

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

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

«Допущений до захисту»
Завідувач кафедри

«__» _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

**на тему: *Математична модель оцінювання трудомісткості тестування
Yii1-проектів та розробка програми для її реалізації***

Виконав:
студент 6151м групи Геленко Ю.В.
(підпис, ПІБ)

Керівник роботи:
доцент, к.т.н. Пономаренко Т.В.
(посада, науковий ступень вчене звання)

(підпис, ПІБ)

Миколаїв – 2021 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

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

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

проф. С.Б.Приходько

(підпис)

«___» _____ 2021 р.

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

Студенту Геленко Юрію Васильовичу

1. Тема роботи: Математична модель оцінювання трудомісткості тестування Yii1-проектів та розробка програми для її реалізації

Керівник роботи: Пономаренко Тетяна Вікторівна

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

2. Термін подання роботи: 06.12.2021 р. _____

3. Вихідні дані по роботі: завдання на розробку видане керівником Пономаренко Т.В.

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 1. Актуальність теми. 2 Мета і завдання дослідження. 3 Об'єкт і предмет дослідження. 4 Методи дослідження. 5 Наукова новизна одержаних результатів. 6 Особистий внесок здобувача. Апробація результатів досліджень. 7 Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. 8 Діаграма варіантів використання ПЗ для оцінювання трудомісткості тестування Yii1-проектів. 9 Діаграма діяльності ПЗ для оцінювання трудомісткості тестування Yii1-проектів 10 ER-діаграма структури даних ПЗ для оцінювання трудомісткості тестування Yii1-проектів 11 Інтерфейс користувача 12 Висновки _____

6. Консультанти розділів роботи

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

7. Дата видачі завдання 13.10.2021 р.**КАЛЕНДАРНИЙ ПЛАН**

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	18.10.2021	НС*
2.	Підготовка розділу з огляду літератури та обґрунтування необхідності проведення досліджень за обраною темою	19.10.2021	НС*
3.	Підготовка розділу (ів) з результатів власних досліджень	22.10.2021	НС*
4.	Підготовка розділу з проекту програмного забезпечення	16.11.2021	
5.	Підготовка організаційно-економічного розділу	19.11.2021	
6.	Підготовка розділу з охорони праці	22.11.2021	
7.	Підготовка розділу з охорони навколишнього середовища	24.11.2021	
8.	Підготовка розділу Висновки	25.11.2021	
9.	Оформлення списку використаних джерел та додатків	29.11.2021	
10.	Подання на кафедрі ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версії роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	06.12.2021	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
11.	Підготовка презентації та доповіді	09.12.2021	
12.	Попередній захист роботи на засіданні кафедри ПЗАС	10.12.2021	
13.	Подання на кафедрі ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	20.12.2021	

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

Студент

(підпис)

(ПІБ)

Керівник роботи

(підпис)

(ПІБ)

РЕФЕРАТ

Геленко Юрій Васильович

«Математична модель оцінювання трудомісткості тестування Yii1-проектів та розробка програми для її реалізації»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 121 – «Інженерія програмного забезпечення».

Національний університет кораблебудування імені адмірала Макарова.

Миколаїв,

2021 р.

Обсяг роботи: 80 стор., 12 табл., 9 рис., 33 використаних джерела, 5 додатків.

Актуальність теми роботи: визначається необхідністю підвищення достовірності оцінювання трудомісткості тестування Yii1-проектів, тому що кількість проектів Yii1- є значною, спадкоємність версій відсутня, а розвиток проекту завжди зв'язаний з підтримкою якості шляхом тестування, яке, в свою чергу, потребує оцінювання.

Мета і завдання дослідження. Метою даної магістерської роботи є підвищення достовірності оцінювання трудомісткості тестування Yii1-проектів. Для досягнення поставленої мети необхідно вирішити такі завдання:

1. проаналізувати існуючі моделі та методи оцінювання трудомісткості тестування ПЗ, сформулювати постановку задачі;
2. розробити регресійну модель оцінювання трудомісткості тестування Yii1-проектів;
3. перевірити адекватність моделі для оцінювання трудомісткості тестування Yii1-проектів;
4. розробити ПЗ для оцінювання трудомісткості тестування, використовуючи побудовану нелінійну регресійну модель.

Об'єктом дослідження є процес оцінювання трудомісткості тестування Yii1-проектів.

Предметом дослідження є багатofакторна регресійна модель оцінювання трудомісткості тестування Yii1-проектів.

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

Наукова новизна отриманих результатів. Удосконалено нелінійну регресійну модель на основі нормалізації даних за допомогою десяткового

логарифму, що дозволяє підвищити достовірність оцінювання трудомісткості тестування Yiii1-проектів.

Практичне значення отриманих результатів полягає в розробці програмного забезпечення для оцінювання трудомісткості тестування проектів на основі нелінійної багатфакторної регресійної моделі та десяткового логарифмічного нормалізуючого перетворення.

Особистий внесок здобувача. Проведені дослідження, зібрані емпіричні дані та отримані результати, наведені в кваліфікаційній роботі, одержані здобувачем особисто.

Апробація результатів роботи. Результати досліджень, викладених у роботі, були оприлюднені на II Всеукраїнській науково-практичній інтернет конференції “Інформаційні технології: моделі, алгоритми, системи” (ITMAS – 2021) 26-28 жовтня 2021 р.

Публікації. За результатами досліджень було опубліковано тези конференції.

Ключові слова: ОЦІНКА ТРУДОМІСТКОСТІ ТЕСТУВАННЯ, БАГАТОФАКТОРНА РЕГРЕСІЙНА МОДЕЛЬ, ВЕБ ЗАСТОСУНКИ, YIII1, PNR.

ABSTRACT

Gelenko Yuriy

"A mathematical model for testing effort estimation of Yii1 projects and
developing the software for its implementation"

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

Admiral Makarov National University of Shipbuilding. Mykolayiv,
2021

The work contains: 80 pages, 12 tables, 9 figures, 33 references, 5 appendices.

Relevance of the topic is determined by the need to increase the reliability of estimating the size of applications created using the Anaconda platform, because the number of Yii1- projects is significant and there is no succession of versions. So project development is always related to maintaining quality through testing, which in turn requires an evaluation.

The purpose and objectives of the study. The purpose of this master's thesis is to increase the reliability of estimating the complexity of testing Yii1-projects. To achieve this goal it is necessary to solve the following tasks:

- to analyze the existing models and methods of estimating the complexity of software testing, to formulate the problem statement;
- develop a regression model for estimating the complexity of testing Yii1-projects depending on the number of test cases, defects that need to be checked, and the amount of software and hardware documentation;
- check the adequacy of the model to assess the complexity of testing Yii1-projects;
- develop software to assess the complexity of testing, using the constructed nonlinear regression model.

The object of research is the process of estimating the complexity of testing Yii1 projects.

The subject of research is a nonlinear multifactor regression model for estimating the complexity of testing Yii1-projects.

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

The scientific novelty of the obtained results. The nonlinear regression model based on data normalization using the decimal logarithm has been improved, which allows to increase the reliability of estimating the complexity of testing Yii1 projects.

The practical significance of the obtained results is the development of software for estimating the complexity of project testing based on nonlinear multifactor regression model and decimal logarithmic normalization transformation.

Personal contribution of the applicant. Conducted research, collected empirical data and the results obtained in the master's thesis, obtained by the applicant personally.

Approbation of research results. The results of the research presented in the paper were published at the II All-Ukrainian scientific-practical Internet conference on October 26-28, 2021.

Publications. According to the results of the research, the abstracts of the conference were published.

Keywords: TESTING EFFORT, MULTIFACTOR REGRESSION MODEL, WEB APPLICATIONS, Yii1, PHP.

ЗМІСТ

ВСТУП	10
1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ , ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ YII1	13
1.1 Аналіз особливостей тестування платформи Yii1	13
1.2 Аналіз існуючих моделей оцінювання трудомісткості тестування за стосунків	16
1.3 Висновки	21
2 МАТЕМАТИЧНА МОДЕЛЬ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ , ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ YII1	22
2.1 Отримання апріорних даних	24
2.2 Регресійна модель для оцінювання трудомісткості тестування застосунків, що створюються за допомогою платформи Yii1	25
2.3 Оцінювання адекватності регресійної моделі	26
2.4 Постановка задачі на розробку програмного забезпечення для реалізації математичної моделі для оцінювання трудомісткості тестування застосунків, що створюються за допомогою платформи YII1	27
3 ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ , ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ YII1	30
3.1 Ескізний проект	
3.1.1 Діаграма варіантів використання	
3.1.2 Концептуальна модель	
3.2. Технічний проект	
3.3. Робочий проект	
3.3.1 Вибір мови та середовища розробки програмного забезпечення	
3.3.2 Кодування та випробування ПЗ	

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	ДЛЯ
ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ , ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ YPI	39
5 ОХОРОНА ПРАЦІ	45
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	56
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ	67
ДОДАТОК Б – ТЕКСТ ПРОГРАМИ	71
ДОДАТОК В – ПРОГРАМА І МЕТОДИКА ВИПРОБОВУВАНЬ	74
ДОДАТОК Г – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	75
ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА	76

ВСТУП

Актуальність. Здатність програмного продукту дозволити користувачам досягати визначених в технічному завданні цілей за допомогою ефективності, продуктивності, безпеки та задоволення від зручного, швидкого та безвідмовного використання програмного забезпечення завжди є важливим фактором у взаємовідносинах з бізнесом. Тому оцінювання необхідної трудомісткості тестування є важливою складовою при плануванні розробки програмного забезпечення.

Техніка динамічного тестування вимагає, щоб певні елементи програми були охоплені, були чутливими до помилок, пов'язаних з цими елементами. Крім того, ймовірність помилок зростає зі складністю. Складність можна охарактеризувати з точки зору кількох властивостей, які можна використовувати, щоб запропонувати різні стратегії тестування. Складність різних властивостей програмного забезпечення можна виміряти за допомогою відповідних показників складності. Властивості з незвичайними мірами високої складності повинні бути перевірені дуже ретельно. У роботах [1,2] авторами було опубліковано математичну модель для оцінювання трудомісткості тестування програмного забезпечення в рамках гнучкої методології, але уточнення та удосконалення цієї моделі для фреймворку Yii1, з метою зниження довірчого інтервалу та підвищення точності оцінювання є актуальним питанням.

Приклади показників якості у використанні програмного забезпечення наведено в ISO/IEC TR 9126. За даними компанії IBM [3] на ранніх етапах проекту тестування можна задати високорівневі оцінки часу, необхідного для виконання дій з планування тесту, часу і трудомісткості виконання всіх тестів. Одиницею вимірювання трудомісткості тестування можуть бути: РН: людино-година; РД: людино-день; РМ: людино-місяць або РУ: людино-рік.

Ці оцінки часто ґрунтуються на тому, що відомо про вимоги до проекту з технічного завдання, але найчастіше беруть класичні 20 % часу. Але виходячи з практичного досвіду можна зробити висновок, що кількість дефектів програмного забезпечення не є сталою величиною, а змінюється на протязі всього життєвого циклу програмного забезпечення. Тому розробка математичної моделі яка ляже в основу автоматизованого інструменту оцінки трудомісткості тестування програмного забезпечення є актуальним завданням.

Мета і завдання дослідження. Метою даної магістерської роботи є підвищення достовірності оцінювання трудомісткості тестування Yii1-проектів. Для досягнення поставленої мети необхідно вирішити такі завдання:

5. проаналізувати існуючі моделі та методи оцінювання трудомісткості тестування ПЗ, сформулювати постановку задачі;
6. розробити регресійну модель оцінювання трудомісткості тестування Yii1-проектів залежно від кількості тестових випадків, дефектів, які необхідно перевірити, та обсягу програмно-технічної документації;
7. перевірити адекватність моделі для оцінювання трудомісткості тестування Yii1-проектів;
8. розробити ПЗ для оцінювання трудомісткості тестування, використовуючи побудовану нелінійну регресійну модель.

Об'єктом дослідження є процес оцінювання трудомісткості тестування Yii1-проектів.

Предметом дослідження є нелінійна багатофакторна регресійна модель оцінювання трудомісткості тестування Yii1-проектів.

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

Наукова новизна отриманих результатів. Удосконалено багатфакторну регресійну модель оцінювання трудомісткості тестування Yii1-проектів на основі нормалізації даних за допомогою десятичного логарифму, що дозволяє підвищити достовірність оцінювання трудомісткості тестування Yii1-проектів.

Практичне значення отриманих результатів полягає в розробці програмного забезпечення для оцінювання трудомісткості тестування проектів на основі нелінійної багатфакторної регресійної моделі та десятичного логарифмічного нормалізуючого перетворення.

