

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Інформаційних управляючих систем  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Дослідження методів оцінювання зусиль ІТ-проекту створення інформаційної системи в умовах ре-використання її функцій  
(тема)

Виконав:

студент 2 курсу, групи УПГІТм-22-3

Наконечний Артем Володимирович  
(прізвище, ім'я, по батькові)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій  
(повна назва освітньої програми)

Керівник проф.каф. ІУС Максим ЄВЛАНОВ  
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри

  
(підпис)

Костянтин ПЕТРОВ  
(власне ім'я, прізвище)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра Інформаційних управляючих системРівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки

(код і повна назва)

Тип програми освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма Управління проектами в галузі інформаційних технологій

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 

(підпис)

« 01 » квітня 20 24 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Наконечному Артему Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів оцінювання зусиль ІТ-проекту створення інформаційної системи в умовах ре-використання її функційзатверджена наказом університету від 01 квітня 2024 р. № 258 Ст2. Термін подання студентом роботи до екзаменаційної комісії 03 червня 2024 р.3. Вихідні дані до роботи Науково-технічна література, публікації, інформація з інтернетресурсів щодо класичного методу функціональних точок та його повторного використання, методи оцінювання ІТ-проектів.4. Перелік питань, що потрібно опрацювати в роботі Дослідження методу Функціональних точок, адаптація методу функціональних точок, повторне використання методу, аналіз існуючих методів оцінювання, алгоритм класичного методу функціональних точок

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз особливостей процесу оцінювання та контролю проекту	01.04.24 – 06.04.24	Виконано
2	Аналіз існуючих методів оцінювання зусиль ІТ-проектів створення інформаційних систем	06.04.24 – 11.04.24	Виконано
3	Аналіз підходів до удосконалення методу аналізу функціональних точок	11.04.24 – 16.04.24	Виконано
4	Опис методу аналізу функціональних точок	16.04.24 – 20.04.24	Виконано
5	Опис теоретичної основи для модифікації методу аналізу функціональних точок	20.04.24 – 27.04.24	Виконано
6	Модифікація методу аналізу функціональних точок	27.04.24 – 01.05.24	Виконано
7	Опис схеми алгоритма класичного методу функціональних точок	01.05.24 – 07.05.24	Виконано
8	Опис методики використання класичного методу функціональних точок	07.05.24 – 11.05.24	Виконано
9	Опис ІТ-проекту створення функціонального завдання в умовах ре-використання окремих функцій інформаційної системи	11.05.24 – 16.06.24	Виконано
10	Апробація модифікованого методу аналізу функціональних точок	16.05.24 – 23.05.24	Виконано
11	Попередній захист кваліфікаційної роботи	03.06.24	Виконано
12	Захист кваліфікаційної роботи в екзаменаційній комісії	05.06.24	Виконано

Дата видачі завдання 01 квітня 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф.каф. ІУС Євланов М.В.  
(підпис) (посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 88 с., 15 табл., 6 рис., 1 дод., 23 джерел.

АДАПТАЦІЯ МЕТОДУ ФУНКЦІОНАЛЬНИХ ТОЧОК, АЛГОРИТМ КЛАСИЧНОГО МЕТОДУ ФУНКЦІОНАЛЬНИХ ТОЧОК, АПРОБАЦІЯ МЕТОДУ ФУНКЦІОНАЛЬНИХ ТОЧОК, МЕТОД ФУНКЦІОНАЛЬНИХ ТОЧОК, ПОВТОРНЕ ВИКОРИСТАННЯ МЕТОДУ.

Об'єкт дослідження – класичний метод функціональних точок.

Мета кваліфікаційної роботи – дослідження використання методу функціональних точок в ІТ – проектах на методології їх повторного використання.

Метод дослідження – впровадження повторного використання методу функціональних точок та

У роботі проведено аналіз класичного методу функціональних точок та його повторне використання, розрахунок застосування в ІТ – проектах та відмінності в застосуванні класичного методу функціональних точок та його повторне використання.

Даний метод може використовуватися в ІТ – проектах, що значно покраще результати.

## ABSTRACT

Explanatory note to the attestation work: 88 pages, 15 tables, 6 figures, 1 appendices, 23 sources.

ADAPTATION OF THE METHOD OF FUNCTION POINTS, ALGORITHM OF THE CLASSICAL METHOD OF FUNCTION POINTS, APPROVAL OF THE METHOD OF FUNCTION POINTS, METHOD OF FUNCTION POINTS, REUSE OF THE METHOD.

The object of research is the classical method of functional points.

The purpose of the certification work is to study the use of the method of functional points in IT projects based on the methodology of their reuse.

The research method is the implementation of the reuse of the method of functional points and

The paper analyzes the classical method of functional points and its reuse, the calculation of its application in IT projects and the differences in the application of the classical method of functional points and its reuse.

This method can be used in IT projects, which will significantly improve the results.

## ЗМІСТ

Скорочення та умовні позначки .....	7
Вступ.....	8
1 Аналіз існуючих методів оцінювання зусиль іт-проектів створення інформаційних систем в умовах повторного використання їх елементів .....	10
1.1 Аналіз особливостей процесу оцінювання та контролю проекту .....	10
1.2 Аналіз існуючих методів оцінювання зусиль ІТ-проектів створення інформаційних систем.....	14
1.3 Аналіз підходів до удосконалення методу аналізу функціональних точок .....	17
2 Формування вимог до задачі, що розробляється .....	21
2.1 Опис методу аналізу функціональних точок.....	21
2.2 Опис теоретичної основи для модифікації методу аналізу функціональних точок.....	27
2.3 Модифікація методу аналізу функціональних точок .....	29
3 Аналіз особливостей застосування методу Функціональних точок.....	35
3.1.Схема алгоритма класичного методу функціональних точок .....	35
3.2. Методика використання класичного методу функціональних точок .....	36
4.Апробація модифікованого методу функціональних точок .....	47
4.1 “Загальні відомості про іт проект розробки функціональної задачі. Формування та ведення індивідуального плану науково-педагогічного працівника кафедри” Інформаційно-аналітичної системи Університет. ....	47
4.2 Апробація модифікованого методу аналізу функціональних точок.....	53
4.3 Аналіз переваг і недоліків методу .....	58
Висновки .....	64
Перелік джерел посилання .....	66
Додаток А Графічний матеріал.....	69

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПЗ – програмне забезпечення

Function Point Analysis) - метод аналізу функціональних точок

FPA IFPUG — метод функціональних точок

ІС – інформаційна система

PERT (Program Evaluation and Review Technique) - методика оцінювання й аналізу проектів

## ВСТУП

Проблема підвищення точності оцінювання параметрів ІТ-проектів залишається однією з найскладніших у даній області. Основні групи методів оцінювання параметрів ІТ-проектів були визначені ще на початку 2000-х років і з того часу кардинально не змінювалися [1, 2]. Практично не змінюються і діапазони точності оцінювання трудовитрат і бюджету ІТ-проектів [3].

Крім того, особливості застосування моделей і методів оцінювання параметрів ІТ-проекту в значній мірі залежать від виду результату цього проекту. Слід визнати, що переважна більшість подібних моделей і методів спрямовані на оцінювання параметрів програмних ІТ-продуктів. Проектування та розробка таких ІТ-продуктів передбачає, переважно, роботи зі створення програмного забезпечення. Однак застосування подібних моделей і методів для оцінювання параметрів ІТ-проектів створення інформаційних систем (ІС) управління підприємствами та організаціями викликає певні труднощі.

Одним з методів, найбільш придатних для оцінювання параметрів ІТ-проекту створення ІС, є метод аналізу функціональних точок (FPA). Цей метод дозволяє виконувати оцінювання на основі аналізу представлень окремих функцій ІС у вигляді сукупностей структурних одиниць різного призначення. Подібна інформація дозволяє кількісно оцінити як обсяг робіт, необхідних для реалізації конкретної функції ІС, так і обсяг робіт, необхідних для реалізації інтерфейсів цієї функції з іншими функціями ІС, базою даних системи та користувачами. Хоча FPA був розроблений достатньо давно, він до сих пір є основою для досліджень у галузі Software Effort Estimation [4].

Однак FPA не позбавлений недоліків. Одним з таких недоліків є його слабка адаптація до сучасних підходів до проектування та розробки ІТ-продуктів. Цей недолік можна пояснити тим, що метод функціональних точок базується на уявленні окремих функцій ІС як унікальних елементів створюваної системи. Згідно з таким уявленням оцінювання кожної конкретної функції ІС має

здійснюватися окремо навіть у випадках створення цієї функції на основі раніше реалізованих функцій. Застосування методу FPA для оцінювання ІТ-проекту створення ІС в умовах повторного використання окремих функцій системи буде давати значно завищені результати. Тому проведення досліджень, що дозволять адаптувати метод FPA до особливостей сучасних підходів до створення ІС, є актуальним з теоретичної і практичної точок зору.

Метою даної кваліфікаційної роботи є впровадження повторного використання класичного методу функціональних точок в ІТ – проектах.

Задача кваліфікаційної роботи є розробка та адаптація класичного методу функціональних точок до його повторного використання в ІТ - проектах.

Дана кваліфікаційної робота була виконана згідно вимогам ДСТУ

# **1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОЦІНЮВАННЯ ЗУСИЛЬ ІТ-ПРОЕКТІВ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ В УМОВАХ ПОВТОРНОГО ВИКОРИСТАННЯ ЇХ ЕЛЕМЕНТІВ**

## **1.1 Аналіз особливостей процесу оцінювання та контролю проекту**

Інформаційні системи управління проектами стали невід'ємною частиною сучасного бізнесу і відіграють ключову роль у забезпеченні ефективного планування, управління та виконання проектів. Вони забезпечують централізований доступ до інформації, автоматизують процеси управління ресурсами та дозволяють аналізувати результати для прийняття обґрунтованих стратегічних рішень.

