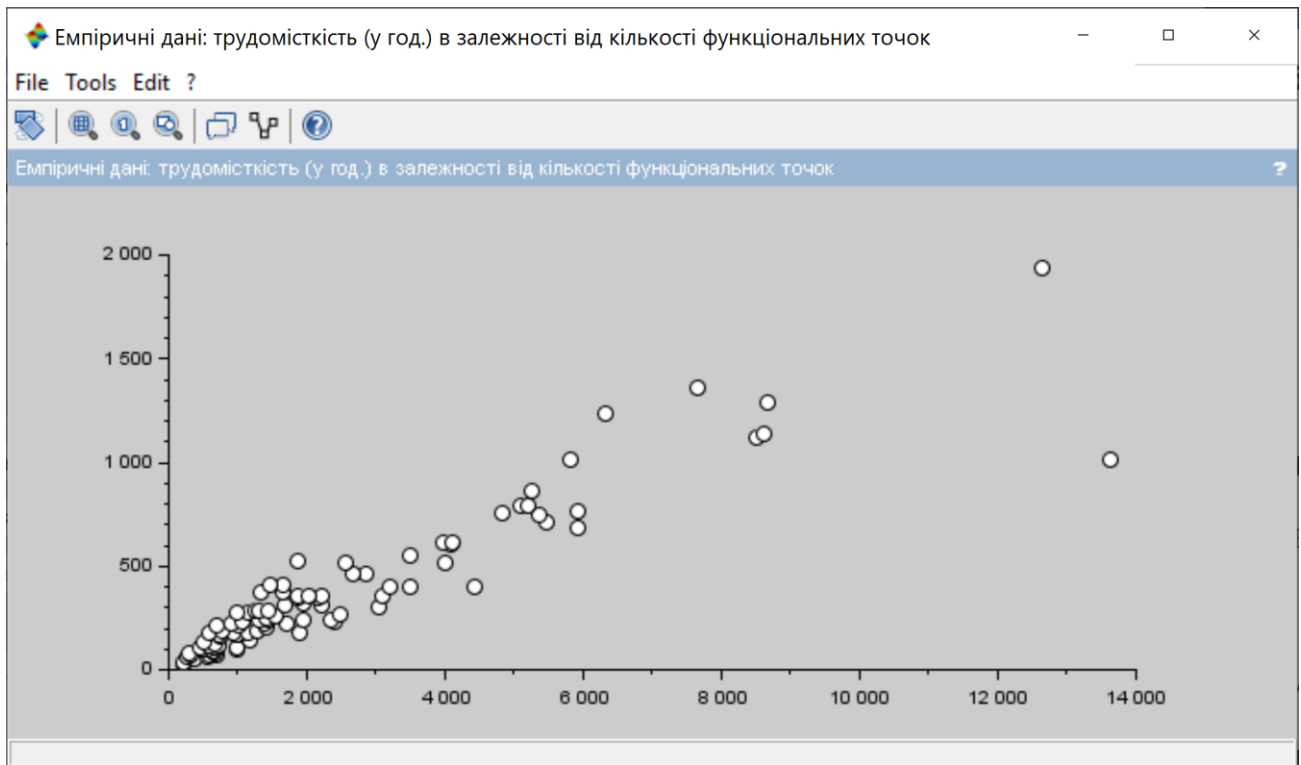


Рисунок 3.5. – Емпіричні дані з трудомісткості розробки програмних проєктів (у годинах) 95 програмних проєктів в залежності від кількості функціональних точок

Результати моделювання трудомісткості розробки програмних проєктів (у годинах), значень нелінійної регресії, нижньої і верхньої границь довірчого інтервалу нелінійної регресії трудомісткості розробки програмного проєкту та результати обчислення коефіцієнту детермінації R^2 наведені на рис.3.6. Також у разі відомого фактичної трудомісткості розробки програмного проєкту може бути обчислено значення відносної похибки (MRE) результату визначення

значень нелінійної регресії в порівнянні з фактичною трудомісткістю проекту для зазначеної кількості функціональних точок.

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості	
Кількість функціональних точок	2663	D:\Data_F\Effort95.txt	☐ є (включити), ☐ немає (виключити)	
Довірча ймовірність	0.95	Ввести	0	

Змодельювати

Результати моделювання			
Коефіцієнт детермінації $R^2=0.8745$			
Результат нелін. регресії трудомісткості (год.)	=409.7602	Значення трудомісткості за нелін. моделлю (год.)	=350.0656
Нижня границя довір.інтервалу трудом. (год.)	=375.7182	Нижня границя інтервалу прогноз. трудом. (год.)	=233.6023
Верхня границя довір.інтервалу трудом. (год.)	=446.8865	Верхня границя інтервалу прогноз. трудом. (год.)	=718.7575

Закрити

Рисунок 3.6. – Результати визначення трудомісткості, нелінійної регресії, довірчого інтервалу та інтервалу прогнозування нелінійної регресії трудомісткості розробки програмного проєкту (у годинах)

На рис.3.7 наведено значення відносної похибки (MRE) результату визначення значень нелінійної регресії в порівнянні з фактичною трудомісткістю розробки програмного проєкту для зазначеної кількості функціональних точок,

що дорівнює 2663. Як можна бачити, відносна похибка склала 0,1169 або 11,69%.