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

Апробація результатів роботи. Результати досліджень, викладених у роботі, були оприлюднені на II Всеукраїнській науково-практичній інтернет конференції 26-28 жовтня 2021 р.

Публікації. За результатами досліджень було опубліковано тези конференції.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ , ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ YII1

1.1 Аналіз особливостей тестування платформи Yii1

Специфічною важливою особливістю фреймворку Yii є те, що він чудово справляється з кешуванням, а за рахунок швидких рішень для розробника економить багато часу, необхідного для створення сайтів, сервісів та додатків.

Не менш важливою позитивною якістю є і те, що платформа володіє зручним набором інструментів, які допоможуть сформувати ефективні і трендові інтерфейси, внаслідок чого її дуже високо оцінюють UI-фахівці. До того ж, Yii перевершує всі інші подібні системи у швидкості роботи завдяки практично миттєвому завантаженню.

Також у цьому фреймворку передбачено інтеграцію з jQuery. Це дуже полегшує роботу команди фахівців над проектом, а також взаємодію із сторонніми розробниками. У Yii доступні елементи, призначені для швидкої розробки.

Водночас усі вищезгадані переваги створюють і деякі проблеми. Основна з них у тому, що фреймворк не підходить для розробника-початківця. Yii1 має завищений поріг входження та велику кількість особливостей, через наявність яких навіть найменша помилка може призвести до пошкоджень у підсумковому продукті. Наприклад, некоректне звернення може значно збільшити обсяг згенерованого коду, що негайно позначиться на швидкодії, яка заявляється як основна перевага. Тим не менш, належний рівень підготовки спеціаліста забезпечить володіння платформою як максимально ефективним інструментом для швидкого створення сайту,

сервісів та програм. Саме тому Yii першої версії часто стає вибором досвідчених фахівців у всьому світі.

Як видно на рисунку 1.1 по даним Google Trends цікавість користувачів аспектами розвитку різних версій, як першої так і найновішої постійно присутня серед пошукових запитів.

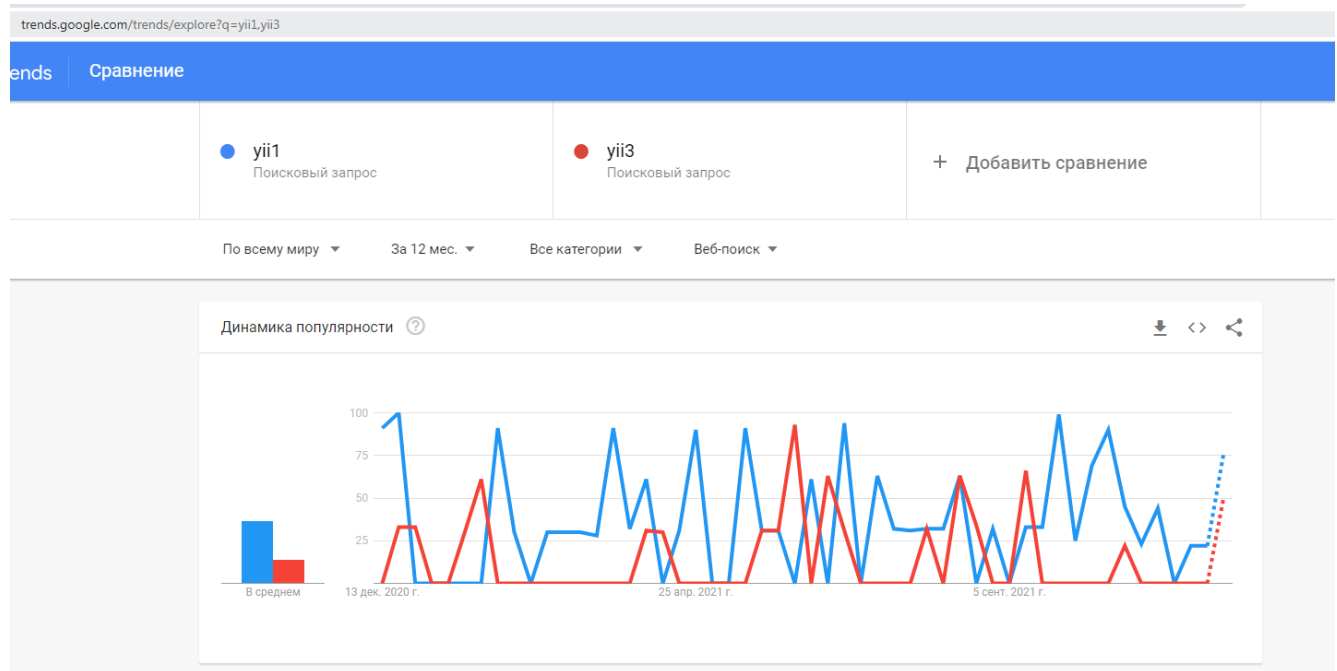


Рисунок 1.1 - Дані Google Trends з порівняння частоти пошукових запитів з версій Yii1 та Yii3

Першу версію Yii було випущено у листопаді 2008 року. Вона була створена як покращення фреймворку Prado, і завдяки своїй простоті та гарній документованості і отримала багато прихильників. У 2010 році вийшла наступна версія 1.1, яка вже мала конструктор форм, підтримку реляційних запитів Active record та готовий каркас для unit тестів. Перша версія фреймворку підтримує php 5.1 та вище. Друга версія фреймворку вийшла у 2014 році.

В Yii1 та Yii2 різна структура ядра, встановлення було без composer, всі класи починались з префіксу c., версія мови php 7 не підтримувалася, не було простору імен, розширеної та базової версій, не було по замовченню встановленого jQuery та Bootstrap.

Обсяг коду в Yii1 порівняно з наступними версіями був значно більший та складніший для аналізу. З моменту виходу Yii1 було створено безліч Yii1 проектів, усі вони потребують розвитку, нарощують функціонал, який потребує тестування, а сумісність між версіями відсутня, тому актуальність задачі оцінки трудомісткості тестування таких проектів є беззаперечною.

Найбільший ефект у зниженні трудомісткості розробки програмного забезпечення досягається на фазах створення дизайну та тестування. Тому актуальність розробки додаткових інструментів оцінювання не згасає.

За документацією компанії IBM основний порядок дій при тестуванні наведено на рисунку 1.2.

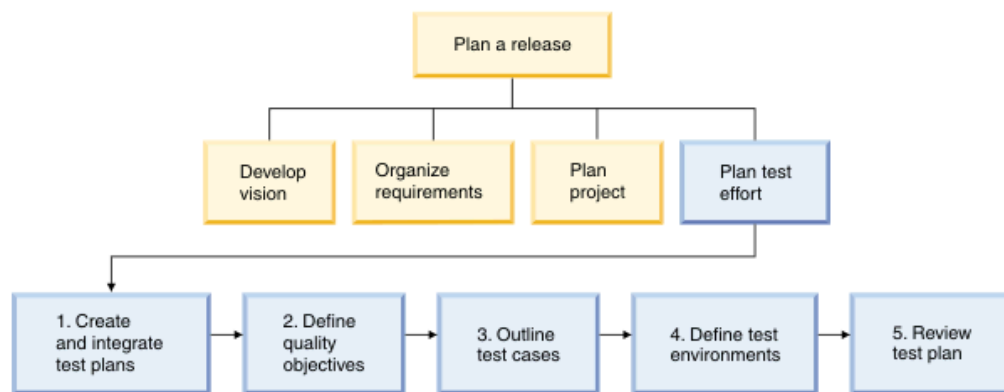


Рисунок 1.2 - План тестування за версією компанії IBM

Найпопулярнішим і найчастіше використовуваним фреймворком для юніт-тестування Yii1 є PHPUnit, скриншот результатів роботи якого наведено на рисунку 1.3.

```

+ PHPUnit 4.7.7 by Sebastian Bergmann and contributors.
- Runtime:      PHP 5.5.9-1ubuntu4.13 with Xdebug 2.2.3
  Configuration: /home/elisdn/test-project/phpunit.xml

.....

Time: 506 ms, Memory: 13.00Mb

OK (5 tests, 11 assertions)
  
```

Рисунок 1.3 – Інструмент Unit-тестування Yii1 - PHPUnit

Приклади показників якості з детальним описом використання наведені в стандарті ISO/IEC TR 9126 [5].

Здатність програмного продукту дозволити певним користувачам досягати визначених цілей за допомогою ефективності, продуктивності, безпеці та задоволення від використання завжди є важливим фактором у взаємовідносинах з бізнесом. Тому оцінювання необхідної трудомісткості тестування є дуже важливою складовою планування програмного забезпечення.

1.2 Аналіз існуючих моделей оцінювання трудомісткості тестування застосунків

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

Одиницею вимірювання трудомісткості тестування можуть бути:

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

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

Таблиця 1.1 Типи проекту

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

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

Атрибути продуктів:

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

Атрибути технічного забезпечення проектів програмного забезпечення (Computer Attributes):

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

Атрибути фахівців:

- **ACAP** Analyst Capability;
- **AEXP** Application Experience;

- **PCAP** Programming Capability;
- **VEXP** Virtual Machine Experience;
- **LEXP** Programming Language Experience.

Атрибути проектів:

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

На рисунку 1.4 наведено трудомісткість для цих типів проектів.

Effort 100k -> 500k LOC

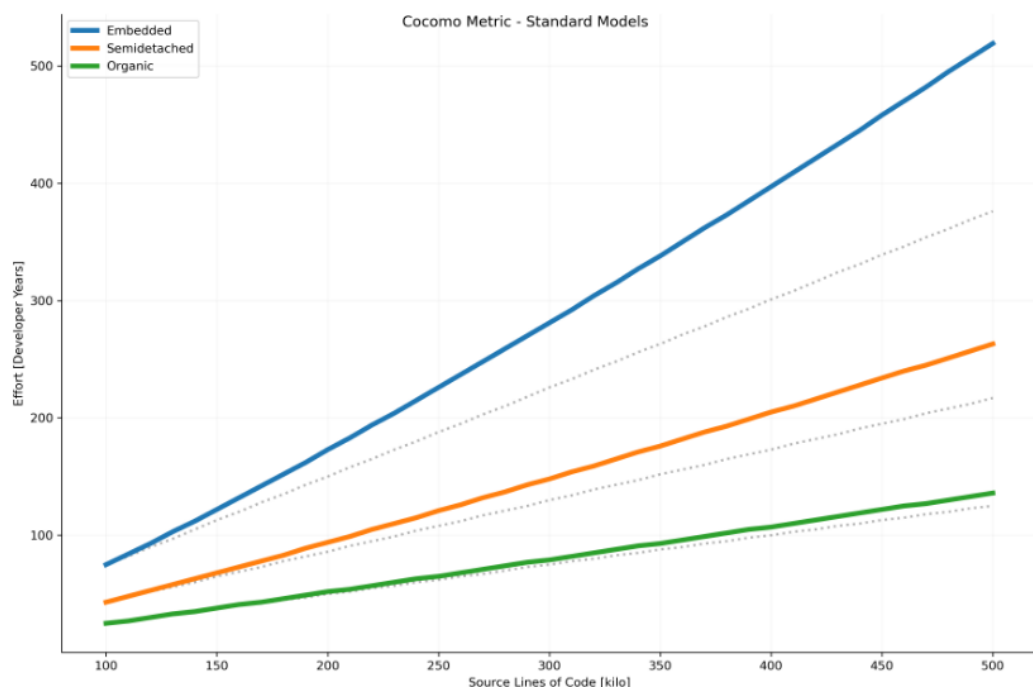


Рисунок 1.4 – Трудомісткість 3 типів проектів

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

- **COQUALMO** (Constructive Quality Model) – модель для оцінювання трудовитрат на тестування, яка є похідною від моделі **COCOMO**. Вона призначена для оцінювання трудових витрат на розробку програмного продукту в цілому (з урахуванням усіх етапів);

- методи, засновані на інформації про кількість рядків коду SLOC (Source Lines of Code);
- методи на основі функціональних точок та варіантів використання;
- методи експертних оцінок, що використовують попередній досвід розробки аналогічних проектів;
- моделі оцінювання трудовитрат на тестування як відсотків від трудовитрат на кодування ПЗ [1-3].

В реальних умовах практично неможливо отримати такі дані про програмні проекти, які будуть розподілені за нормальним законом, і зв'язки між цими даними не завжди є лінійними. Це доводять результати побудови інтервалів передбачення, наведені в табл. 1.1. Інтервал, який було побудовано за допомогою нелінійної регресії та нормалізації даних, не має від'ємних значень нижньої границі для жодного з програмних проектів і для 8 з 14 проектів є меншим на 43.65% за інтервал передбачення лінійної регресії.