Зростаюча роль оцінки проектів у науці управління є наслідком впровадження нових рішень у соціальних, економічних та екологічних сферах. Потреба у впровадженні унікальних проектів у умовах невизначеності фактично веде до теоретичних досліджень і практики, що постійно прагнуть до кращих методів їх оцінки. Ці потреби пов'язані, серед іншого, з оцінкою ресурсів, необхідних для впровадження проектів, та зменшенням ризику досягнення оцінених цілей. Отже, необхідно посилити методи оцінки різних типів проектів, що реалізуються в бізнесі та неприбуткових організаціях. Існує недостатня підтримка прийняття рішень з використанням наразі доступних методів оцінки.

Отже, можна спостерігати наявність дослідницької прогалини щодо потреби та потенціалу наукових рішень у проблемі оцінки з використанням все більш удосконалених методів оцінки проектів. Дослідження, пов'язані з цією областю, визначають нові та захоплюючі напрями розвитку досліджень, а отримані висновки та рекомендації можуть бути цінними джерелами інформації для використання в практиці управління різними проектами в різних типах організацій. Таким чином, є обґрунтування для пошуку нових методів оцінки, і одним із можливих напрямів досліджень може бути вивчення методів на основі аналізу функціональних точок (FPA). Темою цього дослідження є різноманітні

проекти, а об'єктом дослідження є проблема вивчення можливості використання FPA в оцінці проекту. Загальною метою цієї роботи є представлення переваг та недоліків застосування методів на основі FPA в оцінці різних проектів. На основі таких розглядів пропонується модифікована модель FPA.

Мета процесу оцінки та контролю проекту полягає в оцінці відповідності та реалізованості планів, визначенні стану проекту, технічної та процесної продуктивності, та керуванні виконанням для забезпечення виконання згідно з планами та графіками, у межах прогнозованих бюджетів, задоволення технічних об'єктивів. Цей процес періодично та на ключових подіях оцінює прогрес та досягнення у порівнянні з вимогами, планами та загальними бізнес-цілями. Інформація надається для управлінських дій у випадку виявлення значних відхилень. Цей процес також включає перенаправлення активностей та завдань проекту, за необхідності, для виправлення виявлених відхилень і варіацій від інших технічних управлінських або технічних процесів. Перенаправлення може включати перепланування за необхідності.

В результаті успішної реалізації процесу Оцінки та Контролю проекту:

- доступні показники продуктивності або результати оцінки;
- оцінено адекватність ролей, обов'язків, відповідальності та повноважень;
- оцінено достатність ресурсів;
- проведено технічні перегляди прогресу;
- досліджені та проаналізовані відхилення в проектній продуктивності від планів;
- зацікавлені сторони проінформовані про статус проекту;
- визначено та направлено коригуючі дії, коли досягнення проекту не відповідає цілям;
- ініційоване перепланування проекту за необхідності;
- повноваження проекту на продовження (чи ні) від одного запланованого вихідного пункту або події до наступного авторизовано;
- досягнуто об'єктиви проекту.

Проект повинен реалізувати наступні дії та завдання відповідно до політик і процедур організації з урахуванням процесу оцінки та контролю проекту.

Планування оцінки та контролю проекту складається з наступного завдання: Визначити стратегію оцінки та контролю проекту. Стратегія ідентифікує очікувані дії з оцінки та контролю проекту, включаючи заплановані методи оцінки та часові рамки, а також необхідні управлінські та технічні перегляди.

Оцінка проекту складається з наступних завдань:

- оцінка відповідності цілей та планів проекту контексту проекту;
- оцінка управлінських та технічних планів відносно цілей для визначення адекватності та реалізованості;
- оцінка стану проекту та технічного статусу відповідно до відповідних планів для визначення фактичних та прогнозованих відхилень вартості, графіка та продуктивності;
- оцінка адекватності ролей, обов'язків, відповідальності та повноважень. Включає оцінку достатності компетентностей персоналу для виконання проектних ролей та виконання завдань проекту. Об'єктивні показники використовуються у всіх можливих випадках, наприклад, ефективність використання ресурсів, досягнення проекту;
- оцінка адекватності та доступності ресурсів. Ресурси включають інфраструктуру, персонал, фінансування, час або інші важливі елементи. Це завдання включає оцінку можливості повторного використання існуючих процесів та ресурсів інфраструктури, а також підтвердження виконання внутрішньоорганізаційних зобов'язань;
- оцінка прогресу за допомогою вимірювання досягнень та завершення віх. Включає збір та оцінку даних про працю, матеріали, вартість послуг та технічну продуктивність, а також інші технічні дані про об'єктиви, такі як доступність. Ці дані порівнюються з показниками досягнень. Це включає проведення оцінок ефективності для визначення адекватності розвиваючої

програмної системи відносно вимог. Це також включає готовність допоміжних систем надавати свої послуги за потреби;

- проведення необхідних управлінських та технічних переглядів, аудитів та інспекцій. Формальні чи неформальні процедури, які проводяться для визначення готовності переходу до наступного етапу життєвого циклу або події проекту, для забезпечення досягнення проектних та технічних цілей, або для отримання зворотного зв'язку від зацікавлених сторін;

- моніторинг критичних процесів та нових технологій. Включає ідентифікацію та оцінку технологічної зрілості та реалізованості впровадження технологій. Технологічна зрілість - це готовність технології до операційного використання, яка часто вимірюється на шкалі від низької (існує тільки як концепція) до високої (доведена в операційному використанні);

- аналіз результатів вимірювань та формування рекомендацій. Результати вимірювань аналізуються для виявлення відхилень, варіацій або небажаних тенденцій від запланованих значень, включаючи потенційні проблеми, та для формування відповідних рекомендацій щодо виправлень або запобіжних заходів. Це включає, де це можливо, статистичний аналіз показників, що вказують на тенденції, наприклад, щільність дефектів, яка вказує на якість виводів, розподіл вимірюваних параметрів, яка вказує на повторюваність процесу;

- запис та надання статусу та результатів від завдань оцінки. Зазвичай ці завдання визначаються в угоді, політиках та процедурах;

- моніторинг виконання процесу у межах проекту. Включає аналіз показників процесу та огляд тенденцій відносно проектних цілей. Будь-які ідентифіковані заходи для покращення можуть бути вирішені через процес забезпечення якості, процес управління якістю або процес управління моделлю життєвого циклу.

Контроль проекту складається з наступних завдань:

- ініціювати необхідні заходи для вирішення виявлених проблем. Завдання виконується, коли досягнення проекту або технічний результат не

відповідають запланованим цілям. Це включає запобіжні, коригуючі та розв'язання проблем. Заходи, як правило, потребують перепланування або перерозподілу персоналу, інструментів та активів інфраструктури, коли виявлено недостатність або недоступність, або коли досягнення проекту або технічний результат перевищують цілі або плани. Вони часто впливають на вартість, графік або технічний обсяг чи визначення. Деякі заходи можуть потребувати змін у виконанні та виконанні процесів життєвого циклу. Заходи реєструються та перевіряються для підтвердження їх адекватності та своєчасності;

- ініціювати необхідне перепланування проекту. Перепланування проекту ініціюється, коли цілі проекту або обмеження змінилися, або коли припущення планування виявилися недійсними. Будь-яка зміна, яка потребує зміни угоди між замовником і постачальником, викликає процеси придбання та постачання;

- ініціювати змінні заходи, коли змінено договірні питання щодо вартості, часу або якості через вплив запиту від замовника або постачальника. Включає розгляд змінених умов та положень для постачання або ініціювання нового вибору постачальника, що викликає процеси придбання та постачання;

- авторизувати проект на продовження до наступного віхолку або події, якщо це обґрунтовано. Процес оцінки та контролю проекту використовується для досягнення згоди щодо завершення віхолки.

## 1.2 Аналіз існуючих методів оцінювання зусиль ІТ-проектів створення інформаційних систем

Існує багато методів оцінювання зусиль самі використовувани з них це Метод Delphi, Three-point Estimation, Top-down Estimation, Parametric Estimation та метод функціональних точок. Переваги та недоліки кожного відрізняються,

та кожен з цих методів може використовуватися схиляючись на потреби та проблеми котрі будуть чи були винекнені в ході виконання завдання.

Метод Delphi Виник у середині 20-го століття в аналітичному центрі RAND в США. Для оцінки проєктів метод використовує експертний досвід: троє або більше експертів формують свої анонімні оцінки, а після цього об'єднуються для досягнення спільної згоди. Використовується для проєктів з високим рівнем складності, де експертна думка може бути більш точною, ніж статистичні методи. Також якщо доступ до правдивої інформації обмежений, або коли існують суперечності в наявній та коли важливо опитати велику кількість експертів, не залучаючи їх до прямого обговорення.

Three-point Estimation (PERT), або оцінка за трьома точками. Метод PERT (Program Evaluation Review Technique), або оцінка за трьома точками, був розроблений у 1958 році на замовлення спеціального підрозділу ВМС США, щоб створити революційну ракетну систему. Його мета — полегшити створення графіків для складних та масштабних проєктів. Застосовується цей метод коли проєкт має велику кількість завдань, які потрібно виконувати паралельно, у проєкті залучені багато учасників та коли деякі етапи проєкту можуть бути невизначеними або містити ризики а також потрібно оцінити тривалість проєкту, враховуючи різні сценарії його розвитку.

Метод Top-down Estimation «згори вниз» — один з найпоширеніших підходів до оцінки проєкту в ІТ і не тільки. Основою для визначення бюджету проєкту та термінів стає минулий досвід команди, а набуті знання можна ефективно використовувати для розв'язання схожих завдань у майбутньому. Застосовується на ранніх стадіях проєкту, коли детальна інформація відсутня також для великих проєктів, де детальна оцінка вимагає значних ресурсів та коли потрібно отримати швидко оцінку для прийняття рішення.