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості	
Кількість функціональних точок	2663	D:\Data_F\Effort95.txt	☒ є (включити), немає (виключити)	
Довірча ймовірність	0.95	Ввести	464	
Змоделювати				
Результати моделювання		Коефіцієнт детермінації $R^2=0.8745$		Відносна похибка $MRE=0.1169$
Результат нелін. регресії трудомісткості (год.)	=409.7602	Значення трудомісткості за нелін. моделлю (год.)	=445.1488	
Нижня границя довір.інтервалу трудом. (год.)	=375.7182	Нижня границя інтервалу прогноз. трудом. (год.)	=233.6023	
Верхня границя довір.інтервалу трудом. (год.)	=446.8865	Верхня границя інтервалу прогноз. трудом. (год.)	=718.7575	
Закрити				

Рисунок 3.7. – Результати визначення відносної похибки оцінки трудомісткості в порівнянні з фактичною трудомісткістю розробки програмного проєкту для 2663 функціональних точок

Зверніть увагу, що результати моделювання трудомісткості розробки програмного проєкту для 2663 функціональних точок за нелінійною регресійною моделлю (2.5) на рис. 3.6 та 3.7 наведені у полі «Значення трудомісткості за нелін. моделлю (год.)». Вони змінюються як значення випадкової величини Y у

моделі (2.5). Після повторного натискання на кнопку «Змоделювати» у полі «Значення трудомісткості за нелін. моделлю (год.)» значення трудомісткості знову змінюється як значення випадкової величини Y у моделі (2.5). Ця зміна показана на рис.3.8.

Моделювання трудомісткості розробки програмних проектів

File Tools Edit ?

Моделювання трудомісткості розробки програмних проектів

Дані для моделювання

Кількість функціональних точок 2663

Довірча ймовірність 0.95

Файл з даними

D:\Data_F\Effort95.txt

Ввести

Наявність фактичної трудомісткості

☒ є (включити), немає (виключити)

464

Змоделювати

Результати моделювання

Коефіцієнт детермінації $R^2=0.8745$

Відносна похибка $MRE=0.1169$

Результат нелін. регресії трудомісткості (год.) =409.7602

Значення трудомісткості за нелін. моделлю (год.) =365.0855

Нижня границя довір.інтервалу трудом. (год.) =375.7182

Нижня границя інтервалу прогноз. трудом. (год.) =233.6023

Верхня границя довір.інтервалу трудом. (год.) =446.8865

Верхня границя інтервалу прогноз. трудом. (год.) =718.7575

Закрити

Рисунок 3.8. – Зміна значення трудомісткості як значення випадкової величини Y у моделі (2.5) для 2663 функціональних точок після повторного натискання на кнопку «Змоделювати»

Також зверніть увагу, що результати моделювання трудомісткості розробки програмного проекту для 2663 функціональних точок за нелінійною

регресійною моделлю (2.5) на рис.3.6-3.8, що наведені у полі «Значення трудомісткості за нелін. моделлю (год.)», не виходять за границі інтервалу прогнозування.

Отримані результати практично співпадають з результатами інших авторів [6, 13]. 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 розробки програмних проектів, розроблено ескізний, технічний та робочий проекту програми для моделювання трудомісткості розробки програмних проектів.

1) В ескізному проекті було виявлено варіанти використання та побудовано їх сценарії, розроблено зовнішню специфікацію програми для моделювання трудомісткості розробки програмних проектів та стани цієї програми при виконанні.

2) В технічному проекті було спроектовано програму для моделювання трудомісткості розробки програмних проектів на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

3) В робочому проекті було розроблено програму для моделювання трудомісткості розробки програмних проектів, 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:

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

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

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

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

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

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\text{Фм} = 264,5 * \text{Тз}, \quad (4.2)$$

де Тз – 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:

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

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 електробезпечності.

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 штучного освітлення у офісному приміщенні з персональними комп'ютерами

У приміщенні необхідно організувати змішане освітлення, тобто сполучення природного і штучного освітлення. Природне освітлення створюється бічним освітленням через вікно. Штучне освітлення використовується при недостатньому природному освітленні. У даному приміщенні використовується загальне штучне освітлення.

Розрахунок його здійснюється по методу світлового потоку з урахуванням потоку, відбитого від стін і стелі.

Потрібно зробити розрахунок загального освітлення для офісного приміщення. Довжина $A=4,5$ м, ширина $B=3,0$ м, висота $H=2,5$ м. Висота робочої поверхні $h_p = 1,0$ м. Для освітлення використовується світильник УПМ-15. Мінімальна освітленість лампи накаливання по нормах $E_{min}=75$ лк.

Коефіцієнт відображення стелі $S_p=70\%$, стін $S_c=50\%$, робочої поверхні $S_r=10\%$.

Напруга мережі 220 В.

Відстань від стелі до робочої поверхні:

$$H_o = H - h_p = 2,5 - 1 = 1,5 \text{ м.}$$

Відстань від стелі до світильника:

$$h_c = 0,2 * H_o = 0,2 * 1,5 = 0,3 \text{ м.}$$

Висота підвісу світильника над освітленою поверхнею:

$$h = H_o - h_c = 1,5 - 0,3 = 1,2 \text{ м.}$$

Висота підвісу світильника над підлогою:

$$H_n = h + h_p = 1,2 + 1 = 2,2 \text{ м.}$$

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

Відстань між центрами світильників:

$$L = 1,5 * h = 1,5 * 1,2 = 1,8 \text{ м.}$$

Необхідна кількість світильників:

$$N = S/L^2 = (4,5 * 3) / (1,8)^2 = 4,2 \approx 4.$$

Приймаємо кількість світильників: 4.