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

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

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

Використання командою проекту цих методів та моделей є одним з головних аргументів при техніко-економічному обґрунтуванні вартості

розробки. Але частина з цих підходів вже застаріла та їх використання дає неадекватний результат, який приводить до несприятливих наслідків.

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

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

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

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

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

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

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

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

Середня величина відносної похибки MMRE визначається за формулою: .

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

Рівень прогнозування обчислюється за формулою:

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

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

Із вищевикладеного випливає, що на сьогодні існує велика кількість методів та моделей оцінювання трудомісткості тестування, як одного з складових розробки але частина з цих підходів має низьку точність оцінювання, не враховує нелінійність зв'язків між факторами, призначена для даних, що підпорядковуються нормальному закону розподілу. Також відомо, що не існує універсальних методів чи моделей, які є оптимальними для всіх типів програмного забезпечення і середовищ розробки. Тому існує необхідність удосконалювати існуючі та розробляти нові методи та моделі оцінювання трудомісткості тестування програмного забезпечення розробленого за допомогою фреймворка Yii1. Це дозволить підвищити точність оцінювання трудомісткості тестування з урахуванням нелінійних зв'язків між даними, що не підпорядковуються нормальному закону розподілу.

1.3 Висновки

Виходячи з проведеного аналізу можна зробити висновок про доцільність розробки математичної моделі для оцінювання трудомісткості тестування програмних застосунків за допомогою фреймворку Yii1.

2 МАТЕМАТИЧНА МОДЕЛЬ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ ПЛАТФОРМИ YII1

Для побудови нелінійних регресійних моделей використовуються методи простого перебору, лінеаризуючих та нормалізуючих перетворень [20]. Для методу простого перебору задаються різні види рівняння регресії, з яких за певним критерієм обирається найкраще наближення. Основний недолік цього методу – великі затрати часу за рахунок великої кількості обчислень.

Метод лінеаризуючих перетворень дозволяє перейти від нелінійної регресії до лінійної за допомогою заміни вихідних змінних і коефіцієнтів. Але таку заміну не завжди можна підібрати, і вона призводить до втрати інформації за рахунок спрощення регресійної моделі [25, 26, 27].

Метод нормалізуючих перетворень полягає в пошуку перетворення, яке дозволить перейти від вихідних не гаусівських випадкових величин до гаусівських випадкових величин. Для перетворення гаусівських випадкових величин будується лінійне регресійне рівняння, яке за допомогою обраного нормалізуючого перетворення перетворюється в нелінійну модель [26].

Оскільки метод нормалізуючих перетворень дає змогу працювати з даними, які мають нелінійну залежність, є доцільним використати його в даній роботі. В якості нормалізуючого перетворення було обрано логарифмічне перетворення.

Аналіз існуючих методів оцінювання трудомісткості та особливостей тестування Проектів Yii1 показав, що розроблювана модель повинна відповідати певним вимогам з метою пришвидшення та спрощення виконання розрахунків. Ці вимоги стосуються факторів, які використовуються для побудови моделі.

Для побудови лінійної регресійної моделі випадкові величини повинні бути нормально розподіленими, а це виконується лише в окремих випадках, то будемо будувати нелінійну регресійну модель.

Згідно [2] при тестуванні QA спеціаліст проводить додаткову оцінку витрат на виконання тесту додаючи ваги окремих тестових наборів виходячи з їх комплексності, середньої та відносної цикломатичної комплексності на клас, та середньої складності та кількості тестових випадків на клас. Наприклад, тестовий набір з вагою 100 може бути вдвічі важливішим або складнішим порівняно з тестовим набором з вагою 50.

Детальна інформація про розміри використовується при складанні кількох звітів про стан виконання, таких як звіти Стану TER та Стан виконання за допомогою вагових коефіцієнтів. Надавши тестовим наборам різні ваги, можна підвищити точність звітів, у яких враховуватиметься як кількість виконаних тестових наборів, і точне стан виконання кожного тестового набору.

Виходячи з проведеного аналізу можна зробити висновок, що для отримання інтервалу передбачення оцінювання трудомісткості тестування УП1-проектів потрібно використати наступні фактори: кількість помилок – x_1 та середня відносна системна комплексність – x_2 .

За результатами отримаємо таблицю 2.1 з початковими даними для побудови моделі.

Для перевірки нормальності розподілу вихідних даних застосуємо критерій χ Пірсона. Якщо вихідні дані розподілені нормально, то можемо видаляти викиди і дублікати, та будувати лінійну регресійну модель (1.1), відповідні довірчі інтервали (1.11), інтервали прогнозування (1.12) у звичайний спосіб. Якщо ж вихідні дані мають негаусівський розподіл, то такі дані необхідно нормалізувати.

В якості нормалізуючого перетворення буде використовуватися десятковий логарифм.

2.1 Отримання апріорних даних

Збір даних проводився за допомогою програми `phpmetrics`, що доступна за посиланням <https://github.com/phpmetrics/PhpMetrics> на рисунку 2.1 а-г наведено приклади метрик, які отримано цим інструментом для оцінювання трудомісткості Yii1 проектів.

```
Complexity
Average Cyclomatic complexity by class 4.8
Average Weighted method count by class 6.43
Average Relative system complexity 71.59
Average Difficulty 3.76
Bugs
Average bugs by class 0.12
Average defects by class <Man> 0.42
Violations
Critical 0
Error 10
Warning 19
Information 3
```

```
Complexity
Average Cyclomatic complexity by class 6.25
Average Weighted method count by class 10.71
Average Relative system complexity 80.54
Average Difficulty 4.44
Bugs
Average bugs by class 0.28
Average defects by class <Man> 0.47
Violations
Critical 0
Error 16
Warning 14
Information 5
```

```
Complexity
Average Cyclomatic complexity by class 13.23
Average Weighted method count by class 18.02
Average Relative system complexity 99.61
Average Difficulty 10.17
Bugs
Average bugs by class 0.31
Average defects by class <Man> 0.92
Violations
Critical 0
Error 209
Warning 153
Information 58
```

```
Complexity
Average Cyclomatic complexity by class 20.53
Average Weighted method count by class 27.19
Average Relative system complexity 180.99
Average Difficulty 11.53
Bugs
Average bugs by class 0.5
Average defects by class <Man> 1.23
Violations
Critical 0
Error 45
Warning 30
Information 11
```

Рисунок 2.1 Результати вимірювання деяких Yii1 проектів, отриманих за допомогою програмного забезпечення `PHPmetrics`

Таблиця 2.1. Вибірка з метриками Yii1 проектів

Bugs (X1)	Complexity(X2)	Effort
10	71,59	16,68
209	99,61	11,5
16	80,54	12,03
4	80	14,88
45	180,99	13,75
25	330	18,11
12	110	8
37	210	17,83
30	1460	79,24
15	605	21,5
16	688	40,33
10	215	21
14	255	13,5
6	462	19,75
9	448	24
10	776	29
6	200	15,35
7	132	19
13	36	9,5
17	770	35,1
10	140	17,9
26	810	52,32
18	635	19,83
4	150	10,75
15	800	10,56

2.2 Регресійна модель для оцінювання трудомісткості тестування застосунків, що створюються за допомогою платформи Yii1

Побудова моделі виконується на підставі значень спостережень отриманих від програмного забезпечення `phrmetrics` за Yii1 проектами. Метрики вибрано виходячи з публікацій [18-23].

Вид мультиколінеарності, при якому факторні змінні пов'язані деякою стохастичною залежністю, називається частковою. Якщо між факторними змінними є високий ступінь кореляції, то матриця $(X^T X)$ близька до виродженої, тобто. $\det(X^T X) \geq 0$ (чим ближче до 0 визначник матриці

межфакторної кореляції, тим сильніше мультиколінеарності факторів і ненадійніше результати множинної регресії).

2.3 Оцінка адекватності регресійної моделі

Перевірку важливості регресії можна здійснити через обчислення р-значення. У цьому випадку обчислюють ймовірність того, що тестова статистика F прийме значення F_0 (ця ймовірність і є р-значення), потім порівнюють р-значення із заданим рівнем значущості. Якщо р-значення менше рівня значимості, є підстави для відхилення нульової гіпотези, і регресія значуща.

Матриця $X'X$		
25	584	9744,73
584	51994	194966,58
9744,73	194966,6	6717379,352

Матриця зворотня до $X'X$		
0,11228305	-0,00073	-0,000141704
-0,0007298	2,63E-05	2,94643E-07
-0,0001417	2,95E-07	3,45882E-07

Матриця $X'y$
551,41
12060,95
324855,835

Прогнозне значення		
x1	x2	y
1,9	0,3	7,1102823

Оцінки параметрів моделі

Матричний метод

b0	7,078426	7,078426284
b1	0,010801	0,010801282
b2	0,037779	0,037778511

Оцінка дисперсії залишків моделі σ^2

SSres=SSE	MSres=MSE - оцінка дисперсії
1764,62145	80,21006606
1764,62145	80,21006606

1764,62145

Стандартна помилка регресії

8,956007261

Тест на значущість коефіцієнтів регресії

Тест на значимість Регресії					
SSt=SSr+SSres					
SSt	5908,46058				
SSres=SSE	1764,62145				
MSres	80,2100661				
SSr	4143,83912	4143,839123	4143,83912	4143,839123	b'(X'y)
MSr	2071,91956				16305,95865
F0	25,8311664	25,83116637			
Рівень значимости альфа	0,05				
Критичне значення F-статистики	3,44335678	$F_0 > f_{\alpha,k,n-(k+1)}$			
Відхилити H0 (H0 – взаємозв'язок між X та Y незначний)?					
ИСТИНА					
Тест на значущість через p-значення					
p-значение	1,6864E-06				
Отклонить H0?	ИСТИНА				
Коеффициент детерминації R2					
70,1%	70,1%				
Нормированный коэффициент детерминации (R2 adjusted)					
67,4%					
Тест на значимость коэффициентов регрессии					
Дисперсии каждого коэффициента регрессии можно вычислить через матрицу ковариации					
Матрица ковариации					
	0	1	2	альфа	0,05
0	0,11228305	-0,00072981	-0,0001417	DF	22
1	-0,00072981	2,63254E-05	2,9464E-07	таьфа,DF	2,0739
2	-0,0001417	2,94643E-07	3,4588E-07		2,0739

Стандартные ошибки	Через матрицу $X'X^{-1}$	Через ЛИНЕЙН()	t-статистика
se(b0)	3,00103832	3,001038319	2,35865908
se(b1)	0,04595175	0,045951751	0,23505703
se(b2)	0,00526718	0,005267184	7,1724304

р-значення	Значимий коефіцієнт?
0,0276394	ИСТИНА
0,81633856	ИСТИНА
3,442E-07	ИСТИНА

У підсумку					
Дисперсійний аналіз					
	df	SS	MS	F	p-значення
Регресія	2	4 143,8	2 071,9	25,83	1,68636E-06
Остаток	22	1 764,6	80,2		
всього	24	5 908,5			
Рівняння регресії					
$y=7,07+0,01x_1+0,03x_2$					
Прогнозуючий фактор	Оцінка	Стандартна помилка	t-статистика	p-значення	
Сдвиг	7,078	3,001	2,36	0,028	
Кількість помилок (Complexity)	0,0108	0,0460	0,24	0,816	
	0,037779	0,005267	7,17	0,000	
Ст. помилка регресії	8,95601				
R ²	70,1%				

Отримано багатфакторну модель вигляду $Y=7.07+0.1 \cdot x_1+0.03x_2$. Параметри моделі R^2 0,7 є достатнім показником, MMRE дорівнює 0,23, PRED(0.25) дорівнює 0,6.

Проведемо нормалізацію за десятковим логарифмом. Багатфакторна лінійна модель, нормалізовані значення для якої наведено у таблиці 2.3 прийме вигляд $Z_Y=0.2695+0.015847x_1+0.48x_2$.

Для наведених емпіричних даних нелінійне рівняння регресії має вигляд

$$Y=10^{0.2695+\varepsilon} \cdot X^{0.015847} X^{0.48}$$

За результатами розрахунку довірчих та прогнозованих інтервалів а також параметрів моделей можна зробити висновок про найкращу якість нелінійної багатфакторної моделі. Довірчі та прогнозовані інтервали наведено у додатку Е.

2.4 Постановка задачі на розробку програмного забезпечення.

Виходячи з аналізу існуючих методів оцінювання трудомісткості, сформулюємо наступну постановку задачі: розробити проект програмного забезпечення для оцінювання трудомісткості тестування проектів, які створюються з використання фреймворків YII1. У склад програмного забезпечення слід внести побудовану математичну модель, яка допоможе оцінити трудомісткість тестування проектів.