Parametric Estimation – використовується коли треба спрогнозувати ресурси, потрібні для завершення проєкту, методика Parametric Estimation, або оцінка за параметрами, використовує «історичні дані» (наприклад, інформацію

про те, скільки часу потребує те чи інше завдання або які ресурси необхідні для виконання завдання).

Метод функціональних точок - це стандартний метод вимірювання розміру програмного продукту з точки зору користувачів системи. За допомогою даного методу можна оцінити складність програмного забезпечення на основі логічної моделі даних, а також на основі кількості функціоналу, необхідного замовником і поставляється розробником. Перевагою методу є незалежність виміру від технологічної платформи, на якій буде розроблятися продукт, і єдиноподібний підхід до оцінки проектів у компанії.

Задля покращення методу функціональних точок треба проводити додаткові дослідження для визначення найбільш оптимальних і точних правил та рекомендацій для оцінки функціональних точок. Розробка і адаптація методу FPA для більш широкого спектру проектів і необхідностей. Розробка програмних інструментів або інтеграція з існуючими інструментами, які спрощують та автоматизують процес оцінки функціональних точок. Вдосконалення метрик, що використовуються для оцінки функціональних точок, а також розробка нових метрик для врахування додаткових аспектів проекту, таких як складність, якість та ризику. Розробка програм навчання та сертифікації для спеціалістів з оцінки функціональних точок, які б забезпечували стандартизований підхід до використання методу та забезпечували високу кваліфікацію фахівців у цій області. Розробка та впровадження інтеграції методу FPA з іншими методами оцінки, такими як методи оцінки вартості зусиль, що дозволяє отримати більш повне та об'єктивне уявлення про ресурси, необхідні для проекту.

### 1.3 Аналіз підходів до удосконалення методу аналізу функціональних точок

Сучасне уявлення методу FPA було опубліковано у 1990 році. Розвитком і стандартизацією цього методу займається Міжнародна група користувачів функціональних точок (IFPLUG). Ця організація у 2010 році опублікувала версію 4.3.1 опису даного методу, яка на сьогодні є актуальною.

У 2015 році у [6] був проведений огляд і аналіз вдосконалень, спрямованих на підвищення точності методу FPA, які були запропоновані за попередні 13 років. Під час пошуку відповідних публікацій було виявлено 1600 результатів, з яких для аналізу було обрано 454 первинних дослідження. З цих досліджень питання підвищення точності FPA розглядалися лише у 18 публікаціях. На основі отриманих результатів у [6] був зроблений висновок, що в цілому метод FPA досить вдосконалений. Однак у [6] було відзначено велику кількість проблем, які виникають під час розрахунку функціонального розміру. На підставі цього у [6] був зроблений висновок про необхідність перегляду методу FPA, щоб охопити можливі покращення, запропоновані дослідниками.

Одним із напрямів цих досліджень слід виділити дослідження особливостей застосування методу FPA на ранніх етапах створення ІТ-продуктів. Варто відзначити, що процес застосування стандартного методу FPA (далі - FPA IFPLUG) часто буває занадто тривалим, дорогим або вимагає більше знань, ніж є на момент складання оцінки зусиль. Пропоновані ранні методи оцінки, як правило, спрощені порівняно зі стандартним FPA IFPLUG за рахунок виключення одного або декількох етапів початкового методу. Тому у [7] представлені результати досліджень, що стосуються відносного обсягу зусиль, необхідних на етапах методу FPA IFPLUG. Отримані результати пропонується використовувати для оцінки очікуваної економії, яку забезпечують методи ранньої оцінки. Однак у [7] не враховуються особливості застосування методу FPA IFPLUG для різних методологій проектування ІТ-продукту.

Другим напрямом досліджень у сфері вдосконалення методу FPA IFPLUG є робота з його адаптацією до особливостей конкретних етапів життєвого циклу ІТ-продукту. Так, у [8] запропонована модель FPA-SDP, яка адаптує застосування методу FPA IFPLUG до особливостей етапу розробки програмного забезпечення. За аналогією з моделлю FPA-SDP у [9] була розроблена модель SCCE, що базується на комбінації методу аналізу впливу змін і методу FPA IFPLUG. Ця модель адаптує застосування згаданих методів до особливостей етапу супроводження програмного забезпечення. Отримані результати дозволяють узгодити описи станів програмних артефактів і оцінити фактичний розмір запиту на зміну з урахуванням його рівня складності. Однак застосування цих результатів у процесі опису архітектури ІС та під час визначення функціональної структури ІС ускладнено.

Третім напрямом досліджень є вдосконалення методу FPA IFPLUG за допомогою використання засобів штучного інтелекту. Найбільш ранні роботи в цій області базуються на застосуванні апарату нечіткої логіки. Так, у [10] запропоновано підвищити точність методу FPA IFPLUG за рахунок застосування апарату нечіткої логіки для класифікації окремих функцій системи згідно з їх відносною функціональною складністю. Отримані результати дозволяють стверджувати про підвищення точності оцінок затрат і часу виконання ІТ-проекту в результаті застосування модифікованого методу аналізу нечітких функціональних точок. У [11] розглянуто застосування моделей глибокого навчання для класифікації типів функціональних точок нових вимог до програмного забезпечення в крупно-масштабних ІТ-проектах. Як результат дослідження у [11] запропоновано модель розпізнавання іменованих сутностей (NER) на основі глибокого навчання. Проведені випробування підтвердили значне підвищення точності та ефективності застосування модифікованого методу FPA IFPLUG.

Однак застосування отриманих результатів у практиці вирішення завдання оцінювання зусиль ІТ-проекту з створення ІС на сьогоднішній час є дуже ускладненим. Головним фактором цієї складності є відсутність поширеного ІТ-

продукту, який дозволяє автоматизувати вирішення завдання оцінювання на основі методу функціональних точок. На даний момент створення подібних продуктів є, скоріше, науково-прикладною задачею. Прикладом, що підтверджує це твердження, є програмний продукт, описаний в [12].

Тому окремі дослідження спрямовані на такі способи підвищення точності методу FPA IFPLUG, які не вимагають додаткових складних обчислень та алгоритмів обробки великих обсягів даних. Прикладом таких досліджень є описаний у [13] метод аналізу функціональних точок на основі дерева функціональних точок. Даний метод розроблений відповідно до етапів FPA IFPUG. У [13] представлено також опис прототипу інструментального засобу, який дозволяє автоматизувати запропоновану модифікацію методу.

Іншим прикладом подібних досліджень слід вважати роботу [14]. У ній представлено аналіз впливу окремих факторів (підхід до підрахунку функціональних точок, галузь діяльності, галузь промисловості, відносний розмір) на оцінку трудовитрат за методом FPA IFPUG. Також у [14] вивчаються фактори, які впливають на продуктивність, і можливість оцінки продуктивності методом FPA IFPUG.

Методи, засновані на FPA, використовуються організаціями, які будують ІТ-рішення відповідно до зовнішніх специфікацій закупівлі, а також впроваджують їх у власних організаціях. Для отримувача ІТ-рішення його вартість залежить від рівня задоволення користувачів, якості впровадження стратегії організації та досягнення її запланованих економічних цілей. Цей метод дозволяє оцінити проект не лише на етапі підготовки концепції, а й після його впровадження з метою поліпшення оцінки ресурсів, необхідних для впровадження подальших проектів.

Однак всі розглянуті дослідження мають досить серйозний недолік. Цей недолік полягає в визнанні незмінними всіх припущень, на яких базується метод FPA IFPUG. Одним із таких припущень є припущення про унікальність кожної конкретної функції в межах створюваної ІС. У той же час сучасні методології розробки ІС та інших програмних систем передбачають повторне використання

раніше розроблених функцій. Це призводить до порушення даного припущення. Наслідком цього порушення є надто завищені оцінки зусиль і, відповідно, надто завищені витрати на виконання ІТ-проекту. Тому виникає необхідність адаптації методу FPA IFPUG до особливостей багаторазового використання окремих елементів системи під час проектування та розробки ІС.

## 2 ФОРМУВАННЯ ВИМОГ ДО ЗАДАЧІ, ЩО РОЗРОБЛЯЄТЬСЯ

### 2.1 Опис методу аналізу функціональних точок

В основу дослідження покладено метод FPA IFPUG (Release 4.3.1). Даний метод передбачає виконання таких етапів:

- етап 1 – збирання наявної документації;
- етап 2 – визначення обсягу та меж оцінюваної системи та визначення функціональних вимог користувачів;
- етап 3 - визначення складності оцінюваної функції за даними;
- етап 4 – визначення складності оцінюваної функції з транзакцій;
- етап 5 – розрахунок кількості функціональних точок;
- етап 6 – оформлення та публікація результатів розрахунку кількості функціональних точок.

Схема послідовності виконання етапів методу FPA IFPUG наведена на рис. 1. Детальний опис методу FPA IFPUG наведено в літературі.

У згаданому релізі методу FPA IFPUG передбачено такі варіанти розрахунку функціональних точок:

- розрахунок функціональних точок проекту розробки (Development project function point count);
- розрахунок функціональних точок проекту удосконалення (Enhancement project function point count); в) розрахунок функціональних точок програми (Application function point count).