Індекс приміщення:

$$i = (A * B) / (h * (A + B)) = (4,5 * 3) / (1,2 * (4,5 + 3)) = 1,5.$$

Коефіцієнт використання світлового потоку $\eta = 0,51$.

Світловий потік однієї лампи:

$$\begin{aligned} \Phi &= (E_{\min} * S * K_3 * Z) / (N * \eta) = \\ &= (75 * 13,5 * 1,3 * 1,15) / (4 * 0,51) = 741 \text{ лм.} \end{aligned}$$

За знайденим світловим потоком з таблиці “Параметри ламп розжарювання загального призначення з розрахунковими напругами 130 и 220 В” обираємо лампу Б215-225-75 потужністю 75 Вт, що має світловий потік 950 лм, найбільш близький до розрахункового.

Фактична освітленість:

$$E_f = E_{\min} * \Phi_{\text{л}} / \Phi_{\text{р}} = 75 * 950 / 741 = 96,15 \text{ лк.}$$

Загальна потужність:

$$P_{\text{общ}} = P_{\text{л}} * N = 75 * 4 = 300 \text{ Вт} = 0,3 \text{ кВт.}$$

Таким чином, розрахунок загального освітлення для офісного приміщення можна вважати виконаним, а його результати такими, що задовольняють загальноприйнятим нормам промислового освітлення.

5.4 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів

Нормована освітленість робочого столу з ПК повинна бути в межах 300-500 лк. Допускається установка світильників місцевого освітлення для підсвічування документів. При цьому місцеве освітлення так само, як і загальне, не повинно створювати відблисків на екрані відеомоніторів. Зовнішня освітленість екрана монітора не повинна перевищувати 300 лк.

Бажано, щоб природне світло падало ліворуч, хоча допускається і таке розміщення робочого місця, коли віконні прорізи розташовані праворуч. Вікна повинні бути обладнані пристроями типу жалюзі, зовнішніх козирків і інше.

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

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

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

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

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

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

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

До робочого столу теж пред'являються певні вимоги. Його конструкція повинна забезпечувати оптимальне розміщення на його площі використовуваного устаткування. Робочий стілець користувача необхідно розташувати з таким розрахунком, щоб він дозволяв змінювати позу працюючого, а також був підйомно-поворотним. Це допоможе уникнути стомлення м'язів шийно-плечової області і спини. Оптимальна відстань від очей до екрана монітора – 600-700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків і символів. Недотримання цього параметра може

призвести до швидкого розвитку стомлення і перевтоми зорового аналізатора і зі збільшенням тривалості роботи сприяти розвитку короткозорості. Відстань між робочими столами з відеотерміналами повинне бути не менш 2,0 м, а відстань між бічними поверхнями відеомоніторів – не менш 1,2 м. Робочі місця з ПК у залах електронно-обчислювальних машин чи у приміщеннях із джерелами шкідливих виробничих факторів розміщаються в ізольованих кабінах з організованим повітрообміном.

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 Забруднення навколишнього середовища підприємством з розробки та супроводження програмного забезпечення

Хоча підприємством з розробки та супроводження програмного забезпечення не займається промисловим виробництвом з викидом шкідливих речовин у атмосферу, воду та ґрунт, це не виключає існування джерел забруднення довкілля. До таких джерел, насамперед, належать:

- дизельна електростанція;
- твердопаливні котли;
- побутове сміття;
- електромагнітне випромінювання.

Одночасно в офісі компанії можуть працювати до 100 комп'ютерів, стаціонарних телефонів та велика кількість серверної техніки, яка обслуговує роботу цього обладнання, аварійне чи планове відключення електроенергії призводить до значних перешкод у роботі персоналу. Для того, щоб уникнути таких ситуацій, на території компанії знаходиться дизельна електростанція. Вона працює як в аварійному, так і в резервному режимі.

Результатом роботи дизельної електростанції є продукти згоряння дизельного палива та шумове забруднення.

Твердопаливні котли використовуються в холодну пору року в системі опалення офісних приміщень. Для опалення використовуються деревні пелети. Хоча таке джерело тепла має менший відсоток шкідливих викидів в атмосферу, ніж, наприклад, вугілля чи мазут, цей вид опалення не можна вважати екологічно чистим. Порівняльні характеристики продуктів згоряння палива наведені в табл. 6.1.

Таблиця 6.1

Рівні викидів забруднюючих речовин в атмосферу при спалюванні різних видів палива

Вид палива	Викиди забруднюючих речовин в атмосферне повітря без систем очищення, тонн на 1 тис. тонн нат. палива				
	CO ₂	NO ₂	SO ₂	Тверді частинки (пил неорг.)	РАЗОМ
Природний газ	1,18	3,52	0,00	0,00	4,70
Древні брикети, пелети	4,68	9,31	0,28	4,11	17,69
Деревина дров'яна	4,9	9,4	0,3	4,3	18,9
Тирса деревна	5,0	9,6	0,5	5,0	20,0
Древні відходи, обрізки	5,2	9,9	0,4	5,2	20,7
Швидкозростаюча деревина	4,8	9,5	0,0	8,4	22,7
Тріска, сучки, кора	5,6	11,4	0,8	13,4	31,3
Мазут	5,20	5,20	35,30	0,30	45,90
Брикет торф'яний	8,04	26,81	3,00	13,02	50,87
Кам'яне вугілля	9,58	63,56	9,20	65,32	147,66

Побутові відходи – це залишки речовин і предметів, які утворюються в результаті побутової та господарської діяльності людини і які не можуть бути

використані на місці утворення, а їх накопичення і зберігання порушують санітарний стан навколишнього середовища.