Дане програмне забезпечення повинно мати наступні функції:

- 1) надавати можливість вносити значення метрик програмних проектів;
- 2) надавати прогнозоване середнє значення трудомісткості тестування проектів YII1 довірчий інтервал та інтервал передбачення;

3 ПРОЕКТ ПЗ ДЛЯ ОЦІНЮВАННЯ ТРУДОМІСТКОСТІ ТЕСТУВАННЯ YII1-ПРОЕКТІВ

3.1 Ескізний проект

3.1.1 Моделювання варіантів використання

Даний проект передбачає наявність лише одного актора – користувача програми, який буде повноцінно використовувати всі функції програмного продукту.

Виявлені варіанти використання знаходяться в таблиці 3.1.

Таблиця 3.1 – Виявлені варіанти використання

Основні актори	Найменування	Формулювання
Користувач	Внести дані про проекти	Вносить інформацію про проекти
Користувач	Внести назви проектів	Вносить інформацію про назву кожного проекту
Користувач	Внести кількість тестових випадків	Вносить інформацію про кількість тестових випадків для кожного проекту
Користувач	Внести кількість сторінок програмно-технічної документації	Вносить інформацію про кількість сторінок програмно-технічної документації для кожного проекту
Користувач	Внести кількість дефектів, які необхідно перевірити	Вносить інформацію про кількість дефектів, які необхідно перевірити, для кожного проекту
Користувач	Внести час тестування	Вносить інформацію про середній час тестування кожного проекту
Користувач	Нормалізувати дані	Виконує нормалізацію внесених даних
Користувач	Побудувати лінійне рівняння регресії та інтервал передбачення	Будує лінійну регресію оцінювання трудомісткості тестування Yii1-проектів та її інтервал передбачення
Користувач	Виконати оцінювання трудомісткості тестування Yii1-проектів та побудувати інтервал передбачення цієї оцінки	Виконує оцінювання трудомісткості тестування Yii1-проектів на основі нелінійної регресії та будує її інтервал передбачення

Конкретизуємо основні варіанти використання:

1. *Найменування*: Внести дані про проекти

Основна дійова особа: Користувач

Зв'язок з іншими варіантами використання: Узагальнює прецеденти «Внести назви проектів», «Внести кількість тестових випадків», «Внести кількість дефектів», «Внести час тестування».

Короткий опис:

Варіант використання «Внести дані про проекти» відображає три варіанти операцій і дає можливість внести інформацію про проекти у відповідні поля. Для варіанту «Внести назви проектів» – це назви кожного проекту, для варіанту «Внести кількість тестових випадків» – це кількість тестових випадків, розроблених для кожного з проектів, для варіанту «Внести кількість дефектів» – це кількість дефектів, які необхідно перевірити під час тестування, для варіанту «Внести час тестування» – це середній час тестування кожного проекту.

Потоки подій: Прецедент починається, коли у вікні програми користувач обирає комірку, яку необхідно заповнити даними.

Базовий потік – Внесення даних про проекти:

- 1) Користувач запускає програму;
- 2) Користувач заповнює стовпчик «Назва проекту» назвами проектів;
- 3) Користувач заповнює стовпчик «Y» даними про час тестування;
- 4) Користувач заповнює стовпчик «X1 (# TC)» даними про кількість тестових випадків;
- 5) Користувач заповнює стовпчик «X2 (# Des pg)» даними про кількість сторінок програмно-технічної документації.
- 6) Користувач заповнює стовпчик «X3 (# Def cr)» даними про кількість дефектів, які потрібно перевірити.

Альтернативі потоки:

1) Якщо у п. 3-6 внести значення, яке є меншим за 0 чи дорівнює 0, то буде виведено повідомлення про помилку та пропозиція повторити внесення даних. Внесене число не буде збережене.

Спеціальні вимоги: значення кількості та часу повинно завжди бути додатнім.

2. Найменування: Нормалізувати дані

Основна дійова особа: Користувач

Зв'язок з іншими варіантами використання: відсутній.

Короткий опис:

Варіант використання «Нормалізувати дані» дозволяє підрахувати початкове значення трудомісткості та, використовуючи метод простого логарифмування, виконати нормалізацію експериментальних даних для стовпчиків «Y», «X1 (# TC)», «X2 (# Des pg)», «X3 (# Def cr)». При цьому відповідні комірки заповнюються нормалізованими значеннями.

Потоки подій: Прецедент починається, коли користувач викликає макрос «Normalize».

Базовий потік – Нормалізація даних:

1) Користувач викликає макрос «Normalize»;

2) У відповідних комірках з'являються наступні значення: нормалізований Y, X₁, X₂, X₃.

3. Найменування: Побудувати лінійну регресію

Основна дійова особа: Користувач

Зв'язок з іншими варіантами використання: відсутній.

Короткий опис:

Варіант використання «Побудувати лінійну регресію» дозволяє отримати лінійне регресійне рівняння на основі нормалізованих даних та інтервал передбачення лінійної регресії. При цьому відповідні комірки заповнюються знайденими значеннями.

Потоки подій: Прецедент починається, коли користувач викликає макрос «BuildLinearRegression».

Базовий потік – Побудова лінійної регресії та інтервалу передбачення:

- 1) Користувач викликає макрос «BuildLinearRegression»;
- 2) У відповідних комірках з'являються наступні значення: вектору θ , лінійного рівняння регресії, нижньої та верхньої границь інтервалу передбачення лінійної регресії, а також результати проміжних розрахунків, які були використані для отримання інтервалу передбачення.

4. *Найменування:* Отримати інтервал передбачення оцінки трудомісткості тестування

Основна дійова особа: Користувач

Зв'язок з іншими варіантами використання: відсутній.

Короткий опис:

Варіант використання «Отримати інтервал передбачення оцінки трудомісткості тестування» дозволяє отримати остаточну оцінку трудомісткості тестування Yii1-проектів за допомогою нелінійної регресійної моделі та інтервалу передбачення для неї, які будуються на основі лінійної регресії та її інтервалу передбачення. При цьому відповідні комірки заповнюються знайденими значеннями.

Потоки подій: Прецедент починається, коли користувач викликає макрос «GetEstimateAndInterval».

Базовий потік – Отримання інтервалу передбачення оцінки трудомісткості тестування:

- 1) Користувач викликає макрос «GetEstimateAndInterval»;
- 2) У відповідних комірках змінюються наступні значення: оцінка трудомісткості тестування, нижня та верхня границі інтервалу передбачення цієї оцінки.

Діаграма варіантів використання зображена на рисунку 3.1.

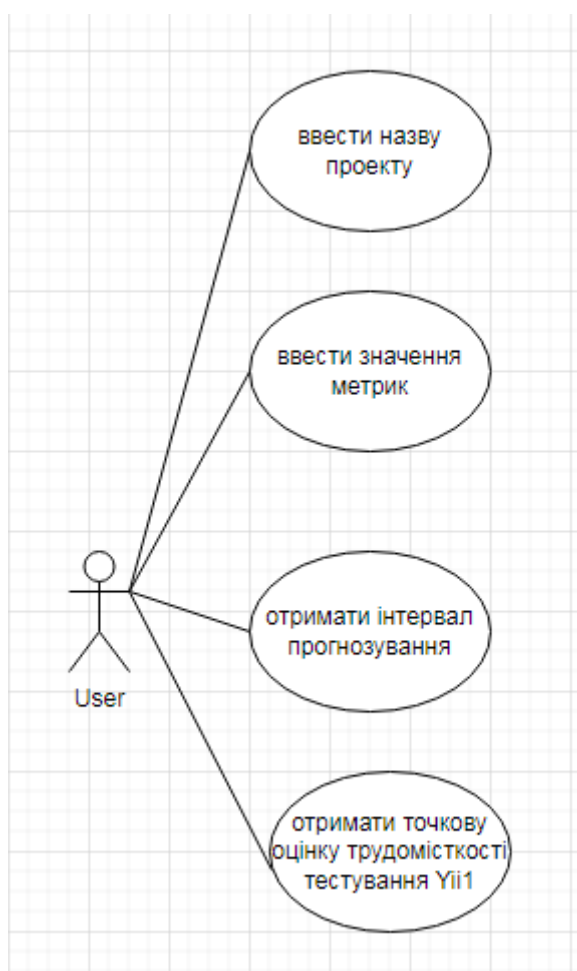


Рисунок 3.1 – Діаграма варіантів використання ПЗ для оцінювання трудомісткості тестування Yii1-проектів

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

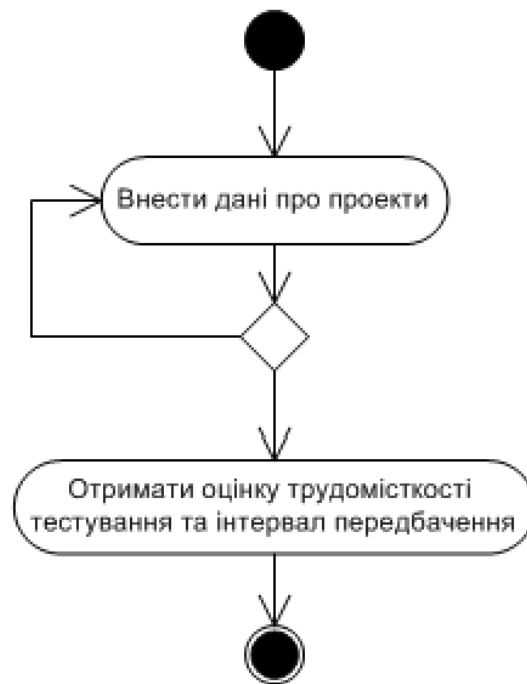


Рисунок 3.2 – Діаграма діяльності ПЗ для оцінювання трудомісткості тестування Yii1-проектів

3.1.2 Моделювання потоків даних

Контекстна діаграма потоків даних 0-го рівня зображена на рисунку 3.3.

Таблиця 3.2 – Словник даних для діаграми потоків даних 0-го рівня

Потік даних	Атрибути
Додавання даних про проекти	Додавання даних про проекти: - назви проектів; - к-ть тестових випадків; - к-ть сторінок документації; - к-ть дефектів; - час тестування.
Результат додавання даних про проекти	Результат додавання даних про проекти: - результат додавання назв - результат додавання к-ті тестових випадків; - результат додавання к-ті сторінок документації; - результат додавання к-ті дефектів; - результат додавання часу тестування.
Результат виконання розрахунку за математичною моделлю	- результат оцінювання трудомісткості тестування у інтервальних та точкових оінках

3.2 Технічний проект

3.2.1 Декомпозиція контекстних процесів діаграми потоків даних

Декомпозиція діаграми потоків даних виконана до рівня специфікації макросів та введення даних про проекти. Вона зображена у вигляді DFD 1-го рівня на рис. 3-4 та 3-5, а відповідність потоків даних DFD 0-го та 1-го рівнів представлена в таблиці 3.3.

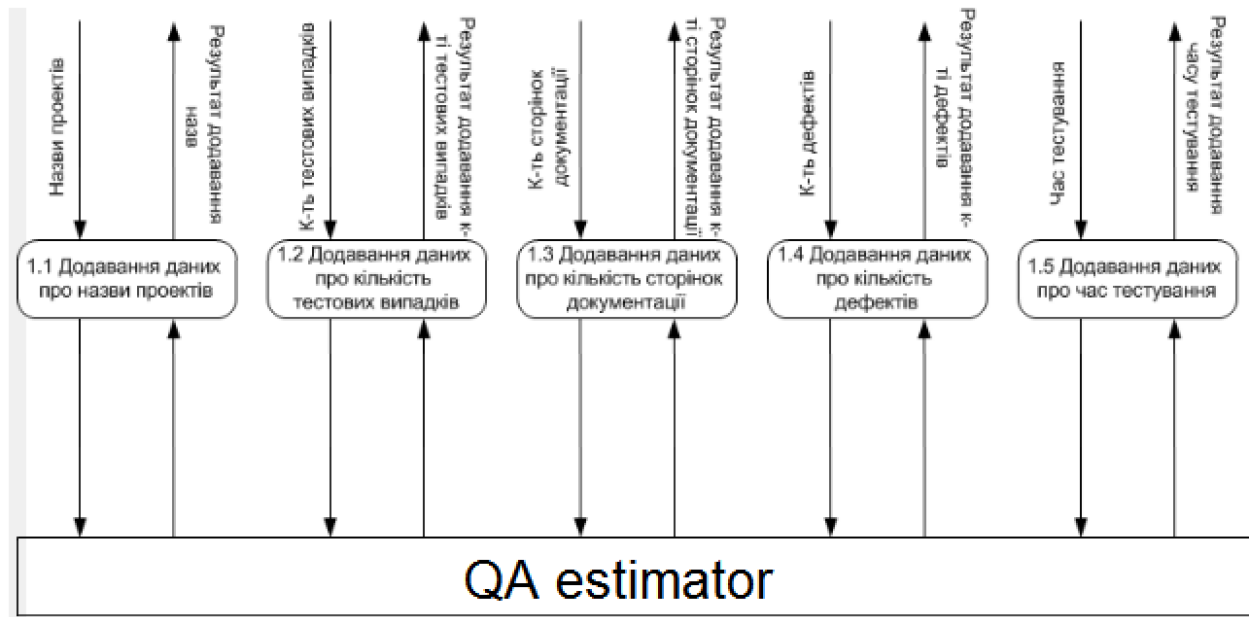


Рисунок 3.4 – Декомпозиція процесу «Додавання даних про проекти»

Таблиця 3.4 – Таблиця відповідності потоків даних 0-го та 1-го рівнів

Потік 0-го рівня	Потік 1-го рівня
Додавання даних про проекти	Додавання даних про проекти
Результат додавання даних про проекти	Результат додавання даних про проекти

3.2.2 Логічна модель даних

Логічна модель структури даних програми для оцінювання трудомісткості тестування веб-проектів представлена у вигляді ER-діаграми на рис. 3.6, а опис атрибутів сутностей – в таблицях 3.4-3.5.