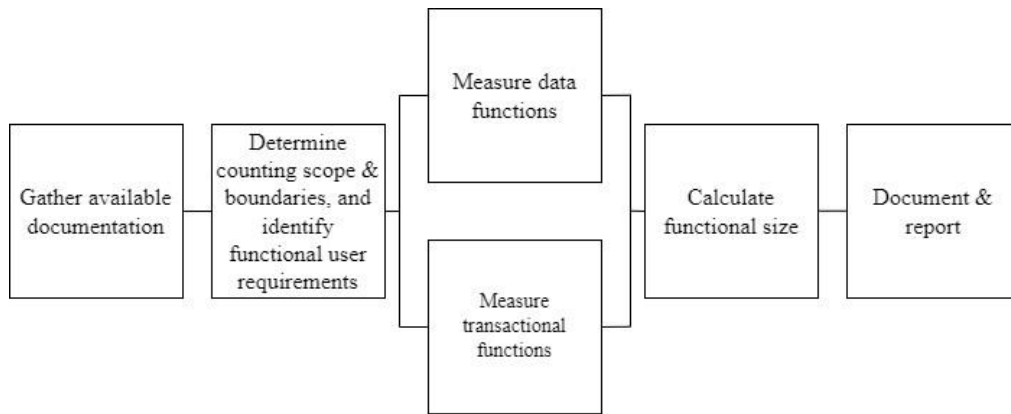


Рисунок 2.1 - Схема послідовності виконання етапів класичного методу функціональних точок

Перший з вказаних варіантів дозволяє оцінити функціональний розмір ІТ-проекту з розробки та постачання користувачам першого релізу застосункового програмного забезпечення ІС. Другий з вказаних варіантів дозволяє оцінити функціональний розмір ІТ-проекту з удосконалення експлуатованого (супроводжуваного) застосункового програмного забезпечення ІС. Третій з вказаних варіантів дозволяє оцінити функціональний розмір ІТ-проекту з розробки ІС на ранніх етапах її життєвого циклу. Під терміном "Додаток" ("Application") у [5] розуміється цілісний набір автоматизованих процедур та даних, що підтримують досягнення бізнес-цілей. Такий набір може складатися з одного або декількох компонентів, модулів або підсистем, що створюються ІС.

Основою методу FPA IFPUG є уявлення про реалізовану систему як множину структурних елементів, що належать до двох основних підгруп: дані та транзакції. Дані можна розділити на два класи: внутрішні логічні файли (ILF) та файли зовнішніх інтерфейсів (EIF). Транзакції поділяються на наступні класи: зовнішні введення (EI), зовнішні виводи (EO) та зовнішні запити (EQ). Приклад виділення цих елементів для додатка відділу кадрів поданий на рис. 2.2 [5].

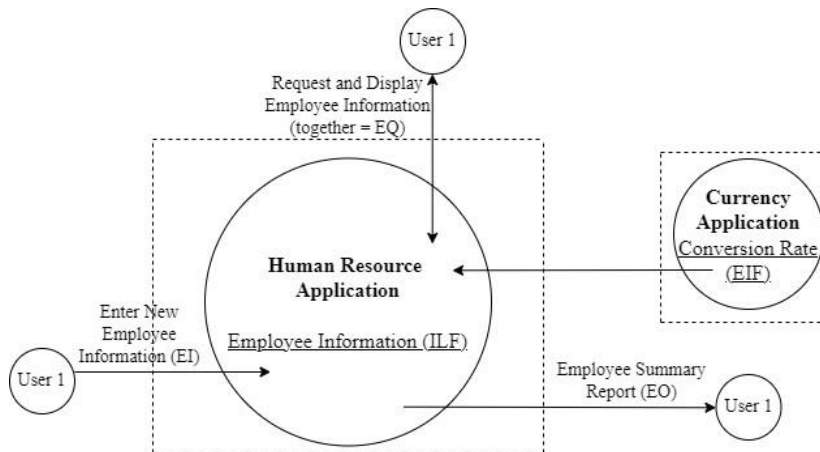


Рисунок 2.2 - Діаграма елементів, які лежать в основі обчислень кількості функціональних точок окремої функції інформаційної системи: EIF – Зовнішній Інтерфейсний Файл; ILF – Внутрішній Логічний Файл; EI – Зовнішнє Введення; EO – Зовнішній Вивід; EQ – Зовнішній Запит

Для фіксації кількісної оцінки функціонального розміру оцінюваної функції ІС у [5] пропонується використовувати наступні таблиці (див. табл. 2.1 і табл. 2.2).

Таблиця 2.1 - Визначення функціональної складності оцінюваної функції

Функції даних		RETs	DETs	Функціональна складність
№ п/п	Найменування			
Внутрішні логічні файли				
Номер ILF	Найменування ILF	Кількість RET	Кількість DET	Оцінка функціональної складності
Зовнішні файли інтерфейсу				
Номер EIF	Найменування EIF	Кількість RET	Кількість DET	Оцінка функціональної складності
Транзакційні функції		FTRs	DETs	Функціональна складність
№ п/п	Найменування			
Зовнішні входи				
Номер EI	Найменування EI	Кількість FTR	Кількість DET	Оцінка функціональної складності
Зовнішні виходи				

Кінець таблиці 2.1

Функції даних		RETs	DETs	Функціональна складність
№ п/п	Найменування			
Номер ЕО	Найменування ЕО	Кількість FTR	Кількість DET	Оцінка функціональної складності
Зовнішні запити				
Номер EQ	Найменування EQ	Кількість FTR	Кількість DET	Оцінка функціональної складності

Таблиця 2.2 - Розрахунок функціонального розміру оцінюваної функції

Тип функції	Кількість	Функціональна складність		Складність всього	Цілком типи функцій
ILF	Кількість ILF низької складності	Низька	x 7 =	Кількісна оцінка складності	Кількісна оцінка складності всіх ILF
Тип функції	Кількість	Функціональна складність	Складність всього	Цілком типи функцій	Тип функції
	Кількість ILF середньої складності	Середня	x 10 =	Кількісна оцінка складності	
	Кількість ILF високої складності	Висока	x 15 =	Кількісна оцінка складності	
EIF	Кількість EIF низької складності	Низька	x 5 =	Кількісна оцінка складності	Кількісна оцінка складності всіх EIF
	Кількість EIF середньої складності	Середня	x 7 =		
	Кількість EIF високої складності	Висока	x 10 =		

Кінець таблиці 2.2

Тип функції	Кількість	Функціональна складність	Складність всього	Цілком типи функцій	Тип функції
ЕІ	Кількість ЕІ низької складності	Низька	$x 3 =$	Кількісна оцінка складності	Кількісна оцінка складності всіх ЕІ
	Кількість ЕІ середньої складності	Середня	$x 4 =$	Кількісна оцінка складності	
	Кількість ЕІ високої складності	Висока	$x 6 =$	Кількісна оцінка складності	
ЕО	Кількість ЕО низької складності	Низька	$x 3 =$	Кількісна оцінка складності	Кількісна оцінка складності всіх ЕО
	Кількість ЕО середньої складності	Середня	$x 4 =$	Кількісна оцінка складності	
	Кількість ЕО високої складності	Висока	$x 6 =$	Кількісна оцінка складності	
EQ	Кількість EQ низької складності	Низька	$x 3 =$	Кількісна оцінка складності	Кількісна оцінка складності всіх EQ
	Кількість EQ середньої складності	Середня	$x 4 =$	Кількісна оцінка складності	
	Кількість EQ високої складності	Висока	$x 6 =$	Кількісна оцінка складності	
Повний функціональний розмір				Кількість функціональних точок	

В таблицях 2.1 і 2.2 прийняті наступні позначення [5]:

- RET - Record Element Type (визначений користувачем підтип елементів даних у межах функції обробки даних);
- DET - Data Element Type (унікальний, визнаваний користувачем, неповторний атрибут);
- FTR - File Type Referenced (функція, яка читає та/або обслуговується транзакційною функцією).

У відповідних графах таблиці 2.1 слід вказувати кількість RET та DET для кожного окремого ILF або EIF, а також кількість FTR та DET для кожного окремого EI, EO або EQ оцінюваної функції. Детальні правила визначення кількості RET, DET та FTR наведені в [5]. У методі FPA IFPUG факти змін елементів під час проектування та розробки ІС враховуються лише у варіанті розрахунку функціональних точок проекту удосконалення (Enhancement project function point count). Даний варіант передбачає розрахунок функціонального розміру ІТ-проекту з удосконалення експлуатованого (супроводжуваного) прикладного програмного забезпечення ІС за такою формулою [5]:

$$aEFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB), \quad (1)$$

де  $aEFP$  - скоригований функціональний розмір ІТ-проекту;

$ADD$  - розмір функцій, які планується додати в ході ІТ-проекту;

$CHGA$  - розмір функцій, які планується змінити в ході ІТ-проекту;

$CFP$  - розмір функцій, які слід піддати конвертації в ході ІТ-проекту;

$VAFA$  - коефіцієнт коригування розміру ІС після завершення ІТ-проекту;

$DEL$  - розмір функцій, які планується видалити в ході ІТ-проекту;

$VAFB$  - коефіцієнт коригування розміру запланованих до видалення функцій до початку ІТ-проекту.

Однак і для подібного варіанту розрахунку розмірів функцій, які слід скоригувати згідно з (1), виконується відповідно до тієї ж послідовності обчислень, що і розрахунок розмірів функцій в інших варіантах ІТ-проектів. При цьому кожна функція ІС в ході розрахунку її розміру розглядається окремо від інших функцій ІС. Таке обмеження серйозно спотворює точність оцінки трудовитрат на виконання ІТ-проектів, в яких передбачається повторне

використання розроблених раніше елементів системи. Під терміном «елемент системи» тут слід розуміти окремі функції ІС або окремі операції з даними, з яких складається опис окремої функції ІС. Суть цього спотворення полягає в тому, що всі структури даних, які беруть участь у визначенні значень елементів EIF, ILF, EI, EO та EQ, розглядаються як створювані «з нуля». Створення цих структур даних передбачає виконання певного обсягу робіт. Тим часом повторне використання вже існуючих структур даних, описи яких максимально збігаються з описами структур даних оцінюваної функції ІС, дозволяє значно скоротити обсяг цих робіт. Отже, основна ідея запропонованої адаптації методу FPA IFPUG повинна полягати в удосконаленні правил розрахунку значень елементів EIF, ILF, EI, EO та EQ в умовах повторного використання відповідних структур даних.