Всі побутові відходи, які утворюються в процесі функціонування компанії, можна розділити на рідкі та тверді. До рідких побутових відходів належать нечистоти з вигребів туалетів, помий (від миття посуду, підлоги, тощо) і стічні води (побутові, злизові). До твердих побутових відходів можна віднести сміття (побутові відходи) та кухонні відходи. Зокрема, твердими відходами є папір, картон, скло, пластмаса, продукти харчування.

Генератором електромагнітного забруднення є, в першу чергу, велика кількість комп'ютерної техніки на території офісу. Більшість із них працює 8 годин на добу, а деякі не вимикаються цілодобово. Випромінювачами в даному випадку є і процесор, і монітор. Випромінювання останнього значно вищі, особливо його бічні і задні стінки, адже вони не мають спеціального захисного покриття, як у лицьовій частині монітора. Крім того, в офісі є дві кухні, кожна з яких обладнана 5 мікрохвильовими печами.

Електромагнітне випромінювання має несприятливий вплив на організм людини. Його наслідками можуть бути головний біль, порушення сну, перевтома і, навіть, у деяких випадках стенокардія.

6.3 Розробка заходів щодо зменшення забруднення

Для того, щоб зменшити рівень забруднення навколишнього середовища, варто вжити певних заходів, які мінімізують цей вплив, якщо його неможливо усунути цілком. Серед таких заходів:

- 1) Очищення димових газів у мокрих золоуловлювачах.

Електрофільтри на електростанціях застосовуються для досягнення найбільш глибокого очищення димових газів в основному на великих енергоблоках потужністю 300 МВт та більше. Мокрі золоуловлювачі, які працюють при маленьких питомих витратах води та невеликих перепадах тиску, встановлюються за котлоагрегатами середньої паропродуктивності. У мокрих золоуловлювачах уловлювання часток золи на плівці води, яка стікає по його стінках, здійснюється за рахунок відцентрової сили, яка діє на частки. Ефективність апарата не перевищує 90 %.

2) Використання природного газу для опалення офісних приміщень замість деревних пелетів.

Згідно з показниками, наведеними в табл. 6.1, рівень шкідливих викидів від згоряння природного газу є найнижчим серед перерахованих видів палива. Проте зміну джерела енергії слід проводити, враховуючи економічну ефективність, оскільки вартість цих видів палива не є однаковою.

3) Раціональне позбавлення від побутових відходів.

Деякі побутові відходи можна безпечно спалювати для отримання енергії. Вторинну сировину (макулатуру, скло, пластмаси) можна відправляти на переробку, а не вивозити на сміттєзвалища. Крім того, зменшити кількість побутових відходів допоможе повторне їх використання (наприклад, скляних чи пластикових пляшок та контейнерів для їжі), скорочення споживання та використання меншої кількості упаковки.

4) Зменшити рівень електромагнітного забруднення та захиститися від його надмірного впливу.

Зрозуміло, що неможливо повністю обійтися без електроприладів та комп'ютерів, проте можна дещо зменшити їх негативний вплив. Для цього слід

вимикати з електромережі комп'ютери, телефони та обладнання, з якими ніхто не працює, виходити на прогулянки і робити перерви в роботі з комп'ютером.

ВИСНОВКИ

В кваліфікаційній роботі вирішене наукове завдання з підвищення достовірності моделювання трудомісткості розробки програмних проектів за допомогою нелінійної регресійної моделі.

1) Для моделювання трудомісткості розробки програмних проектів існують нелінійні регресійні моделі, які побудовані в основному в залежності від кількості функціональних точок. Але у зазначених моделях не були наведені значення оцінок середньоквадратичного відхилення випадкової складової, яка описує залишки. А це не дозволяє здійснювати моделювання трудомісткості розробки програмних проектів. Тому виникає потреба в побудові відповідних моделей для моделювання трудомісткості розробки програмних проектів.

2) Перевірку даних для побудови нелінійної регресійної моделі для моделювання трудомісткості розробки програмних проектів 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, двовимірні дані не містять двовимірних викидів. Застосування критерія на основі квадрата відстані Махаланобіса до даних без їх нормалізації показало, що дані містять двовимірний викид. Але цю точку даних не можна прийняти як двовимірний викид тому, що розподіл даних не є гаусівським.

3) Удосконалено нелінійну регресійну модель для моделювання трудомісткості розробки програмних проектів 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 розробки програмних проектів.

4) На підставі запропонованої нелінійної регресійної моделі розроблено програму для моделювання трудомісткості розробки програмних проектів 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 розробки програмних проектів для системи моделювання Scilab 6.0.0.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Portman, H. (2021, January 6). Review Standish Group – CHAOS 2020: Beyond Infinity. <https://henryportman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/>
2. Wojewoda, S., & Hastie, S. (2015, October 4). Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch. Standish Group. <https://www.infoq.com/articles/standish-chaos-2015/>
3. Boehm B.W. Software Cost Estimation With Cocomo II [Text] / Barry W. Boehm, Chris Abts, A. Winsor Brown, Sun Chulani et. al. – Prentice Hal, Hardcover, 2000. – 544 p. – ISBN-10: 0137025769
4. Kitchenham B. An empirical study of maintenance and development estimation accuracy / B. Kitchenham,, S. L. Pfleeger, B. McColl, S. Eagan // The Journal of Systems and Software. – 2002. – 1(64). – P. 57-77. – DOI: [https://doi.org/10.1016/S0164-1212\(02\)00021-3](https://doi.org/10.1016/S0164-1212(02)00021-3)
5. Lazić L. Challenges in estimating software testing effort / L. Lazić, I. Đokić, S. Milinković // Proceedings of INFOTEH-JAHORINA. – 2014. – Vol. 13. – P. 637-642.
6. Prykhodko S.B. Constructing the non-linear regression equations on the basis of bivariate normalizing transformations / S. B. Prykhodko, N. V. Prykhodko, L. M. Makarova, O. O. Kudin, T. G. Smykodub // Вісник Херсонського національного технічного університету. – 2017. – № 3 (62). – С.333-337. – Т.1. – ISSN 2078-4481