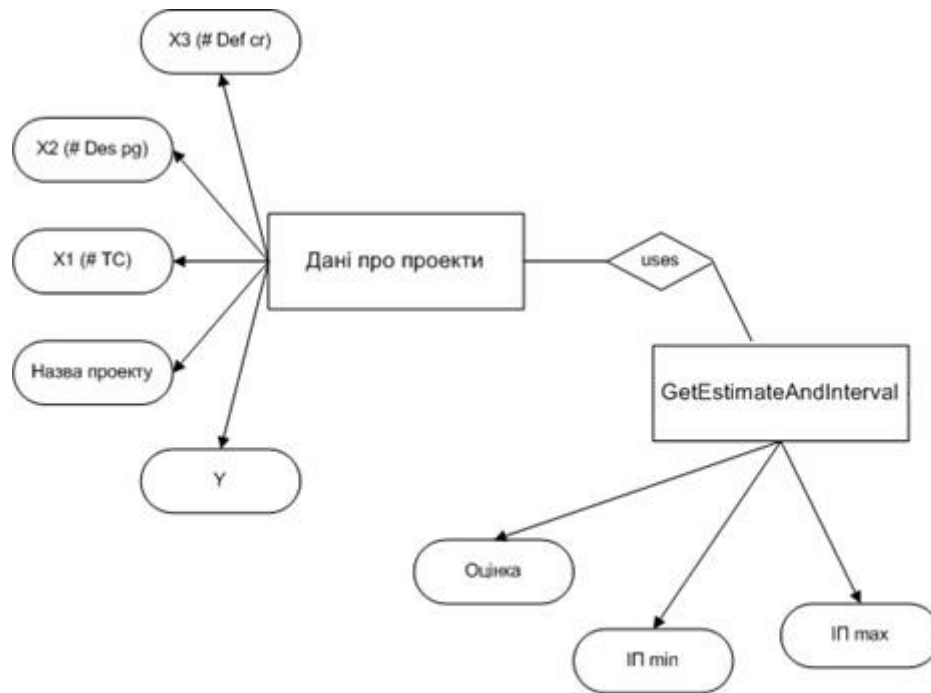


Рисунок 3.6 – ER-діаграма структури даних ПЗ для оцінювання
трудомісткості тестування Yii1-проектів

Таблиця 3.5 – Опис атрибутів сутності «Дані про проекти»

Поле	Тип	Обов'язкове
Назва проекту	символьний	ні
X1 (# TC)	дійсний	так
X2 (# Des pg)	дійсний	так
X3 (# Def cr)	дійсний	так
Y	дійсний	так

Таблиця 3.6 – Опис атрибутів сутності «Get Estimate And Interval»

Поле	Тип	Обов'язкове
Оцінка	дійсний	так
ІП min	дійсний	так
ІП max	дійсний	так

3.3 Робочий проект

Програма для оцінювання трудомісткості тестування веб-проектів розробляється на основі програмного рішення MS Office Excel 2010 від компанії Microsoft. Причиною такого вибору є наявність пакету MS Office практично на будь-якому комп'ютері (на відміну від, наприклад, аналогу OpenOffice Calc), що дозволяє уникнути встановлення додаткового програмного забезпечення. До того ж, більшість користувачів добре знайомі з MS Excel, тому для користування програмою оцінювання трудомісткості користувачу не потрібно буде проходити навчання (на відміну від нового програмного забезпечення з незнайомим інтерфейсом та функціями).

MS Excel 2010 є потужним засобом для статистичного аналізу даних, а сам файл, який містить макроси, займає на порядок менше дискового простору, ніж повноцінна скомпільована програма.

Мета тестового плану тестування програми оцінювання трудомісткості тестування Yii1-проектів є перевірка коректної роботи її функціональних модулів (макросів), зручності програми для користувача, а також виявлення побажань користувача з розширення або виправлення існуючих функціональних можливостей. Тестовий план визначає, яким чином буде встановлено факт коректності зазначеної програми.

Функціональні вимоги до програмного забезпечення

За допомогою програмного забезпечення для оцінювання трудомісткості тестування Yii1-проектів користувач повинен мати змогу:

- 1) вносити дані про програмні проекти (назви, кількість тестових випадків, сторінок програмної документації, дефектів);

- 5) розраховувати оцінку трудомісткості тестування Yii1-проектів та її інтервал передбачення за допомогою вбудованої мат моделі.

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

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

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

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

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

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

4.2 Розрахунок витрат на створення і експлуатацію програми

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

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

Таблиця 4.1 – Витрати на допоміжні матеріали

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

Вартість програми розрахуємо по формулі:

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

T – тривалість розробки (міс);

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

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

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

Z_e – витрати на електроенергію;

Z_m – витрати на основні і допоміжні матеріали;

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

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

Таблиця 4.2 – Витрати на розробку ПЗ.

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

Відповідно вартість програми:

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

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

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

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

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

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

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

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

264,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПК складе:

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

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

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

Експлуатаційні витрати для ПК за рік складатимуть:

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

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

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

4.3 Економічна ефективність розробки і впровадження програми

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

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

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

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

- підвищення продуктивності праці;
- вивільнення робочого часу.

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

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

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

Зарплата 0,6 працівника в рік складає:

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

Річний економічний ефект розраховується по формулі:

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

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

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

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

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

Строк окупності програми розраховується по формулі:

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

Отже строк окупності програмного продукту складає приблизно 5 місяців.

5 ОХОРОНА ПРАЦІ

Питання безпечної життєдіяльності людини необхідно вирішувати на всіх стадіях життя, це стосується і управління проектами зі створення програмного забезпечення.

Забезпечення безпечної життєдіяльності людини в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути які-небудь фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

У даному розділі вирішується питання безпечної життєдіяльності при управління проектами зі створення програмного забезпечення.

5.1 Аналіз шкідливих та небезпечних факторів при управлінні проектами зі створення програмного забезпечення

Небезпечні і шкідливі виробничі фактори[11] по природі виникнення поділяються на наступні групи:

- фізичні;
- хімічні;
- психофізіологічні;
- біологічні.

У приміщенні лабораторії на менеджера проектів можуть негативно діяти наступні фізичні фактори:

- підвищена і знижена температура повітря;

- надмірна запиленість і загазованість повітря;
- підвищена і знижена вологість повітря;
- недостатня освітленість робочого місця;
- перевищуючі припустимі норми шуму;
- підвищений рівень статичної електрики;
- небезпека ураження електричним струмом;
- бляклість екрана дисплея.

До хімічно небезпечних факторів, що постійно діють на менеджера проектів відносяться наступні:

- виникнення, у результаті іонізації повітря при роботі комп'ютера, активних часток.

Біологічні шкідливі виробничі фактори в даному приміщенні відсутні.

До психологічно шкідливих факторів, що впливають на оператора протягом його робочої зміни можна віднести наступні:

- нервово - емоційні перевантаження;
- розумові перевантаження;
- перевантаження зорового аналізатора.

Далі більш докладно розглянуті небезпечні і шкідливі фактори, що впливають на менеджера проектів, що виникли в зв'язку з розробкою даної системи та способи їх запобігання.

Мікроклімат виробничих приміщень - це клімат внутрішнього середовища цих приміщень, що визначається діючими на організм людини сполученнями температури, вологості і швидкості руху повітря

Офіс є приміщенням 1 категорії (виконуються легкі фізичні роботи), тому повинні дотримуватися наступні вимоги:

- оптимальна температура повітря - 220С (припустима - 20-240С),
- оптимальна відносна вологість - 40 -60% (припустима - не більш 75%) ,
- швидкість руху повітря не більш 0.1 м/с.

Для створення й автоматичної підтримки в лабораторії незалежно від зовнішніх умов оптимальних значень температури, вологості, чистоти і швидкості руху повітря, у холодний час року використовується водяне опалення, у теплий час року застосовується кондиціонування повітря. Кондиціонер являє собою вентиляційну установку, яка за допомогою приладів автоматичного регулювання підтримує в приміщенні задані параметри повітряного середовища.

Робота, виконувана з використанням обчислювальної техніки, має наступні недоліки:

- ймовірність появи прямого блиску;
- погіршена контрастність між зображенням і фоном;
- відбивання екрана.

У зв'язку з тим, що природне освітлення слабке, на робочому місці повинне застосовуватися також штучне освітлення. Далі буде зроблений розрахунок штучного освітлення.

Розміщення світильників визначається наступними розмірами:

$H = 3$ м. - висота приміщення

$h_c = 0,25$ м. - відстань світильників від перекриття

$h_n = H - h_c = 3 - 0,25 = 2,75$ м. - висота світильників над підлогою

h_p = висота розрахункової поверхні = 0,7 м (для приміщень, зв'язаних з роботою ПЕОМ)

$h = h_n - h_p = 2,75 - 0,7 = 2,05$ - розрахункова висота.

Світильники типу ЛДР (2x40 УТ). Довжина 1,24 м, ширина 0,27 м, висота 0,10 м.

L - відстань між сусідніми світильниками (рядами люмінесцентних світильників), L_a (по довжині приміщення) = 1,76 м, L_b (по ширині приміщення) = 3 м.

l - відстань від крайніх світильників або рядів світильників до стіни, $l = 0,3 - 0,5L$.

$l_a = 0,5L_a$, $l_b = 0,3L_b$

$l_a = 0,88$ м., $l_b = 0,73$ м.

Світильники з люмінесцентними лампами в приміщеннях для роботи рекомендують установлювати рядами.

Метод коефіцієнта використання світлового потоку призначений для розрахунку загального рівномірного освітлення горизонтальних поверхонь при відсутності великих предметів, що затемнюють. Потрібний потік ламп у кожному світильнику

$$\Phi = E \cdot r \cdot S \cdot z / N \cdot \eta,$$

де E - задана мінімальна освітленість = 300 лк., так як розряд зорових робіт = 3

r - коефіцієнт запасу = 1,3 (для приміщень, зв'язаних з роботою ПЕОМ)

S - освітлювана площа = 30 м².

z - характеризує нерівномірне освітлення, $z = E_{cp} / E_{min}$ - залежить від відношення $\square = L/h$, $\square_a = La/h = 0,6$, $\square_y = Lv/h = 1,5$. Так як $\square$ перевищує припустиме значення, то $z=1,1$ (для люмінесцентних ламп).

N - число світильників, до розрахунку. Спочатку намічається число рядів n , що підставляється замість N . Тоді Φ - потік ламп одного ряду.

$N = \Phi / \Phi l$, де Φl - потік ламп у кожному світильнику.

$\square$ - коефіцієнт використання. Для його знаходження вибирають індекс приміщення i і приблизно оцінюються коефіцієнти відображення поверхонь приміщення $\square_{стл.}$ (стелі) = 70%, $\square_{ст.}$ (стіни) = 50%, $\square_p$ (підлоги) = 30%.

$$\Phi = 300 \square 1,3 \square 25 \square 1,1/2 \square 0,3 = 21450 \text{ лм.}$$

Я пропоную установити два світильники в ряд. Світильники вміщаються в ряд, тому що довжина ряду близько 4 м. Застосовуємо світильники з лампами 2x40 Вт із загальним потоком 5700 лм.