## 2.2 Опис теоретичної основи для модифікації методу аналізу функціональних точок

Як було відзначено в підрозділі 2.1, метод FPA IFPUG ґрунтується на припущенні, що кожна функція ІС створюється окремо. При цьому не враховується схожість створюваної функції ІС з раніше спроектованими та розробленими функціями тієї ж системи. Це припущення можна формалізувати, описавши описи всіх функцій ІС як неупорядковане множество:

$$F_{IS} = (f_1, f_2, \dots, f_i, \dots, f_n), \quad (2)$$

де  $f_i$  - опис  $i$ -ої функції створюваної ІС у вигляді сукупності структурованих множин;

$i$  - ідентифікатор функції ІС,  $i=1, \dots, n$ ;  $n$  - кількість функцій у створюваній ІС.

Застосування методології повторного використання окремих функцій у ході створення ІС передбачає первинну оцінку та виконання функцій, які потім будуть повторно використовуватися у ході створення інших функцій даної ІС. Для виконання цього вимоги введемо на множині (2) відношення часткового порядку:

$$R^{re-use}(F_{IS}) = (f_a \preceq f_b \mid f_a, f_b \in F_{IS}, f_a \rightarrow f_b), \quad (3)$$

Це відношення представляє випадки повторного використання функції  $f_a$  у ході створення функції  $f_b$  як імплікація між описами функції  $f_a$  та  $f_b$ .

Застосування методології повторного використання окремих функцій у створенні ІС дозволяє представити усю сукупність функцій цієї ІС у вигляді категорії структурованих множин:

$$L_{F_{IS}}^{re-use} = \left[ Ob = F_{IS}^{re-use}; Mon = \left( R^{re-use}(f_a, f_b) \right) \right], \quad (4)$$

де  $L_{F_{IS}}^{re-use}$  - позначення категорії структурованих описів функцій створюваної ІВ;

$Ob$  – позначення класу об'єктів категорії  $L_{F_{IS}}^{re-use}$ ;  $F_{IS}^{re-use}$  – безліч описів функцій створюваної ІВ із запровадженими відносинами часткового порядку;

$Mon$  – клас морфізмів категорії  $L_{F_{IS}}^{re-use}$ , які можна вважати узагальненням імплікацій між описами функцій  $f_a$  і  $f_b$ .

З урахуванням введених відношень часткового порядку  $\left( R^{re-use}(f_a, f_b) \right)$  множина  $F_{IS}^{re-use}$  приймає наступний вигляд:

$$F_{IS}^{re-use} = \left( \begin{array}{c} \langle f_1, (f_{11}, f_{12}, \dots, f_{1j}, \dots, f_{1m}) \rangle, \langle f_2, (f_{21}, f_{22}, \dots, f_{2j}, \dots, f_{2m}) \rangle, \dots \\ \langle f_i, (f_{i1}, f_{i2}, \dots, f_{ij}, \dots, f_{im}) \rangle, \dots, \langle f_k, (f_{k1}, f_{k2}, \dots, f_{kj}, \dots, f_{km}) \rangle, \dots \\ f_{k+1}, \dots, f_n \end{array} \right), \quad (5)$$

де  $(f_{i1}, f_{i2}, \dots, f_{ij}, \dots, f_{im})$  - функції, які створюються при повторному використанні функції  $f_i, i=1, \dots, k$ ;

$k$  - кількість функцій, які планується повторно використовувати;  $m$  - максимальна кількість функцій, які створюються при повторному використанні функції  $f_i, i=1, \dots, k$ ;

$f_{k+1}, \dots, f_n$  - функції, які не можуть бути створені при повторному використанні раніше розроблених функцій ІС.

Застосування множини (5) при плануванні ІТ-проекту створення ІС вимагає виконання операцій з повторним використанням функції  $f_i$  при створенні кожної функції з підмножини  $(f_{i1}, f_{i2}, \dots, f_{ij}, \dots, f_{im})$  негайно після створення функції  $f_i$ . Це обумовлено бажанням максимізувати використання досвіду виконавця робіт ІТ-проекту, отриманого ним під час створення функції  $f_i$ .

### 2.3 Модифікація методу аналізу функціональних точок

Застосування теоретико-категорного підходу до представлення ІС (4) та множини функцій ІС (5) під час планування ІТ-проекту створення ІС дозволяє переглянути правило обчислення функціонального розміру окремих функцій даної системи.

Для вихідної функції  $f_i$ , рішення про повторне використання якої було прийнято під час формування множини  $F_{IS}^{re-use}$ , правило обчислення кількості некоригованих функціональних точок або (див. таблиця 2.2) Total Functional Size описується, згідно з таблицею 2.2, наступним виразом [5, 16]:

$$\begin{aligned}
FP_{f_i} = & \sum_{a=1}^{n_{ILF}} FC_{ILF_{ia}} + \sum_{b=1}^{n_{EIF}} FC_{EIF_{ib}} + \\
& + \sum_{c=1}^{n_{EI}} FC_{EI_{ic}} + \sum_{d=1}^{n_{EO}} FC_{EO_{id}} + \sum_{e=1}^{n_{EQ}} FC_{EQ_{ie}},
\end{aligned} \tag{6}$$

де  $FP_{f_i}$  – кількість нескоригованих функціональних точок як оцінка значення Total Functional Size для функції  $f_i$ ;

$a$  – числовий ідентифікатор  $ILF_{ia}$ , що бере участь у описі функції  $f_i$ ;

$n_{ILF}$  – кількість  $ILF_{ia}$  що бере участь у описі функції  $f_i$ ;

$FC_{ILF_{ia}}$  – кількісна оцінка функціональної складності  $ILF_{ia}$ , бере участь у описі функції  $f_i$ ;

$b$  – числовий ідентифікатор  $EIF_{ib}$ , що бере участь в описі функції  $f_i$ ;

$n_{EIF}$  – кількість  $EIF_{ib}$ , що беруть участь в описі функції  $f_i$ ;

$FC_{EIF_{ib}}$  – кількісна оцінка функціональної складності  $EIF_{ib}$ , що бере участь в описі функції  $f_i$ ;

$c$  – чисельний ідентифікатор  $EI_{ic}$ , що бере участь в описі функції  $f_i$ ;

$n_{EI}$  – кількість  $EI_{ic}$ , що беруть участь в описі функції  $f_i$ ;

$FC_{EI_{ic}}$  – кількісна оцінка функціональної складності  $EI_{ic}$ , що бере участь в описі функції  $f_i$ ;

$d$  – числовий ідентифікатор  $EO_{id}$ , що бере участь в описі функції  $f_i$ ;

$n_{EO}$  – кількість  $EO_{id}$  що беруть участь у описі функції  $f_i$ ;

$FC_{EO_{id}}$  – кількісна оцінка функціональної складності  $EO_{id}$ , що бере участь в описі функції  $f_i$ ;

$e$  – числовий ідентифікатор  $EQ_{ie}$ , що бере участь в описі функції  $f_i$ ;

$n_{EQ}$  – кількість  $EQ_{ie}$ , що беруть участь у описі функції  $f_i$ ;

$FC_{EQ_{ie}}$  – кількісна оцінка функціональної складності  $EQ_{ie}$ , що бере участь у описі функції  $f_i$ .

Правила розрахунку значень  $FC_{ILF_{ia}}$ ,  $FC_{EIF_{ib}}$ ,  $FC_{EI_{ic}}$ ,  $FC_{EO_{id}}$  та  $FC_{EQ_{ie}}$  описуються, згідно з табл. 1, наступними виразами

$$\begin{aligned} FC_{ILF_{ia}} &= f(\langle n_{RET_{ia}}, n_{DET_{ia}} \rangle), & FC_{EIF_{ib}} &= f(\langle n_{RET_{ib}}, n_{DET_{ib}} \rangle), \\ FC_{EI_{ic}} &= f(\langle n_{FTR_{ic}}, n_{DET_{ic}} \rangle), & FC_{EO_{id}} &= f(\langle n_{FTR_{id}}, n_{DET_{id}} \rangle), \\ FC_{EQ_{ie}} &= f(\langle n_{FTR_{ie}}, n_{DET_{ie}} \rangle), \end{aligned} \quad (7)$$

де  $n_{RET_{ia}}$  – кількість елементів RET, присутніх в описі  $ILF_{ia}$ ;

$n_{DET_{ia}}$  – Кількість елементів DET, присутніх в описі  $ILF_{ia}$ ;

$n_{RET_{ib}}$  – Кількість елементів RET, присутніх в описі  $EIF_{ib}$ ;

$n_{DET_{ib}}$  – Кількість елементів DET, присутніх в описі  $EIF_{ib}$ ;

$n_{FTR_{ic}}$  – Кількість елементів FTR, присутніх в описі  $EI_{ic}$ ;

$n_{DET_{ic}}$  – Кількість елементів DET, присутніх в описі  $EI_{ic}$ ;

$n_{FTR_{id}}$  – Кількість елементів FTR, присутніх в описі  $EO_{id}$ ;

$n_{DET_{id}}$  – Кількість елементів DET, присутніх в описі  $EO_{id}$ ;

$n_{FTR_{ie}}$  – Кількість елементів FTR, присутніх в описі  $EQ_{ie}$ ;

$n_{DET_{ie}}$  – Кількість елементів DET, присутніх в описі  $EQ_{ie}$ ;

$f(\langle \bullet \rangle)$  – функція, що ставить у відповідність кортежу значень структурних елементів оцінку функціональної складності відповідного  $ILF_{ia}$ ,  $EIF_{ib}$ ,  $EI_{ic}$ ,  $EO_{id}$  або  $EQ_{ie}$ .