7. Приходько С.Б. Побудова нелінійних регресійних рівнянь для оцінювання трудомісткості виконання програмних проєктів на основі нормалізуючих перетворень / С. Б. Приходько, Н. В. Приходько, Т. Г. Смикодуб // Прикладні науково-технічні дослідження: матеріали міжнар. наук.-прак. конф., 5-7 квіт. 2017 р. – Івано-Франківськ: Симфонія форте, 2017. – С.18. – ISBN 978-966-284-110-7

8. Приходько С.Б. Трьохфакторне нелінійне регресійне рівняння для оцінювання трудомісткості розробки мобільних застосунків у фазі планування / С. Б. Приходько, Н. В. Приходько, К. О. Книрик // Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки» Том 30 (69) № 5, 2019. – С. 154-160. – DOI: <https://doi.org/10.32838/2663-5941/2019.5-1/25>

9. Літвінов А.В. Розробка нелінійної регресійної моделі для оцінювання трудомісткості виконання програмних проєктів на основі перетворення Джонсона / А. В. Літвінов, С. Б. Приходько // Інформатика, інформаційні системи та технології: тези доповідей чотирнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 14 квітня 2017 р. – Одеса, 2017. – С.59-61.

10. Приходько С.Б. Комп'ютерне моделювання трудомісткості виконання програмних проєктів за нелінійною регресійною моделлю / С. Б. Приходько, Т. Г. Смикодуб, І. А. Комаровський, А. В. Літвінов // Проблеми моделювання та розроблення інформаційних систем: матеріали II науково-практичної інтернет-конференції (Дрогобич, 20 травня 2017 року). – Дрогобич: ДДПУ ім. І. Франка, 2017. – С.69-73.

11. Приходько С.Б. Трьохфакторна нелінійна регресійна модель для оцінювання трудомісткості розробки мобільних застосунків / С. Б. Приходько, Н. В. Приходько, К. О. Книрик // Моделювання та інформаційні технології : зб.

наук. пр. / Нац. акад. наук України, Ін-т пробл. моделювання в енергетиці ім. Г. Є. Пухова. – Київ: [б. в.], 2019. – Вип. 86. – С.12-19. ISSN 2309-7647 (print) DOI: <http://doi.org/10.5281/zenodo.3610632>

12. Приходько С.Б. Удосконалення трьохфакторної нелінійної регресійної моделі для оцінювання трудомісткості розробки мобільних застосунків / С. Б. Приходько, Н. В. Приходько, К. О. Книрик // Проблеми інформаційних технологій. – 2019. – № 2 (026). – С. 21-31. – ISSN 1998-7005 (Print), ISSN 2313-0687 (Online) DOI: <https://doi.org/10.35546/2313-0687.2019.26.21-31>

13. Куриленко О.С. Удосконалення нелінійної регресійної моделі для моделювання трудомісткості виконання програмних проєктів / О. С. Куриленко, С. Б. Приходько // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2020): Матеріали Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2020 р.). – Миколаїв: НУК імені адмірала Макарова, 2020. – С. 14-15.

14. Приходько С.Б. Математичне моделювання трудомісткості розробки програмних проєктів за нелінійною регресійною моделлю / С. Б. Приходько, А. В. Пухалевич, **О. О. Машкін** // 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. 16-17. Режим доступу: <http://itconf.nuos.edu.ua/2022/wp-content/uploads/sites/5/NUOS-ITMAS-2022-Proceedings.pdf>

15. 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>

16. 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.

17. 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

18. 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>

19. 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>

20. 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>

21. 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>

22. 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

23. 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>

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

25. Prykhodko S.B. Statistical anomaly detection techniques based on normalizing transformations for non-Gaussian data / S. B. Prykhodko // Computational Intelligence (Results, Problems and Perspectives): Proceedings of the International Conference, May 12-15, 2015, Kyiv-Cherkasy, Ukraine / Ministry of Education and Science of Ukraine, Taras Shevchenko National University of Kyiv

and [etc]; Vitaliy Ye. Snytyuk (Editor). – Cherkasy: editor July Chabanenko, 2015. – p.286-287. – ISBN 978-966-493-975-8

26. 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.

27. Prykhodko S.B. Modeling a dependent variable of non-linear regression on the basis of normalizing transformations / S. B. Prykhodko, N. V. Prykhodko, K. S. Pugachenko // Контроль і управління в складних системах (КУСС-2016). XIII Міжнародна конференція. Тези доповідей. Вінниця, 3-6 жовтня 2016 року. – Вінниця: ВНТУ. ПП «ТД «Едельвейс», 2016. – С.13-15.

28. Prykhodko S.B. Simulation of dependent variables of non-linear regressions in Scilab / S. B. Prykhodko, N. V. Prykhodko, K. S. Pugachenko // Матеріали VIII-ої науково-практичної конференції «Free and Open Source Software», Харків, 22-24 листопада 2016 р. – Харків: Харківський національний університет будівництва та архітектури, 2016. – С.12.