У приміщеннях з низьким рівнем загального шуму, яким є офіс де працює програміст, джерелами шумових перешкод можуть стати вентиляційні установки, кондиціонери або периферійне устаткування для ЕОМ (плотери, принтери й ін). Тривалий вплив цих шумів негативно позначаються на емоційному стані персоналу.

Відповідно ДО ДЕРЖСТАНДАРТУ 12.1.003-76 ССБТ еквівалентний рівень звуку не повинний перевищувати 50 дБА. Для того, щоб домогтися цього рівня шуму рекомендується застосовувати звукопоглинаюче покриття стін.

Як міри по зниженню шуму можна запропонувати наступне:

— облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 дБ);

- екранування робочого місця (постановкою перегородок, діафрагм);
- установка в комп'ютерних приміщеннях устаткування, що робить мінімальний шум;
- раціональне планування приміщення.

Тому я пропоную для зменшення шуму в лабораторії використовувати замість матричного принтера, що робить багато шуму, більш тихий – лазерний принтер.

Захист від шуму варто виконувати відповідно до ДСТ 12.1.003-76, а звукоізоляція огорожжуваних конструкцій повинна відповідати вимогам розділу Сніп 11-12-77 «Захист від шуму. Норми проектування».

Приміщення офісу по небезпеці ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, безпильне, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

На робочому місці менеджера проектів з всього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартам фірми IBM, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення. Таким чином, устаткування обмінного пункту виконано по класу 1

Електробезпечність приміщення забезпечується відповідно до ПУЕ. Небезпечний і шкідливий вплив на людей електричного струму, електричної дуги й електромагнітних полів виявляється у виді електротравм і професійних захворювань.

Ступінь небезпечного і шкідливого впливу на людину електричного струму, електричної дуги й електромагнітних полів залежить від:

- Роду і величини напруги і струму
- Частоти електричного струму
- Шляхи струму через тіло людини
- Тривалості впливу на організм людини

Електробезпеність у приміщенні лабораторії забезпечується технічними способами і засобами захисту, а так само організаційними і технічними заходами.

Розглянемо основні причини ураження людини електричним струмом на робочому місці:

- Дотик до металевих неструмоведучих частин (корпусові, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції.

- Нерегламентоване використання електричних приладів.

Відсутність інструктажу співробітників по правилах електробезпеності.

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кв/м, тобто в 10 разів перевищує норму 20 кв/м. Для зменшення напруженості застосовувати зволожувачі і нейтралізатори, антистатичне покриття підлоги.

Крім того, при несправності яких-небудь блоків комп'ютера корпус може виявитися під струмом, що може привести до електричних травм або електричних ударів. Для усунення цього я пропоную забезпечити приєднання металевих корпусів устаткування до жили, що заземлює.

Електробезпе́чність забезпечується відповідно до ДСТУ 12.1. 030. - 81. Небезпечний і шкідливий вплив на людей електричного струму виявляється у виді електротравм і професійних захворювань.

Електробезпе́чність у лабораторії забезпечується технічними способами і засобами захисту, а так само організаційними і технічними заходами.

Розглянемо основні причини ураження менеджера проектів електричним струмом на робочому місці:

1. Дотик до металевих неструмоведучих частин системного блоку ПЕОМ, які можуть здійснюватись під напругою в результаті ушкодження ізоляції.

2. Заборонене використання електричних приладів, таких як електричні плити, чайники, обігрівачі.

Тому що усі струмоведучі частини ЕОМ ізольовані, той випадковий дотик до струмоведучих частин виключено.

Для забезпечення захисту від ураження електричним струмом при дотику до металевих неструмоведучих частин, що можуть виявитися під напругою в результаті ушкодження ізоляції, я рекомендую застосовувати захисне заземлення.

Заземлення корпусу ЕОМ забезпечено підведенням жили, що заземлює, до живильних розеток. Опір заземлення 4 Ом, згідно (ПУЕ) для електроустановок з напругою до 1000 В.

Основним організаційним елементом є інструктаж і навчання безпечним методам праці, а також перевірка знань правил безпеки й інструкцій відповідно до займаної посади стосовно до виконуваної роботи.

При проведенні незапланованого і планового ремонту обчислювальної техніки виконуються наступні дії:

- Відключення комп'ютера від мережі
- Перевірка відсутності напруги

Після виконання цих дій проводиться ремонт несправного устаткування.

Якщо ремонт проводиться на струмоведучих частинах, що знаходяться під напругою, то виконання роботи проводиться не менш чим двома особами з застосуванням електрозахисних засобів.

Ступінь вогнестійкості будинків приймається в залежності від їхнього призначення, категорії по вибухопожежній і пожежній небезпеці, по поверховості, площі поверху в межах пожежного відсіку.

Будинок, у якому знаходиться офіс по пожежній небезпеці будівельних конструкцій відноситься до категорії К1 (малопожежонебезпечні), оскільки тут присутні займисті (книги, документи, меблі, оргтехніка і т.д.) і легкогорючі речовини (сейфи, різне устаткування і т.д.), що при взаємодії з вогнем можуть горіти без вибуху.

По конструктивних характеристиках будинок можна віднести до будинків з несучими і огорожуючими конструкціями із природних або штучних кам'яних матеріалів, бетону або залізобетону, де для перекриттів допускається використання дерев'яних конструкцій, захищених штукатуркою або важкогорючими листовими, а також плитними матеріалами.

Отже, ступінь вогнестійкості будинку можна визначити як третю (III).

Приміщення лабораторії по функціональній пожежній небезпеці відноситься до класу Ф 4.2 – вищі навчальні заклади, установи підвищення кваліфікації.

Пожежна профілактика являє собою комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, на запобіганні пожежі, обмеження його поширення, а також створення умов для

успішного гасіння пожежі. Для профілактики пожежі надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів і засобів пожежопередження і захисту.

Одне з умов забезпечення пожежонебезпеки - ліквідація можливих джерел запалення.

У лабораторії джерелами запалення можуть бути:

- несправне електроустаткування, несправності в електропроводці, електричних розетках і вимикачах. Для виключення виникнення пожежі з цих причин необхідно вчасно виявляти й усувати несправності, проводити плановий огляд і вчасно усувати всі несправності;
- несправні електроприлади. Необхідні міри для виключення пожежі містять у собі своєчасний ремонт електроприладів, якісне виправлення поломок, не використання несправних електроприладів;
- обігрівання приміщення електронагрівальними приладами з відкритими нагрівальними елементами. Відкриті нагрівальні поверхні можуть спричинити пожежу, тому що в приміщенні знаходяться паперові документи і довідкова література у виді книг, посібників, а папір – легкозаймистий предмет. З метою профілактики пожежі пропоную не використовувати відкриті обігрівальні прилади в приміщенні лабораторії;
- коротке замикання в електропроводці. З метою зменшення імовірності виникнення пожежі внаслідок короткого замикання необхідно, щоб електропроводка була схованою.
- влучення в будинок блискавки. У літній період під час грози можливе влучення блискавки внаслідок чого можливий пожежа. Щоб уникнути цього я рекомендую установити на даху будинку блискавковідвід;
- недотримання мір пожежної безпеки і паління в приміщенні також може спричинити пожежу. Для усунення загоряння в результаті

паління в приміщенні лабораторії треба категорично заборонити паління, а дозволити тільки в строго відведеному для цього місці.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Система державних, суспільних та міжнародних заходів, які забезпечують раціональне використання, відновлення, примноження та збереження природних ресурсів від руйнування, забруднення та виснаження. Охорона навколишнього середовища має велике економічне та соціально-політичне значення, вона здійснюється з господарською, науковою, оздоровчою та культурною метою. При оцінюванні наслідків антропогенного впливу на навколишнє середовище важливе місце належить визначенню допустимих масштабів впливу, зокрема гранично допустимих концентрацій різних речовин — забруднювачів атмосфери, води та ґрунту. Кількісно та якісно вплив людини на навколишнє середовище стрімко зростає при НТП. Тому, починаючи з середини ХХ ст., значне виснаження природних багатств та забруднення навколишнього середовища змусили органи влади багатьох країн вжити заходів з охорони надр, атмосферного повітря, вод, лісів, тваринного та рослинного світу.

У більшості країн з високим та середнім рівнем розвитку (у т.ч. і в Україні) створені національні парки, заповідники, заказники та інші території, що охороняються. Згодом прийшло розуміння того, що НТП не лише не суперечить охороні навколишнього середовища, а в сучасних умовах тільки використання його досягнень — єдиний реальний засіб забезпечити охорону навколишнього середовища. при одночасному задоволенні різноманітних потреб людської цивілізації за рахунок єдиного в кінцевому підсумку джерела задоволення цих потреб, наявного на нашій планеті — природних ресурсів. Тому головну увагу стали приділяти впровадженню маловідходних технологій, обладнання для знешкодження викидів, стоків та відходів, економічному стимулюванню робіт з охорони навколишнього середовища [19].

Основними напрямками робіт з охорони навколишнього середовища. на сьогодні є:

- 1) якнайповніше вилучення з надр Землі мінеральної сировини при видобутку; її комплексна переробка; попередження забруднення атмосферного повітря, вод та ґрунтів промисловими, комунальними та іншими викидами, відходами, отрутохімікатами.
- 2) Система заходів, спрямованих на підтримання взаємодії між діяльністю людини та навколишнього природного середовища, яка забезпечує збереження та відновлення природних ресурсів, попереджає прямий та непрямий вплив наслідків діяльності суспільства на природу та здоров'я людини.
- 3) Система заходів, спрямованих на економічно ефективно якнайповніше вилучення природних ресурсів та використання природних умов, забезпечення мінімальних їх питомих витрат на одиницю готової продукції з метою забезпечення збереження природно-ресурсного потенціалу.
- 4) Комплексна міжгалузева наукова дисципліна, яка розробляє загальні принципи та методи збереження і відновлення природних ресурсів.

Сьогодні ми не можемо уявити собі життя без персонального комп'ютера чи мобільного телефону. Електроніка всюди – вдома та на роботі, в школі та ВНЗ, в аптеці та банку, на вокзалі, в магазині, в лікарні і т.д. Кількість електроніки зростає з кожним роком. Техніка стрімко застаріває, їй на зміну приходять більш потужні, більш сучасні ПК та оргтехніка. Ми радіємо новим моделям персональної техніки з новими можливостями. Ми даруємо один одному сучасні, престижні марки мобільних телефонів. Поступово виникає проблема: а що робити з морально застарілою технікою або тією, яка з тих чи інших причин вийшла з ладу. Виникає поняття "електронне сміття", про обсяги якого ми можемо тільки здогадуватися. Екологи б'ють на сполох, загрожують санкціями провідним виробникам електроніки, якщо ті не приймуть заходів з утилізації персональної та іншої техніки. Деякі виробники на це реагують, деякі - ні.

США, Європа, Японія починають задихатися від електронного сміття. В цих країнах накопичилось понад сотню мільйонів одних тільки кінескопів і моніторів.

Нижче наведена таблиця, де зазначені орієнтовні матеріали, з яких складається ПК, представлений монітором, системним блоком, клавіатурою та мишею.

Таблиця 6.1 – Орієнтовані матеріали, які входять до складу ПК

Благородні метали		Кольорові метали			Полімери і скло	
Au, г	Ag, г	Al, кг	Cu, кг	Fe, кг	ABS, кг	Скло, кг
0,05 – 0,09	0,8 – 1.1	0,1 – 0,4	0,1 – 0,1	4 – 4	4 – 3, 5	10 – 20

Крім того, звичайний персональний комп'ютер складається як з цінних металів - міді, срібла і золота, так і з небезпечних матеріалів - кадмію, свинцю, цинку, нікелю, ртуті. До них додаються пластмаси, індикатори, монітори на рідких кристалах, батареї - всього понад 90 компонентів. Якщо почувши про цінні метали ви спокусились ідеєю заробити на електронному смітті, то одразу вас розчаруємо - вартість переробки і вилучення цінних металів перевищує вартість самих металів. Але утилізувати електронне сміття в будь-якому випадку треба, так що людство все одно приречене на ці витрати [20].

Утилізація оргтехніки фізичними особами відбувається дуже прозаїчно - викинув на смітник у гіршому випадку, в кращому - розібрав на частини для подальшого застосування, що в решті решт закінчується тим ж звалищем. Цей процес в даний час не хвилює ні більшу частину населення, ні державу. Утилізацією в цьому випадку займаються шукачі кольорових і чорних металів.

З юридичними особами все набагато складніше. Відповідно до законодавства України персональні комп'ютери відносяться до основних засобів і підлягають бухгалтерському обліку на підприємстві із зазначенням кількості дорогоцінних металів, які в них містяться. Мало того, на цю техніку

поширюється правило про амортизацію протягом 4 років. Іншими словами, списати і утилізувати дане обладнання можна тільки через 4 роки.