Тоді для функції  $f_{ij}$ , яку в ході формування множини  $F_{IS}^{re-use}$  планується створити шляхом ре-використання функції  $f_i$ , правило розрахунку кількості нескоригованих функціональних точок або (див. табл. 2.2) Total Functional Size пропонується описувати наступним виразом

$$\begin{aligned}
 FP_{f_{ij}} = & \sum_{a=1}^{n_{ILF}} FC_{ILF_{ija}} + \sum_{b=1}^{n_{EIF}} FC_{EIF_{ijb}} + \\
 & + \sum_{c=1}^{n_{EI}} FC_{EI_{ijc}} + \sum_{d=1}^{n_{EO}} FC_{EO_{ijd}} + \sum_{e=1}^{n_{EQ}} FC_{EQ_{ije}}.
 \end{aligned} \tag{8}$$

Правила розрахунку значень  $FC_{ILF_{ija}}$ ,  $FC_{EIF_{ijb}}$ ,  $FC_{EI_{ijc}}$ ,  $FC_{EO_{ijd}}$  и  $FC_{EQ_{ije}}$  описуються згідно табл. 2.1, наступними виразами

$$\begin{aligned}
 FC_{ILF_{ija}} &= f \left( \left\langle n_{RET_{ija}-RET_{ia}}, n_{DET_{ija}-DET_{ia}} \right\rangle \right), \\
 FC_{EIF_{ijb}} &= f \left( \left\langle n_{RET_{ijb}-RET_{ib}}, n_{DET_{ijb}-DET_{ib}} \right\rangle \right), \\
 FC_{EI_{ijc}} &= f \left( \left\langle n_{FTR_{ijc}-FTR_{ic}}, n_{DET_{ijc}-DET_{ic}} \right\rangle \right), \\
 FC_{EO_{ijd}} &= f \left( \left\langle n_{FTR_{ijd}-FTR_{id}}, n_{DET_{ijd}-DET_{id}} \right\rangle \right), \\
 FC_{EQ_{ije}} &= f \left( \left\langle n_{FTR_{ije}-FTR_{ie}}, n_{DET_{ije}-DET_{ie}} \right\rangle \right),
 \end{aligned} \tag{9}$$

де  $n_{RET_{ija}-RET_{ia}}$  – кількість елементів RET різниці множин  $(RET_{ija} - RET_{ia})$  з опису  $ILF_{ija}$ ;

$n_{DET_{ija}-DET_{ia}}$  – Кількість елементів DET різниці множин  $(DET_{ija} - DET_{ia})$  з опису  $ILF_{ija}$ ;

$n_{RET_{ijb}-RET_{ib}}$  – кількість елементів RET різниці множин  $(RET_{ijb} - RET_{ib})$  з  
 опису  $EIF_{ijb}$ ;

$n_{DET_{ijb}-DET_{ib}}$  – кількість елементів DET різниці множин  $(DET_{ijb} - DET_{ib})$  з  
 опису  $EIF_{ijb}$ ;

$n_{FTR_{ijc}-FTR_{ic}}$  – кількість елементів FTR різниці множин  $(FTR_{ijc} - FTR_{ic})$  з  
 опису  $EI_{ijc}$ ;

$n_{DET_{ijc}-DET_{ic}}$  – кількість елементів DET різниці множин  $(DET_{ijc} - DET_{ic})$  з  
 опису  $EI_{ijc}$ ;

$n_{FTR_{ijd}-FTR_{id}}$  – кількість елементів FTR різниці множин  $(FTR_{ijd} - FTR_{id})$  з  
 опису  $EO_{ijd}$ ;

$n_{DET_{id}}$  – кількість елементів DET різниці множин  $(DET_{ijd} - DET_{id})$  з опису  
 $EO_{ijd}$ ;

$n_{FTR_{ije}-FTR_{ie}}$  – кількість елементів FTR різниці множин  $(FTR_{ije} - FTR_{ie})$  з  
 опису  $EQ_{ije}$ ;

$n_{DET_{ije}-DET_{ie}}$  – кількість елементів DET різниці множин  $(DET_{ije} - DET_{ie})$  з  
 опису  $EQ_{ije}$ ;

$f(\langle \bullet \rangle)$  – функція, що ставить у відповідність кортежу значень структурних  
 елементів оцінку функціональної складності відповідного  $ILF_{ija}$ ,  $EIF_{ijb}$ ,  $EI_{ijc}$ ,  
 $EO_{ijd}$  або  $EQ_{ije}$ .

Однак запропонований спосіб розрахунку кількості функціональних точок не відображає обсяг зусиль з повторним використанням функції  $f_i$ , викликаних

синтаксичними відмінностями між предметними областями функцій  $f_i$  і  $f_{ij}$ .  
Для оцінювання цих зусиль пропонується модифікувати вираз (5)

наступним чином:

$$FP_{f_{ij}} = \left( \sum_{a=1}^{n_{ILF}} FC_{ILF_{ija}} + \sum_{b=1}^{n_{EIF}} FC_{EIF_{ijb}} + \sum_{c=1}^{n_{EI}} FC_{EI_{ijc}} + \sum_{d=1}^{n_{EO}} FC_{EO_{ijd}} + \sum_{e=1}^{n_{EQ}} FC_{EQ_{ije}} \right) + FP_{f_i} \times \frac{|Name_{ij} - Name_i|}{|Name_i|}, \quad (10)$$

де  $Name_{ij}$  – множина елементів, присутніх у найменуванні функції  $f_{ij}$ ;

$Name_i$  – множина елементів, що у найменуванні функції  $f_i$ .

Множина елементів  $Name_{ij}$  та  $Name_i$  пропонується формувати шляхом стеммінгу найменувань функцій  $f_{ij}$  і  $f_i$  з подальшим видаленням стоп-слів.

Підсумуючи розділ можемо сказати, що даний метод який було розглянуто у [20] та його подальша модифікація є безсумнівно новітньою. Тож даний метод безсумнівно може бути використаний та модифікований.

## 3 АНАЛІЗ ОСОБЛИВОСТЕЙ ЗАСТОСУВАННЯ МЕТОДУ ФУНКЦІОНАЛЬНИХ ТОЧОК

### 3.1.Схема алгоритма класичного методу функціональних точок

Схема алгоритму класичного методу функціональних точок передбачає використання різних прийомів і інструментів. Один підхід поєднує метод Function Point з JIRA для оцінки вимог до програмного забезпечення, вилучення значень функцій за допомогою CNN та визначення класифікації функцій та кількості балів на елемент, в кінцевому підсумку отримуючи загальну кількість функціональних точок шляхом зважування рівнів елементів. Інший обговорюваний алгоритм зосереджується на вирішенні задач зворотної томографії без лінеаризації або ітераційного уточнення, використовуючи експериментальні акустичні дані для реконструкції характеристик розсіювача . Крім того, побудовано функціонал, що генерує різні точкові розв'язки хвильового рівняння з випадковими коефіцієнтами, з подальшим статистичним методом усереднення цих коефіцієнтів [3]. Нарешті, надано комплексне лікування автоморфізмів, еквайзерів та зникаючих локусів для узагальнення класичної теорії перехресного співвідношення до  $S$ -значень точок на довільних схемах



Рисунок 3.1 - Послідовність кроків використання методу функціональних  
точок

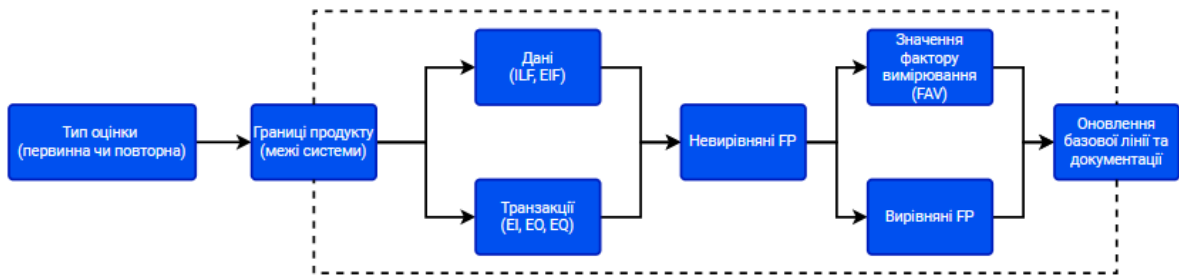


Рисунок 3.2 - Послідовність кроків повторного використання методу функціональних точок

### 3.2 Методика використання класичного методу функціональних точок

При оцінці якості методом функціональних точок може застосовуватися наступна формула:

$$\text{кількість дефектів/кількість рядків коду}$$

Висока величина буде свідчити про високу якість коду.

Метод функціональних точок (*Function point, FP*) базується на тому, що розмір ПЗ краще за все оцінювати в термінах кількості і складності функцій, реалізованих у даному програмному кодї, а не за допомогою кількості рядків коду. При використанні методу функціональних точок вимірюються категорії користувацьких бізнес-функцій.

Метод функціональних точок дозволяє виконати такі задачі:

- оцінити категорії користувацьких бізнес-функцій;
- вирішити проблему, пов'язану з намаганням застосувати одиниці вимірювання LOC на ранніх стадіях життєвого циклу розробки;

- визначати кількість і складність вхідних і вихідних даних, запити до бази даних, файли або структури даних, а також зовнішні інтерфейси, пов'язані з програмною системою.

Процес використання методу функціональних точок.

У таблиці 3.2 наведений робочий аркуш аналізу по методу функціональних точок.

Крок 1. Підрахунок функцій в кожній категорії.

Інструкція по визначенню рівня складності приведені у таблиці 3.3. При підрахунку структур даних (файлів) необхідно враховувати таке:

- внутрішні файли являють собою логічні файли у складі програми;
- структури даних (раніше відомі як «файли») являють собою первину логічну групу користувацьких даних, які постійно знаходяться повністю у середині меж програмної системи;
- структури даних доступні для користувачів за допомогою вводу, виходу, опитувань або інтерфейсів.

Структури даних поділяються на прості, середні і складні.

Інструкції по визначенню ступеню складності приводяться у таблиці 3.1.