29. 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/>

30. 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.

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

32. 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 розробки програмних проектів

Вступ

Назва розробки – “Програма для моделювання трудомісткості розробки програмних проектів”, далі програма.

Підстави для розробки

Завдання на кваліфікаційну роботу.

Призначення розробки:

Програма призначена для моделювання трудомісткості розробки програмних проектів 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 розробки програмних проектів: трудомісткість розробки програмних проектів (у годинах) Y , кількість функціональних точок X_1 .

2) Вводити з інтерфейсу програми довірчу ймовірність, кількість функціональних точок за якою потрібно змоделювати трудомісткість розробки програмного проекту (у годинах) та кількість функціональних точок X_1 .

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 та відносної похибки.

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 розробки програмних проєктів невеликого веб проєкту (у годинах), значення коефіцієнту детермінації R^2 та відносної похибки.

9) Виводити у текстовий файл результати моделювання трудомісткості розробки програмних проєктів у годинах.

Вимоги до організації вхідних та вихідних даних

Вхідні дані:

- емпіричні дані для моделювання трудомісткості розробки програмних проєктів: трудомісткість розробки програмних проєктів (у годинах) Y , кількість функціональних точок X_1 ;

- значення довірчої ймовірності, значення кількість функціональних точок, за якою потрібно змоделювати трудомісткість розробки програмного проєкту (у годинах).

Вихідні дані:

- значення трудомісткості, нелінійної регресії, довірчий інтервал та інтервал прогнозування нелінійної регресії трудомісткості розробки програмного проєкту (у годинах);

- значення коефіцієнту детермінації та відносної похибки.

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 розробки програмних проєктів
f=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f,'position',[0,0,700,300])
//Устанавливаем заголовок окна.
set(f,'figure_name',...
'Моделювання трудомісткості розробки програмних проєктів');

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\Effort95.txt','position',[320,240,140,20]);

BtDatInp=uicontrol('style','pushbutton','string','Ввести',...
'CallBack',[x,yx,f1,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 функціональних точок','position',...
[20,240,200,20]);
edit_text4=uicontrol(f,'style','edit','string',...
'2663','position',[230,240,60,20]);
lab_text7=uicontrol(f,'style','text','string',...
'Довірча ймовірність','position',...
[20,210,200,20]);
edit_text7=uicontrol(f,'style','edit','string',...
'0.95','position',[230,210,60,20]);
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_text10=uicontrol(f,'style','text','string',...
'Нижня границя довір.інтервалу трудом. (год.)','position',...
[20,60,245,20]);
lab_text11=uicontrol(f,'style','text','string',...
'Верхня границя довір.інтервалу трудом. (год.)','position',...
[20,30,245,20]);
lab_text12=uicontrol(f,'style','text','string',...

```

```

'Нижня границя інтервалу прогноз. трудом. (год.)','position',...
[355,60,260,20]);
lab_text13=uicontrol(f,'style','text','string',...
'Верхня границя інтервалу прогноз. трудом. (год.)','position',...
[355,30,260,20]);
lab_text14=uicontrol(f,'style','text','string',...
'Значення трудомісткості за нелін. моделлю (год.)','position',...
[355,90,260,20]);
//
//Текстовое поле, определяющее вывод результатов.
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]);

//Флажок, отвечающий за выбор типа преобразования.
radio_bikv_x=uicontrol('style','radiobutton','string',...
'є (включити), немає (виключити)','value',1,'position',...
[490,240,200,20]);
//
if (get(radio_bikv_x,'value')==0)

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

Btnr_pred_int=uicontrol('style','pushbutton','string','Змоделювати','Call
Back',...
'SoftwareDevEffortIntEst(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, n]=DataInput()
dir_name=get(edit_text,'string');

file_PC=mopen(dir_name,'r');
file_PC2=mopen('D:\Data_F\Effort95_data2.txt','w+');

```

```

file_PC3=fopen('D:\Data_F\Effort95_data3.txt','w+');
i=1;
//n=52;
while (feof(file_PC))<1
    y(i)=fscanf(file_PC,'%f'); x1(i)=fscanf(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);
    x(i,1)=x1(i); x(i,2)=x1(i);
end;
fprintf(file_PC2,'%f\t%f\n',y,x1);

for i = 1:n
    for j = 1:1 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 функціональних точок');
plot2d(x1,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);
end;
Sxx(1,1)=Sx1x1;

z1A=0;

```

```

for i=1:n
    z1A=z1A+log10(x(i,1));
end;
z1A=z1A/n;
Sz1z1=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
end;
Szz(1,1)=Sz1z1;

zi(1)=log10(x(1,1))-z1A;

result=zi(1)/Sz1z1;

Df=n-2;
P=0.975;
Q=1-P;

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

b0=-0.38926;
b1=0.87634;

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

zy_result=b0+b1*log10(x(1,1));

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

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

file_PC5=mopen('D:\Data_F\Effort95_data5.txt','w+');

    fprintf(file_PC5, "%5.6f ", yUBC);

fclose(file_PC5);

endfunction

//Функція визначення трудомісткості
function SoftwareDevEffortIntEst(x, yx, n)
//Считываем значение переменных из текстовых полей и

```

```

//преобразовываем их к числовому типу.
x1i=eval(get(edit_text4,'string'));
Pconf=eval(get(edit_text7,'string'));