Однак ми знаємо, що персональна техніка морально застаріває набагато раніше. Це пов'язано зі стрімким розвитком електроніки та програмного забезпечення. У такому випадку юридична особа повинна звернутися до спеціалізованих підприємств, які займаються виробництвом, ремонтом і обслуговуванням персональної техніки для проведення технічної експертизи обладнання з отриманням висновку про те, що персональна техніка морально застаріла, знята з виробництва, ремонтна база відсутня, підлягає списанню та утилізації в встановленому порядку.

Тільки після цього необхідно укласти договір з підприємством, яке займається роботою з відходами і має відповідну ліцензію державного органу.

В Україні робота з відходами регламентується законодавством. Підприємствам при роботі з відходами, що містять [20]:

- 1) дорогоцінні метали - необхідна ліцензія Міністерства фінансів України;
- 2) кольорові метали - ліцензія Міністерства промислової політики України;
- 3) чорні метали - ліцензія Міністерства промислової політики України;
- 4) полімери - ліцензія Міністерства екології України;
- 5) екологічно небезпечні відходи - ліцензія Міністерства екології України.

Персональний комп'ютер містить всі види відходів. Тому здавати їх слід на підприємства, які мають ліцензію на роботу з дорогоцінними металами. Ці підприємства-переробники дорогоцінних металів, у свою чергу, укладають договори з підприємствами, що мають інші ліцензії і таким чином персональна техніка розбирається і утилізується згідно з законодавством України.

В кінцевому результаті виконаної роботи можна стверджувати, що вдосконалення сучасних інформаційних технологій, слід направляти не тільки для того щоб створювати людині максимально комфортні умови життя теперішнього часу. Але і для досягнення безвідходного процесу

утилізації відпрацьованої техніки, не завдаючи шкоду навколишньому середовищу.

ВИСНОВКИ

Підсумовуючи підсумки роботи впевнено можна стверджувати що розробка математичної моделі для підвищення достовірності оцінювання трудомісткості тестування застосунків, що створюються з використанням фреймворку Yii1 та розробка відповідного програмного забезпечення є успішною.

Вирішено такі завдання:

- проаналізовано існуючі моделі та методи оцінювання трудомісткості тестування застосунків, що створюються з використанням фреймворку Yii1,
- сформульовано постановку задачі;
- побудовано математичну модель для оцінювання трудомісткості тестування застосунків, що створюються з використанням фреймворку Yii1;
- розроблено програмне забезпечення для оцінювання трудомісткості Yii1 доданків, використовуючи розроблену математичну модель.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Авдєєва Т. В. Вдосконалення нелінійної регресійної моделі для оцінювання трудомісткості тестування web-проектів на основі перетворення Джонсона / Т. В. Авдєєва, С. Б. Приходько [Електронний ресурс]. – Режим доступу: https://atl.pdpu.edu.ua/doc/content/2016/inf/Zbirka_tez_IIS_T-2016.pdf#page=9
2. Prykhodko S Mathematical Modeling of Effort of Mobile Application Development in a Planning Phase. / S Prykhodko, N Prykhodko, K Knyrik, A Pukhalevych // ICTES, 96-105. [Електронний ресурс]. – Режим доступу: <http://ceur-ws.org/Vol-2516/paper7.pdf>
3. IBM. Engineering Lifecycle Management. Estimating the size of test efforts. [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/docs/ru/elm/7.0.1?topic=plans-estimating-size-test-efforts>
4. Ponomarenko T., Gelenko Y., Kotula V. SOFTWARE APPLICATION FOR BUILDING A REGRESSION MODEL OF ESTIMATING THE PHP WEB APPLICATIONS EFFORT // Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2021): Матеріали II Всеукраїнської науково-практичної інтернет конференції (26-28 жовтня 2021 р.). – Миколаїв: НУК імені адмірала Макарова, 2021. – 70 с. [Електронний ресурс]. – Режим доступу: <http://itconf.nuos.edu.ua/2021/proceeding>
5. Tan H.B.K. Estimating LOC for information systems from their conceptual data models / H. B. K. Tan, Y. Zhao, H. Zhang // Proceedings of the 28th International Conference on Software Engineering (ICSE '06), Shanghai, China, May 20-28, 2006. – P. 321-330.
6. Tan H.B.K. Conceptual data model-based software size estimation for information systems / H. B. K. Tan, Y. Zhao, H. Zhang // Transactions on Software

Engineering and Methodology. – 2009. – Vol. 19. – Issue 2. – October 2009. – Article No. 4. DOI: 10.1145/1134285.1134331

7. Prykhodko S.B. Constructing the non-linear regression equation to estimate the software size of open source PHPbased information systems / S. B. Prykhodko, N. V. Prykhodko, T. G. Smykodub, A. V. Spinov // Проблеми інформаційних технологій. – 2018. – № 1 (023). – С.118-125. – ISSN 1998-7005

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

9. Boehm B. Software engineering economics / B. Boehm. – New Jersey: Prentice-Hall, 1981. – 42 p.

10. Boehm B. Software Cost Estimation with Cocomo II. New Jersey, Prentice-Hall. 2000. 544 p

11. Титов А. И. Выбор метрики размера проекта в модели оценки трудоемкости разработки программ / Титов А. И. // Intellectual Technologies on Transport. – 2016. – №1. –С.31-37.

12. Prykhodko N. V. The non-linear regression model to estimate the software size of open source java-based systems / N. V. Prykhodko, S. B. Prykhodko // Радіоелектроніка, інформатика, управління. - 2018. - № 3. - С. 158-166. - Режим доступу: http://nbuv.gov.ua/UJRN/riu_2018_3_19

13. Грешилов, А. А. Математические методы построения прогнозов [Текст] / А. А. Грешилов, В. А. Стакун, А. А. Стакун. – М.: Радио и связь, 1997. – 112 с.

14. Демиденко, Е.З . Линейная и нелинейная регрессии [Текст] / Е. З. Демиденко. – М.: Финансы и статистика, 1981. – 302 с.

15. Bates, Douglas M. Nonlinear Regression Analysis and Its Applications [Text] / Douglas M. Bates, Donald G. Watts. – Wiley, 1988. – 384 p.

16. Pardoe, Iain Applied regression modeling [Text] / Iain Pardoe. – Wiley, 2012. – 325 p.
17. Seber, George A. F. Nonlinear Regression [Text] / George A. F. Seber, C. J. Wild. – John Wiley & Sons, Inc., 2003. – 792 p.
18. Yan, Xin Linear regression analysis: theory and computing [Text] / Xin Yan, Xiao Gang Su. – Singapore: World Scientific Publishing Co. Pte. Ltd., 2009. – 328 p.
19. Айвазян, С. А. Прикладная статистика. Основы эконометрики: Учебник для вузов [Текст]: В 2 т. 2-е изд., испр. – Т. 1: Теория вероятностей и прикладная статистика / С. А. Айвазян, В. С. Мхитарян. – М.: ЮНИТИ-ДАНА, 2001. – 656 с.
20. Кобзарь, А. И. Прикладная математическая статистика. Для инженеров и научных работников [Текст] / А. И. Кобзарь. – М.: ФИЗМАТЛИТ, 2006. – 816 с.
21. Chatterjee, Samprit Handbook of Regression Analysis [Text] / Samprit Chatterjee, Jeffrey S. Simonoff. – Wiley, 2012. – 240 p.
22. Приходько, С. Б. Інтервальне оцінювання статистичних моментів негаусівських випадкових величин на основі нормалізуючих перетворень [Текст] / С. Б. Приходько // Математичне моделювання: науковий журнал. – Дніпродзержинськ: ДДТУ. – 2011. – № 1 (24). – С. 9–13.
23. Приходько, С. Б. Метод побудови нелінійних рівнянь регресії на основі нормалізуючих перетворень [Текст] : тези доп. міждерж. наук.-методич. конф. / С. Б. Приходько // Проблеми математичного моделювання. – Дніпродзержинськ: ДДТУ, 2012. – С. 31–33.
24. Ryan, Thomas P. Modern Regression Methods [Text] / Thomas P. Ryan – Wiley, 2008. – 672 p. 31 Математика и кибернетика - прикладные аспекты
25. Приходько, С. Б. Розробка нелінійної регресійної моделі тривалості програмних проєктів на основі нормалізуючого перетворення

Джонсона [Текст] / С. Б. Приходько, А. В. Пухалевич // Радіоелектронні і комп'ютерні системи. – – 2012. – № 4 (56). – С. 90–93.

26. Приходько, С. Б. Определение доверительных интервалов статистических моментов времени наработки между отказами устройств терминальной сети [Текст] / С. Б. Приходько, Л. Н. Макарова // Наукові праці: науково-методичний журнал. Комп'ютерні технології. – 2013. – Вип. 201, Т. 213. – С. 82–86.

27. Moulla D. K. Duration Estimation Models for Open Source Software Projects [Текст] / Donatien Koulla Moulla, Alain Abran, Kolyang / // I.J. Information Technology and Computer Science, 2021, 1, 1-17 Published Online February 2021 in MECS (<http://www.mecs-press.org/>) DOI: 10.5815/ijitcs.2021.01.01

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

29. Жидецький В. Ц. Охорона праці користувачів комп'ютерів / В.Ц. Жидецький. – Львів: Афіша, 2000. – 176 с.

30. Безпека людини у життєвому середовищі: Навч. посібник / Голінько В.І., Шибка М.В., Безщасний О.В. За ред. В.І.Голінька.– 3-є вид., перероб. і доп. – Д.: Національний гірничий університет, 2004. – 187 с.

31. Безпека людини у надзвичайних ситуаціях: Навч. посібник / В.І.Голінько, С.О.Алексєнко, М.Ф.Кременчуцький та ін.; За ред. В.І.Голінька. – 3-є вид., перероб. і доп. – Д.: Національний гірничий університет, 2004. – 160 с.

32. Безопасность производственных процессов: Справочник / С. В. Белов, В.Н. Бринза, Б.С. Векшин и др.; Под общ. ред. С.В. Белова. – М.: Машиностроение, 1985. - 448 с.

33. Закон України "Про охорону навколишнього природного середовища" N 1264-XII від 25 червня 1991 року.

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Назва проекту: ПЗ оцінювання трудомісткості тестування Yii1-проектів

Умовне позначення програмного комплексу «Yii Estimator1»

Розробник Геленко Юрій

Підстави для розробки

«Yii Estimator1» розробляється на підставі завдання на магістерську роботу “ Математична модель оцінювання трудомісткості тестування Yii1-проектів та розробка програми для її реалізації” затвердженого наказом № 1239-уч від 13.10.2021 р., що видане кафедрою ПЗАС НУК ім. адмірала Макарова.

2. Призначення розробки

2.1. Функціональне призначення

Функціональним призначенням розробки є оцінювання трудомісткості тестування Yii1-проектів.

2.2. Експлуатаційне призначення.

Сервіс повинний бути доступний як веб додаток.

3. Вимоги до програмного продукту

3.1. Вимоги до функціональних характеристик

3.1.1. Вимоги до складу виконуваних функцій

Користувачам програмного комплексу мають бути доступні функції:

- ведення інформації про метрики Yii1 додатку, трудомісткість тестування якого буде оцінена;
- перегляд звіту оцінки трудомісткості тестування додатку.

3.2. Вимоги до надійності

Передбачити обробку ситуацій некоректного вводу інформації з інформуванням користувача про шляхи усунення.

3.3. Умови експлуатації

Умови експлуатації серверної частини згідно з ASHRAE: - рекомендована температура в приміщенні 18 - 27 ° С, для цього необхідно кондиціонування повітря; - вологість повітря в серверній повинна бути в межах від 20% до 80% без конденсації вологи; швидкість зміни вологості 6% в годину; 70 - запиленість не повинна перевищувати 0,75 мг / м³; - тиск в серверній повинен перевищувати тиск в сусідніх приміщеннях. Рекомендується перевищення тиску не менше 14.7 Па.; - рівень освітлення має становити не менше 500 лк, вимірюваному на висоті 1 метр в горизонтальній площині; - рівень електромагнітного випромінювання не повинні перевищувати 3 В / м в усіх діапазонах частот; - для певних видів обладнання і кресів необхідно обмежити вібрацію.

3.4. Вимоги до складу та параметрів технічних засобів.

3.5 Вимоги до інформаційної та програмної сумісності

Браузер Chrome, FireFox, Opera.

4 Вимоги до програмної документації

Програмна документація повинна мати у своєму складі наступне:

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

Розробка ведеться у навчальних цілях, техніко-економічні показники не розраховуються.