Таблиця 3.1 - Інструкції по визначенню ступеню складності

Категорія	Простий	Середній	Складний	Функціональн і точки
Крок 1.	Підрахунок кількості функцій в кожній категорії			
Крок 2.	Використання вагових множників складності			
Кількість виходів	*4	*5	*7	
Кількість входів	*3	*4	*6	
Кількість опитувальних висновків	*4	*5	*7	
Кількість опитувань входів	*3	*4	*6	
Кількість файлів	*7	*10	*15	
Кількість інтерфейсів	*5	*7	*10	
Загальна кількість функціональних точок				
Крок 3.	Використання факторів середовища			
Фактор середовища				Рейтинг (0,1,2,3,4,5)

Кінець таблиці 3.1

Категорія	Простий	Середній	Складний	Функціональні точки
Канали передачі даних				
Розподіл обчислення				
Вимоги до продуктивності				
Конфігурування з обмеженнями				
Частота транзакцій				
Інтерактивний запит та/або запис				
Ефективність на рівні кінцевого користувача				
Інтерактивне оновлення				
Тяжка обробка				
Повторне використання				
Спрощення перетворення/установки				
Спрощення операції				
Використання на декількох вузлах				
Потенціал зміни функції				
Всього (N):				
Крок 4.	Обчислення корегувального множника складності (CAF)			
	$CAF = 0,65 + (0,01 * N) =$			
Крок 5.	Обчислення скоригованих функціональних точок (AFP)			
	$AFP = FP \text{ (початкове число)} * CAF =$			
Крок 6.	Перетворення в рядки LOC (додатково)			
	$LOC = AFP * LOC / AFP =$			

Таблиця 3.2 - робочий аркуш аналізу по методу функціональних точок

Частина виводу	Від 1 до 5 елементів	Від 6 до 19 елементів	20 і більше елементів даних
Від 0 до 1	Простий 4	Простий 4	Середній 5
Від 2 до 3	Простий 4	Середній 5	Складний 7
4 і більше	Середній 5	Складний 7	Складний 7
Частина вводу			
Від 0 до 1	Простий 3	Простий 3	Середній 4
Від 2 до 3	Простий 3	Середній 4	Складний 6
4 і більше	Середній 4	Складний 6	Складний 6

Таблиця 3.3 - Інструкція по визначенню рівня складності

Враховуючи логічні взаємозв'язки, а не фізичні типи записів	Від 1 до 9 елементів даних	Від 20 до 50 елементів даних	51 і більше елементів даних
1 логічний запис формат/взаємозв'язок	Простий 7	Простий 7	Середній 10
Від 2 до 5 логічних записів типу формат/взаємозв'язок	Простий 7	Середній 10	Складний 15
6 і більше логічних записів типу формат/взаємозв'язок	Середній 10	Складний 15	Складний 15

При підрахунку кількості інтерфейсів необхідно враховувати таке:

- зовнішні фактори являють собою файли, сгенеровані комп'ютером, які використовуються програмою;
- інтерфейси являють собою дані ( і систему управління), які зберігаються за межами програмної системи при виконанні оцінки;
- структури даних, які розділяються декількома системами, враховуються у вигляді інтерфейсів і структури даних;
- враховується кожний потік даних і управління у будь-якому напрямку в якості унікального інтерфейсу. Інтерфейси поділяються на прості, складні середні.

Інструкції по визначенню рівня складності приведені у таблиці 3.4

Таблиця 3.4 - Інструкції по визначенню рівня складності

Враховуючи логічні взаємозв'язки, а не фізичні типи записів	Від 1 до 9 елементів даних	Від 20 до 50 елементів даних	51 і більше елементів даних
1 логічний запис формат/взаємозв'язок	Простий 5	Простий 5	Середній 7
Від 2 до 5 логічних записів типу формат/взаємозв'язок	Простий 7	Середній 7	Складний 10
6 і більше логічних записів типу формат/взаємозв'язок	Середній 7	Складний 15	Складний 10

Крок 2. Застосування вагових множників складності.

Необхідно помножити кожен величину певного типу (простий, середній, складний) в середині кожної категорії (вивід, введення, опитування, структура даних, інтерфейси) на відповідні вагові множники (таблиці 3.2-3.4) в кожену категорію додаються вихідні результати, виражені в кількості «фізичних функціональних точок».

Крок 3. Застосування факторів середовища.

Короткий опис виконання процесу зважування в середовищі. Використовуючи матеріал таблиці 3.5, оцінемо кожний фактор по шкалі від 0 до 5 ( в даному випадку 0 значить неможливість застосування фактора).

Таблиця 3.5 – Фактори середовища

Фактор середовища	Рейтинг (0, 1, 2, 3, 4, 5)
Канали передачі даних	Дані або контрольна інформація, що приймається або одержується через канали передачі даних. Інтерактивна система завжди зазнає певного впливу з боку каналів передачі даних
Розподілені обчислення	Додаток, що використовує дані, що зберігаються, обробляються і до яких забезпечується доступ до сховища або системи обробки, що відрізняються від використаних в основній системі.
Вимоги до продуктивності	Вимоги, схвалені користувачем, застосовані для отримання високої швидкості передачі даних або меншого часу відгуку
Конфігурування з обмеженнями	Додаток виконується із застосуванням інтенсивної використовуваної, обмеженої або наповненої конфігурації
Частота транзакцій	Високий мережевий трафік, перевантаженість екранів інформацією та графікою, висока частота оновлення екрану
Інтерактивний запит та/або запис	Висока ступінь інтерактивності
Ефективність на рівні кінцевого користувача	Необхідно додатково урахувати людський фактор

Кінець таблиці 3.5

Фактор середовища	Рейтинг (0, 1, 2, 3, 4, 5)
Інтерактивне оновлення	Динамічне оновлення бази даних, розподілені бази даних
Тяжка обробка	Висока ступінь безпеки, обробка з великою кількістю транзакцій, важкі алгоритми, логіка, керована перериваннями
Повторне використання	Код, розроблений з метою повторного використання, повинен володіти високою якістю.
Спрощення використання/ встановлення	Процеси перетворення та встановлення вимагають наявності документів із планування, які були протестовані
Спрощення операції	Ефективні, але прості процедури запуску резервування, відновлення помилок і завершення. мінімальне втручання з боку користувача
Використовується на кількох вузлах	Облік відмінностей в бізнес-функціях
Потенціал змінення функції	Модульність, керованість за допомогою таблиць, підтримка з боку користувачів, можливості формування запитів і так далі.

З метою отримання «ефекту присутності» на одному з країв спектру оцінювання, в табл. 3.6 приводяться приклади програмних систем з високим рейтингом – показники рейтингу від 4 до 5 на шкалі.

Просумуємо рейтинги факторів ( $F_n$ ) з метою розрахунку кінцевого фактора впливу середовища ( $N$ ).

$$N = \text{sum} (F_n) \quad (11)$$

Значення заповнюються в робочий аркуш аналізу по методу функціональних точок. Заповнений робочий аркуш наведений на таблиці 3.7

Крок 4. Розрахунок множника коригування складності.

Таблиця 3.6 - приклади програмних систем з високим рейтингом

№	Фактор середовища	Приклади високопродуктивних систем
1	Складні комунікації даних	Програма, призначена для міжнародного банку, яка виконує грошові перекази з фінансових установ, розкиданих по всьому світу

Продовження таблиці 3.6

№	Фактор середовища	Приклади високопродуктивних систем
2	Розподілена обробка	Механізм пошуку в Internet при реалізації якого обробка виконується кількома десятками комп'ютерів, які працюють у паралельному режимі
3	Цілі, що вимагають досягнення максимальної продуктивності	Система контролю повітряного трафіку підтримує точне та своєчасне визначення позиції повітряного судна на основі показань радіолокатора
4	Конфігурування, що інтенсивно використовується	Університетська система реалізує одночасну реєстрацію кількох сотень студентів, які перебувають у лабораторії
5	Висока частота транзакцій	Банківська програма виконує мільйони транзакцій за аналіз з метою зведення балансу за всіма книгами на початок нового робочого дня
6	Інтерактивний ввід даних	Програма затвердження заставних, за допомогою якої клерки можуть вводити дані в комп'ютерну систему в інтерактивному режимі, використовуючи паперові анкети, заповнені майбутніми домовласниками
7	Проект, дружній по відношенню до користувача	Програмне забезпечення для комп'ютеризованих кас на станціях метро обладнаних сенсорними екранами, що дозволяють купувати квитки за допомогою кредитних карток
8	Інтерактивне оновлення даних	Авіаційна система, що дозволяє агентом туристичних фірм замовляти авіарейси та отримувати інформацію про наявність вільних місць. Це ПЗ повинно забезпечувати можливість блокування та модифікації деяких записів у базі даних з метою запобігання повторному продажу квитків на ті самі місця.
9	Комплексна обробка	Медичне ПЗ, яке на основі різних симптомів виявляє у пацієнта, а також шляхом великого логічного висновку, дозволяє встановлювати попередній діагноз.
10	Повторне використання	Робочий процесор який може бути спроектований таким чином, що його панель інструментів і меню можуть вбудовуватися в інші програми такі як електронна таблиця або генератор звітів

Кінець таблиці 3.6

№	Фактор середовища	Приклади високопродуктивних систем
11	Полегшене установка	Додаток виконує контроль обладнання за допомогою якого навіть не фахівець може встановити та налаштувати обладнання
12	Полегшене оперування	Програма, призначена для аналізу великої кількості хронологічних фінансових записів, яка повинна оптимізувати обробку інформації таким чином, щоб не змушувати оператора постійно змінювати носії даних
13	Декілька вузлів	ПЗ з підтримки платіжних відомостей для транснаціональної корпорації, які повинні враховувати різні характеристики різних країн, включаючи різну валюту та правила визначення податкових ставок
14	Гнучкість	Програма фінансового прогнозу, яка може формувати місячні, квартальні чи річні прогнози, необхідні конкретному бізнес-менеджеру

Рівень невизначеності оцінок є функцією фази життєвого циклу. Це підтримує теорію емпіричних даних, відповідно до якої максимальний ступінь впливу факторів середовища на результати аналізу функціональних точок може складати +/- 35%. Причому розглядається максимальний ступінь впливу, так як аналіз функціональних точок виконується на початку життєвого циклу. Як наслідок, у цьому випадку існує максимальна вірогідність неточності вимірювань. З метою компенсації подібної невизначеності у випадку недостатнього рівня знань множник корегування складності (CAF) буде застосовуватися к результативним значенням факторів середовища.

$$CAF = 0.65 + (0.01 * N) , \quad (12)$$

де N – сума зважених факторів середовища.