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

Sx1x1=0;
SyA=0;
for i=1:n
    Sx1x1=Sx1x1+(x(i,1)-x1A)*(x(i,1)-x1A);
    SyA=SyA+(yx(i,1)-yA)*(yx(i,1)-yA);
end;
Sxx(1,1)=Sx1x1;

z1A=0;
for i=1:n
    z1A=z1A+log10(x(i,1));
end;
z1A=z1A/n;
Sz1z1=0;
for i=1:n
    Sz1z1=Sz1z1+(log10(x(i,1))-z1A)*(log10(x(i,1))-z1A);
end;
Szz(1,1)=Sz1z1;

//Szz_1=1/Szz;

zi(1)=log10(x1i)-z1A;

result=zi(1)/Sz1z1;

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

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

b0=-0.38926;
b1=0.87634;

```

```

Szyzyn4=0; Syy=0;
for i=1:n
    Szyzyn4=Szyzyn4+...
        (b0+b1*log10(x(i,1))-log10(yx(i,1)))^2;
    Syy=Syy+(yx(i,1)-10^b0*x(i,1)^b1)^2;
end;
Szyzyn4=Szyzyn4/(n-2)
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(xli);

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=0.1208*zd;

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

file_PC5=mopen('D:\Data_F\Effort95_data5.txt','w+');
//for i=1:3
    fprintf(file_PC5, "%5.6f ", yUBC);
//end
fclose(file_PC5);

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

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

```

ДОДАТОК В – Опис програми

Загальні відомості

Програма для моделювання трудомісткості розробки програмних проєктів, для системи моделювання Scilab 6.0.0. Файл програми SoftwareDevEffortSim.sce має розмір 14 КБ.

Функціональне призначення

Програма SoftwareDevEffortSim призначений для моделювання трудомісткості розробки програмних 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 SoftwareDevEffortSim.sce стандартними засобами ОС.

ДОДАТОК Г – Інструкція користувача

Загальний опис роботи з програмою

Після завантаження файлу SoftwareDevEffortSim.sce на екрані з'являється форма для відображення початкового стану програми для моделювання трудомісткості розробки програмних проектів (рисунк Г.1).

Моделювання трудомісткості розробки програмних проектів

File Tools Edit ?

Моделювання трудомісткості розробки програмних проектів ?

Дані для моделювання

Кількість функціональних точок: 2663

Довірча ймовірність: 0.95

Файл з даними

D:\Data_F\Effort95.txt

Ввести

Наявність фактичної трудомісткості

☒ є (включити), немає (виключити)

0

Змодельовати

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 розробки програмних проектів» (рис.Г.1)

потрібно відредагувати ім'я текстового файлу з початковими емпіричними даними, що будуть використовуватися для моделювання трудомісткості розробки програмних проектів.

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

Потім у графічному вікні «Моделювання трудомісткості розробки програмних проектів» (рис. Г.1) у відповідних вікнах редагування потрібно відредагувати кількість функціональних точок, довірчу ймовірність.

Крім того у графічному вікні «Моделювання трудомісткості розробки програмних проектів» (рис. Г.1) потрібно за допомогою радіо кнопки здійснити вибір наявності фактичної трудомісткості розробки програмного проекту та ввести його для подальшого порівняння з відповідною оцінкою трудомісткості.

Далі потрібно натиснути на командну кнопку «Змоделювати». Після цього з'являться результати моделювання трудомісткості розробки програмного проекту, метрики для якого були введені.

І на останок у графічному вікні «Моделювання трудомісткості розробки програмних проектів» 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 розробки програмних проектів SoftwareDevEffortSim.

Ціль випробувань: перевірка відповідності програми SoftwareDevEffortSim 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 SoftwareDevEffortSim. Далі виконується перевірка роботи програми. Виконується перевірка всіх екранних форм програми, а також всіх допустимих для виконання операцій всередині екранних форм згідно до інструкції 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 розробки програмних проєктів

File Tools Edit ?

Моделювання трудомісткості розробки програмних проєктів

Дані для моделювання		Файл з даними	Наявність фактичної трудомісткості
Кількість функціональних точок	2663	D:\Data_F\Effort95.txt	☒ є (включити), немає (виключити)
Довірча ймовірність	0.95	Ввести	0
Змодельовати			
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

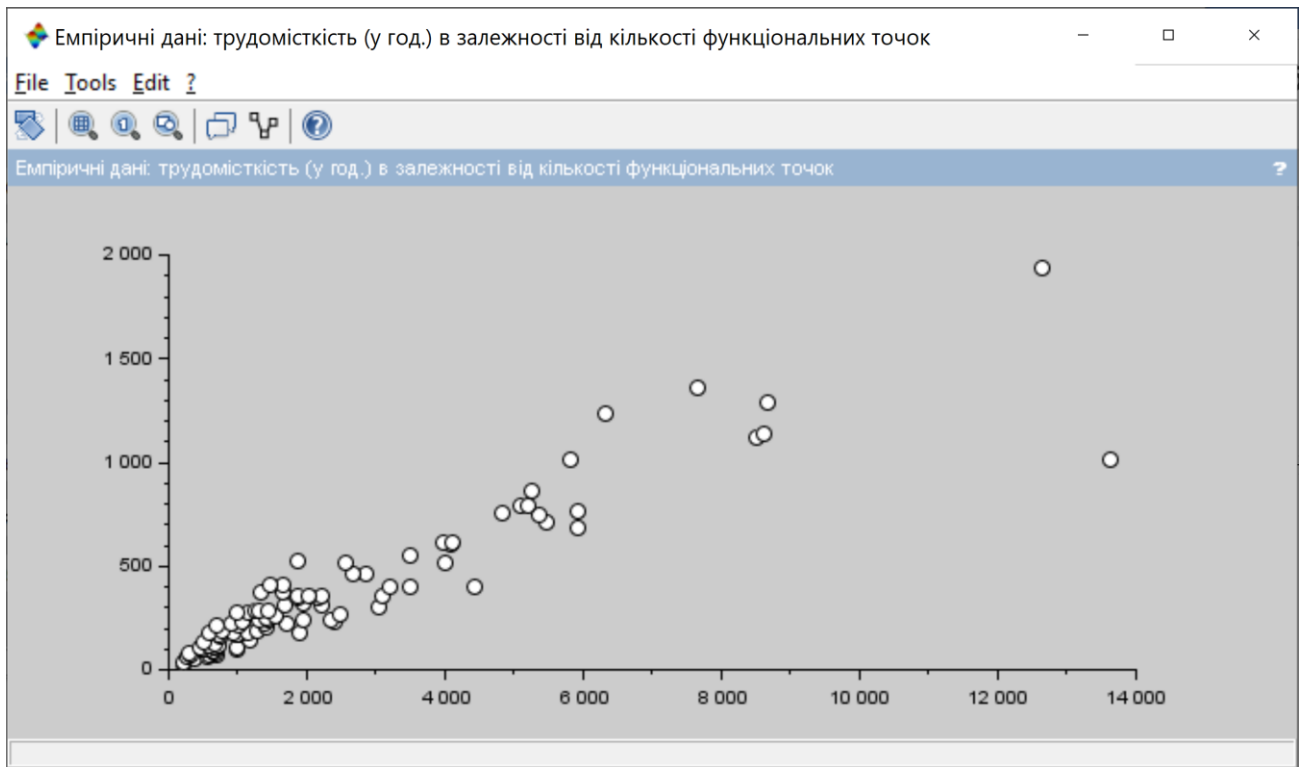


Рисунок Д.2 – Візуалізація тестових емпіричних даних.

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

Моделювання трудомісткості розробки програмних проєктів

File Tools Edit ?

Моделювання трудомісткості розробки програмних проєктів

Дані для моделювання

Кількість функціональних точок: 2663

Довірча ймовірність: 0.95

Файл з даними: D:\Data_F\Effort95.txt

Ввести

Наявність фактичної трудомісткості: ☐ є (включити), немає (виключити)

0

Змоделювати

Результати моделювання

Коефіцієнт детермінації $R^2=0.8745$

Результат нелін. регресії трудомісткості (год.)	=409.7602	Значення трудомісткості за нелін. моделлю (год.)	=487.1492
Нижня границя довір.інтервалу трудом. (год.)	=375.7182	Нижня границя інтервалу прогноз. трудом. (год.)	=233.6023
Верхня границя довір.інтервалу трудом. (год.)	=446.8865	Верхня границя інтервалу прогноз. трудом. (год.)	=718.7575

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 розробки програмного проєкту, наведені на рисунку Д.4.

Моделювання трудомісткості розробки програмних проєктів

File Tools Edit ?

Моделювання трудомісткості розробки програмних проєктів

Дані для моделювання

Кількість функціональних точок: 2663

Довірча ймовірність: 0.95

Файл з даними: D:\Data_F\Effort95.txt

Ввести

Наявність фактичної трудомісткості: ☒ є (включити), немає (виключити)

464

Змоделювати

Результати моделювання

Коефіцієнт детермінації $R^2=0.8745$

Відносна похибка MRE=0.1169

Результат нелін. регресії трудомісткості (год.)	=409.7602	Значення трудомісткості за нелін. моделлю (год.)	=337.7301
Нижня границя довір.інтервалу трудом. (год.)	=375.7182	Нижня границя інтервалу прогноз. трудом. (год.)	=233.6023
Верхня границя довір.інтервалу трудом. (год.)	=446.8865	Верхня границя інтервалу прогноз. трудом. (год.)	=718.7575

Закрити

Рисунок Д.4 – Результати роботи програми після натискання на командну кнопку “Змоделювати» у разі, коли радіо кнопка включена та введена фактична трудомісткість – 464 год.

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