5 Стадії та етапи розробки

Стадії та етапи розробки наведені в таблиці А.1.

Таблиця А.1 — Стадії та етапи розробки

Стадії розробки	Етапи робіт	Зміст робіт	Строк виконання
1	2	3	4
1. Технічне завдання	Обґрунтування необхідності розробки програми	Постановка задачі Збір матеріалу Вибір й обґрунтування критеріїв ефективності та якості програми, що розроблюється.	02.09. 2021
	Розробка й затвердження технічного завдання	Визначення вимог до програми. Визначення стадій, етапів й строків розробки програми та документації на неї. Вибір мови програмування. Узгодження й затвердження технічного завдання.	02.10. 2021
2. Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних. Уточнення методів розв'язку задачі. Розробка загального опису алгоритму розв'язку задачі	16.10. 2021
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження й затвердження ескізного проекту.	23.10. 2021
3. Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних. Розробка алгоритму розв'язання задачі. Визначення форми представлення вхідних та вихідних даних. Визначення семантики та синтаксису мови. Розробка структури програми. Остаточне визначення конфігурації технічних засобів.	30.10. 2021
	Затвердження технічного проекту	Розробка плану заходів з розробки та впровадження програм. Розробка пояснювальної записки. Узгодження та затвердження технічного проекту.	06.11. 2021

Продовження таблиці А.1			
4. Робочий проект	Розробка програми	Програмування та налагодження програми.	17.11.2021
	Розробка програмної документації	Розробка програмних документів	25.11.2021
	Випробування програми	Розробка, узгодження та затвердження порядку та методики випробувань. Проведення випробувань. Коректування програми та програмної документації за результатами випробувань.	01.12.2021
5. Впровадження	Підготовка та передача програми.	Підготовка та передача програми та програмної документації для супроводження та (або) виготовлення. Оформлення та затвердження акту про передачу програми на супроводження та (або) виготовлення.	15.12.2021

6. Порядок прийому та контролю

Для контролю та прийому повинен бути наданий опис програмного продукту, а також сам програмний продукт і методика випробування. Порядок контролю та прийому даної розробки здійснюється представником розробника згідно з програмою та методикою випробувань. Якщо програма не пройшла випробування, виконавець зобов'язаний виправити помилки та недоліки у строк, не більш ніж один місяць від дня випробування. За результатами прийому складається акт, який підписується представником замовника та представником розробника та утверджується керівниками організації-замовника і організації розробника. У випадку виявлення помилок під час прийому програмного продукту складаються акт про виявлені помилки, який підписується представниками замовника та розробника і утверджується керівниками організації – замовника й організації – розробника. Розробник повинен у термін, який становить не більше як один місяць виправити вказані зауваження й повідомити замовника про повторне проведення перевірки, не пізніше як за два тижні до початку прийому програмного продукту

ДОДАТОК Б – ТЕКСТ ПРОГРАМИ

Лістинг для виконання в Google Colaboratory

```
import pandas as pd
import numpy as np

dataframe = pd.read_csv('QA_effort.csv')

#independent variables
x = dataframe.iloc[:,4].values

#dependent variable
y = dataframe.iloc[:,4].values

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer

ct = make_column_transformer(
                (OneHotEncoder(),[3]),
                remainder="passthrough"
            )

x=ct.fit_transform(x)

x

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2, random_state=5)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
regressor.coef_
regressor.intercept_

y_pre=regressor.predict(x_test)
y_pre
```

```
from sklearn.metrics import mean_squared_error, r2_score  
r2_score(y_test,y_pre)
```

ДОДАТОК В – ПРОГРАМА І МЕТОДИКА ВИПРОБОВУВАНЬ

Програма і методика випробувань програмного забезпечення для оцінювання трудомісткості тестування Yii1-проектів

1 Об'єкт випробувань

Об'єктом випробувань є програмне забезпечення для оцінювання трудомісткості тестування Yii1-проектів.

2 Мета випробувань

Перевірка функціональних можливостей програмного забезпечення на відповідність вимогам технічного завдання.

3 Вимоги до програми

В ході тестування програмного продукту необхідно перевірити, що розроблене ПЗ реалізує наступні функції:

- внесення даних про проекти;
- підрахунок довірчого інтервалу для нормалізованих даних;
- підрахунок інтервалу передбачення для нормалізованих даних;
- отримання нелінійного регресійного рівняння за допомогою зворотного перетворення;
- підрахунок довірчого інтервалу для нелінійного рівняння;
- підрахунок інтервалу передбачення для нелінійного рівняння.

4 Вимоги до програмної документації

Склад програмної документації, що надається на випробування:

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

5 Засоби і порядок випробувань

Випробування проводилось на

- перевірити можливості введення значень метрик у відповідні комірки;
- перевірити можливість продивитися довірчі інтервали та інтервали передбачення;
- перевірити наявність метрик якості;

6 Методика випробувань

- перевірити валідацію даних при введенні
- перевірити, що результати розрахунку середньої трудомісткості тестування, лівої і правої границі довірчого і прогностного інтервалу з'являються на екрані у вигляді рядку.

ДОДАТОК В – ОПИС ПРОГРАМИ

Загальні відомості

Програма для оцінювання трудомісткості тестування Yii1-проектів

Функціональне призначення

Програма призначена для оцінювання трудомісткості тестування Yii1-проектів.

Технічні засоби, що використовуються

Програма має запускатись у середовищі Google Colab, тому вимог до обладнання не висувається.

Виклик та завантаження

Для запуску та роботи з програмним забезпеченням необхідний браузер Chrome , Mozilla або FireFox

Завантаження програми здійснюється шляхом відкриття файлу

CakePhpSizeEst.py.

ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА

Введіть значення кількості помилок в якості X1 та значення комплексності отриманої phpmetrics в якості X2 та побачите значення трудомісткості в людино-годинах, довірчий та прогнозний інтервал трудомісткості тестування.

YIII QA Effort

ГОЛОВНА ПОСЛУГИ ЗВ'ЯЗАТИСЬ З НАМИ

ВІТАЄМО

What will be the QA Effort?

$$Y=6.75+0.007X1+0.04x2$$

x1=10

x2=72

16.5

dov_int(15-29)

progn_int(14.5-32)

ДОВІРЧА ВІРОГІДНІСТЬ 95 % R2 0,837 MMRE 0,027

your data (xlsx)

СКАЧАТЬ

Додаток Е. Прогнозні та довірчі інтервали

ліва границя доверит інтервала	права границя доверит інтервала	left prediction interval	right prediction interval
8,00	20,15781531	7,48846657	20,66959226
6,61	21,54653891	6,34869714	21,80936168
7,79	20,3710402	7,33678861	20,82127022
8,41	19,74795226	7,73234128	20,42571754
7,32	20,84286102	6,96308681	21,19497202
7,58	20,57434148	7,18111758	20,97694125
7,92	20,2403905	7,43134293	20,72671589
7,40	20,75336472	7,03709164	21,12096719
7,50	20,65755613	7,1148864	21,04317242
7,82	20,34169791	7,35843565	20,79962318
7,79	20,3710402	7,33678861	20,82127022
8,00	20,15781531	7,48846657	20,66959226
7,85	20,31035	7,38130695	20,77675187
8,23	19,92787619	7,63450457	20,52355425
8,05	20,11019641	7,52037084	20,63768798
8,00	20,15781531	7,48846657	20,66959226
8,23	19,92787619	7,63450457	20,52355425
8,16	19,99698578	7,59284132	20,5652175
7,88	20,27670221	7,40555061	20,75250821
7,76	20,39861804	7,31624034	20,84181848
8,00	20,15781531	7,48846657	20,66959226
7,57	20,59223579	7,16698739	20,99107143
7,73	20,42463162	7,29668364	20,86137518
8,41	19,74795226	7,73234128	20,42571754
7,82	20,34169791	7,35843565	20,79962318
7,54	20,62077782	7,14432031	21,01373851
7,83	20,33000552	7,36699776	20,79106106
7,72	20,4413191	7,28405188	20,87400694
7,28	20,88097405	6,93121293	21,22684589
7,30	20,85405597	6,9537458	21,20431302
7,23	20,9301811	6,88976613	21,2682927
7,57	20,58658519	7,17145619	20,98660263
8,43	19,73130069	7,74053437	20,41752445
7,58	20,57532581	7,18034195	20,97771687

ліва границя доверит інтервала	права границя доверит інтервала
13,49	32,72
10,31	35,90
10,32	35,89
11,59	34,62
13,08	33,13
13,53	32,68
13,56	32,65
11,50	34,71
11,33	34,88
8,61	37,60
13,65	32,56
13,99	32,21
11,59	34,62
13,08	33,13
13,91	32,30
13,99	32,21
13,08	33,13
13,48	32,73
10,26	35,95
13,53	32,68
13,99	32,21
12,22	33,99
13,91	32,30
13,75	32,46
11,59	34,62
12,48	33,73
14,02	32,19
12,48	33,73
10,26	35,95
11,59	34,62
13,08	33,13
0,48	45,72
7,48	38,73
-12,98	59,19

10	71
209	99
16	80
4	80
45	181
25	330
12	110
37	210
30	1460
15	605
16	688
10	215
14	255
6	462
9	448
10	776
6	200
7	132
13	36
17	770
10	140
26	810
18	635
4	150
15	800

M: 23.36 M: 389.72

Розрахунок багатфакторної моделі

Sum of $X_1 = 584$

Sum of $X_2 = 9743$

Sum of $Y = 551.41$

Mean $X_1 = 23.36$

Mean $X_2 = 389.72$

Mean $Y = 22.0564$

Sum of squares (SS_{X1}) = 38351.76

Sum of squares (SS_{X2}) = 2920049.04

Sum of products (SP_{X1Y}) = -819.9876

Sum of products (SP_{X2Y}) = 109937.1148

Sum of products (SP_{X1X2}) = -32771.48

Regression Equation = $\hat{y} = b_1X_1 + b_2X_2 + a$

$b_1 = ((SP_{X1Y})*(SS_{X2})-(SP_{X1X2})*(SP_{X2Y})) / ((SS_{X1})*(SS_{X2})-(SP_{X1X2})*(SP_{X1X2})) =$
 $1208397954.73/110915050068.92 = 0.01089$

$$b_2 = ((SP_{X_2Y}) * (SS_{X_1}) - (SP_{X_1X_2}) * (SP_{X_1Y})) / ((SS_{X_1}) * (SS_{X_2}) - (SP_{X_1X_2}) * (SP_{X_1X_2})) = 4189409634.67 / 110915050068.92 = 0.03777$$

$$a = M_Y - b_1M_{X_1} - b_2M_{X_2} = 22.06 - (0.01 * 23.36) - (0.04 * 389.72) = 7.08165$$

$$\hat{y} = 0.01089X_1 + 0.03777X_2 + 7.08165$$

X1-M _{x1}	X2-M _{x2}	Y-M _y	(X1-M _{x1}) ²	(X2-M _{x2}) ²	SP _{x1y}	SP _{x2y}	SP _{x1x2}
-13.3	-318.72	-5.376	178.49	101582.	71.829	1713.56	4258.09
6	-290.72	-10.556	34462.2	438	-1959.6	6	9
185.6	-309.72	-10.026	1	84518.1	9	3068.95	-53969.
4	-309.72	-7.176	54.17	18	73.794	7	261
-7.36	-208.72	-8.306	374.81	95926.4	138.935	3105.37	2279.53
-19.3	-59.72	-3.946	468.29	78	-179.75	7	9
6	-279.72	-14.056	2.69	95926.4	-6.472	2222.67	5996.17
21.64	-179.72	-4.226	129.05	78	159.681	5	9
1.64	1070.28	57.184	186.05	43564.0	-57.648	1733.71	-4516.7
-11.3	215.28	-0.556	44.09	38	379.699	2	01
6	298.28	18.274	69.89	3566.47	4.652	235.679	-97.941
13.64	-174.72	-1.056	54.17	8	-134.49	3931.85	3177.61
6.64	-134.72	-8.556	178.49	78243.2	4	6	9
-8.36	72.28	-2.306	87.61	78	14.114	759.569	-2451.3
-7.36	58.28	1.944	301.37	32299.2	80.088	61202.4	81
-13.3	386.28	6.944	206.21	78	40.039	63	7106.65
6	-189.72	-6.706	178.49	1145499	-27.91	-119.78	9
-9.36	-257.72	-3.056	301.37	.278	-92.766	2	-1799.7
-17.3	-353.72	-12.556	267.65	46345.4	116.423	5450.64	41
6	380.28	13.044	107.33	78	50.003	9	-2195.3
-14.3	-249.72	-4.156	40.45	88970.9	130.084	184.574	41