```

30 220
35 219
56 380
60 557
61.06 254
69 322
73.92 698
74.4 585
81.48 302

```

84.28 669
86.45 652
88.2 630
100.8 570
101.65 485
102 980
103.4 989
105.28 464
108.64 654
113.52 583
115.14 718
121.55 676
130 502
144.72 1173
157.56 750
165.36 718
169.85 1003
172 995
177 1884
177.63 1143
178.1 593
179 928
182.4 763
191 1280
203.01 1425
210.08 1016
210.45 693
219.88 1388
219.88 896
225.54 1698
229.71 1066
230 2408
240 1940
240.84 1428
241.86 2356
243.86 1304
248.88 1413
258.24 1557
267 2480
274.92 1159
279 988
285.12 1252
287.28 1318

287.64 1431
303.52 3050
311.85 1693
313.95 2223
320.12 1939
343.54 1868
346.8 2172
351.9 2203
356.31 1860
357 3088
359.64 2037
370 1328
377 1660
401 4414
401.98 3483
403 3189
409.4 1650
410 1455
458.38 2860
464 2663
515.07 4009
518.4 2570
525.96 1877
550.14 3504
603.58 4068
616.08 3966
616.32 4115
681 5927
711.9 5456
746.24 5359
756.25 4824
761.08 5927
789.66 5094
791.84 5192
862 5249
1010.88 13635
1015.98 5816
1121.48 8509
1136.34 8600
1234.2 6307
1292.56 8656
1362.11 7649
1940.68 12646