Так як ми маємо справу з 14 факторами середовища, кожний з яких має вагу, що змінюється і діапазоні від 0 до 5, найменше значення для N може бути 0 (кожний з 14 факторів не має застосування); а найбільше значення для N може бути 70 (кожний з 14 факторів має максимальну вагу, дорівнює 5). Виходячи з цих граничних умов, приходимо до висновку, що мінімум CAF = 0.65 + (0,01 \*

0) = 0,65. Максимум САФ =  $0.65 + (0,01 * 70) = 1,35$ . ( $1.35 - 0.65 = 0.70$ ). Рання оцінка розміру і трудовитрат може відхилитись при умові використання фактору на величину +/- 35%.

Крок 4 проілюстрований в таблиці 3.6.

Крок 5. Обрахування скорегованих функціональних точок.

Скореговані функціональні точки (AFP) – фізичні функціональні точки \* САФ.

Крок 6. Перетворення у рядки LOC (додатково).

Метод функціональних точок забезпечує спосіб попередньої оцінки розміру потенціальних програм або програмних систем. При цьому здійснюється аналіз майбутніх функціональних властивостей з користувачької точки зору. Мовипрограмування є зовсім різним з точки зору їх характеристик, проте існує деяка середня кількість виконуваних операторів, необхідних для реалізації однієї функціональної точки.

Перетворення функціональних точок у рядки LOC може вимагатися в силу таких причин:

- з метою вимірювання і порівняння продуктивності або розміру програм або систем, які були написані на різних мовах програмування;
- з метою використання стандартних одиниць вимірювання для здійснення введення даних в інструментальні засоби оцінки;
- для перетворення розміру програми або додатку, написаних на будь-якій мові програмування, в еквівалентний розмір у випадку додатку, написаного на іншій мові програмування.

На таблиці 3.7 ілюструється перетворення функціональних точок в LOC (перший і третій стовбці). В таблиці перераховані не всі мовні перетворення, схвалені 11-TUG. Також необхідно відмітити, що цей перелік постійно поповнюється по мірі розробки нових мов програмування.

$LOC = \text{скореговані функціональні точки} * LOC \text{ на скореговану функціональну точку}$   
 $AFP \times \# LOC \text{ на } AFP = LOC$

Таблиця 3.7 - перетворення функціональних точок в LOC

Категорія	Простий	Середній	Складний	Функціональні точки
Крок 1.	Підрахунок кількості функцій в кожній категорії			
Крок 2.	Використання вагових множників складності			
	1	2	3	(1+2+3)
Кількість виходів	*4	*5	*7	48+55+35=138
Кількість входів	*3	*4	*6	96
Кількість опитувальних висновків	*4	*5	*7	76
Кількість опитувань входів	*3	*4	*6	71
Кількість файлів	*7	*10	*15	144
Кількість інтерфейсів	*5	*7	*10	127
Загальна кількість функціональних точок				652
Крок 3.	Використання факторів середовища			
Фактор середовища			Рейтинг (0,1,2,3,4,5)	
Канали передачі даних			5	
Розподіл обчислення			5	
Вимоги до продуктивності			3	
Конфігурування з обмеженнями			0	
Частота транзакцій			5	
Інтерактивний запит та/або запис			4	
Ефективність на рівні кінцевого користувача			5	
Інтерактивне оновлення			4	
Тяжка обробка			2	
Повторне використання			2	
Спрощення перетворення/установки			3	
Спрощення операції			4	
Використання на декількох вузлах			5	
Потенціал зміни функції			4	
Всього (N):			51	
Крок 4.		Обчислення корегувального множника складності (CAF)		
		$CAF = 0,65 + (0,01 * N) = 0,65 + (0,01 * 51) = 1,16$		
Крок 5.		Обчислення скоригованих функціональних точок (AFP)		
		$AFP = FP \text{ (початкове число)} * CAF = 652 * 1,16 = 756,32$		
Крок 6.		Перетворення в рядки LOC (додатково)		
		$LOC = AFP * LOC / AFP = 756,32 * 128 = 96,808.96$		

Виходячі з вищесказаного можемо сказати, що метод Функціональних точок може підійти до різних ІТ- проектів. Його алгоритм та використання не є занадто складним тому його використання значно полегшує роботу з ІТ- проектом.

## 4 АПРОБАЦІЯ МОДИФІКОВАНОГО МЕТОДУ ФУНКЦІОНАЛЬНИХ ТОЧОК

4.1 “Загальні відомості про ІТ проект розробки функціональної задачі. Формування та ведення індивідуального плану науково-педагогічного працівника кафедри” Інформаційно-аналітичної системи Університет

Для перевірки отриманих результатів пропонується використовувати опубліковані в [19] дані ІТ-проекту розробки функціонального завдання «Формування та ведення індивідуального плану (ІП) науково-педагогічного працівника кафедри». Метою цього ІТ-проекту було створення функціонального завдання, яке дозволяє користувачеві автоматизувати роботи з формування та подальших змін документа «ІП науково-педагогічного працівника кафедри». Цей ІТ-проект виконувався за активної участі студентки гр. УПГІТм-20-1 Кузьми Єлизавети Анатоліївни

У ході виконання Етапів 1 та 2 модифікованого методу були розроблені:

- операційний опис архітектури функціонального завдання «Формування та ведення ІП науково-педагогічного працівника кафедри» у вигляді діаграми потоків даних;
- структурний опис архітектури функціонального завдання «Формування та ведення ІП науково-педагогічного працівника кафедри» у вигляді діаграми «сутність – зв'язок».

Операційний опис архітектури функціонального завдання наведено в табл. 3. Як числові номери в табл. 1 вказані номери робіт, вхідних та вихідних потоків, які були сформовані CASE-засобом AllFusion Process Modeler під час створення діаграми потоків даних [19].

Структурний опис архітектури функціонального завдання наведено на рис. 4.1 [19]. Багато описів сутностей, представлених на рис. 4.1, наведено у табл. 4.1 [19].

Таблиця 4.1 - Операційний опис функціонального завдання «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» (на основі діаграми потоків даних)

Робота		Вхідний потік		Вихідний потік	
№	Найменування	№	Найменування	№	Найменування
1	Конвертація розділу «Навчальна робота»	1	Навчальне навантаження викладача на навчальний рік	2	Інформація з розділу індивідуального плану (ІІІ) «Навчальна робота»
2	Формування розділу «Наукова робота»	2	Інформація про викладача	3	Інформація з розділу ІІІ «Наукова робота»
		3	Інформація про заплановані для виконання роботи		
		5	Інформація про рекомендовані до виконання роботи		
		8	Інформація з розділу ІІІ «Наукова робота»		
		12	Залишок годинника		
3	Формування розділу «Методична робота»	2	Інформація про викладача	4	Інформація з розділу ІІІ «Методична робота»
		3	Інформація про заплановані для виконання роботи		
		5	Інформація про рекомендовані до виконання роботи		
		9	Інформація з розділу ІІІ «Методична робота»		
		12	Залишок годинника		
4	Формування розділу «Організаційна робота»	2	Інформація про викладача	5	Інформація з розділу ІІІ «Організаційна робота»
		3	Інформація про заплановані для виконання роботи		
		5	Інформація про рекомендовані до виконання роботи		
		10	Інформація з розділу ІІІ «Організаційна робота»		
		12	Залишок годинника		
5	Формування переліку посад та довгострокових доручень	4	Інформація про посади та довгострокові доручення	6	Інформація з розділу ІІІ «Перелік посад та довгострокових доручень»

Кінець таблиці 4.1

Робота			Вхідний потік	Вихідний потік	
№	Найменування	№	Найменування	№	Найменування
		11	Інформація з розділу ІІІ «Перелік посад та довгострокових доручень»		
6	Формування переліку рекомендованих до виконання робіт	5	Інформація про рекомендовані до виконання роботи	1	Інформація про рекомендовані до виконання роботи
7	Формування та ведення нормативно-довідкової інформації за ключовими показниками ефективності (КРІ)	6	Інформація про ключові КРІ кафедри	7	Інформація про ключові КРІ кафедри
8	Формування КРІ викладача та частини від КРІ кафедри	8	Інформація з розділу ІІІ «Наукова робота»	9	Інформація про КРІ викладача та частини від КРІ кафедри
9	Формування зведеної таблиці на навчальний рік	9	Інформація з розділу ІІІ «Методична робота»	8	Інформація про кількість годин за розділами ІІІ
		7	Інформація з розділу ІІІ «Учбова робота»		
		8	Інформація з розділу ІІІ «Наукова робота»		
		10	Інформація з розділу ІІІ «Організаційна робота»		
10	Формування вихідного документа "ІІІ"	9	Інформація з розділу ІІІ «Методична робота»	10	ІІІ
		7	Інформація з розділу ІІІ «Учбова робота»		
		8	Інформація з розділу ІІІ «Наукова робота»		
		10	Інформація з розділу ІІІ «Організаційна робота»		
		11	Інформація з розділу ІІІ «Перелік посад та довгострокових доручень»		

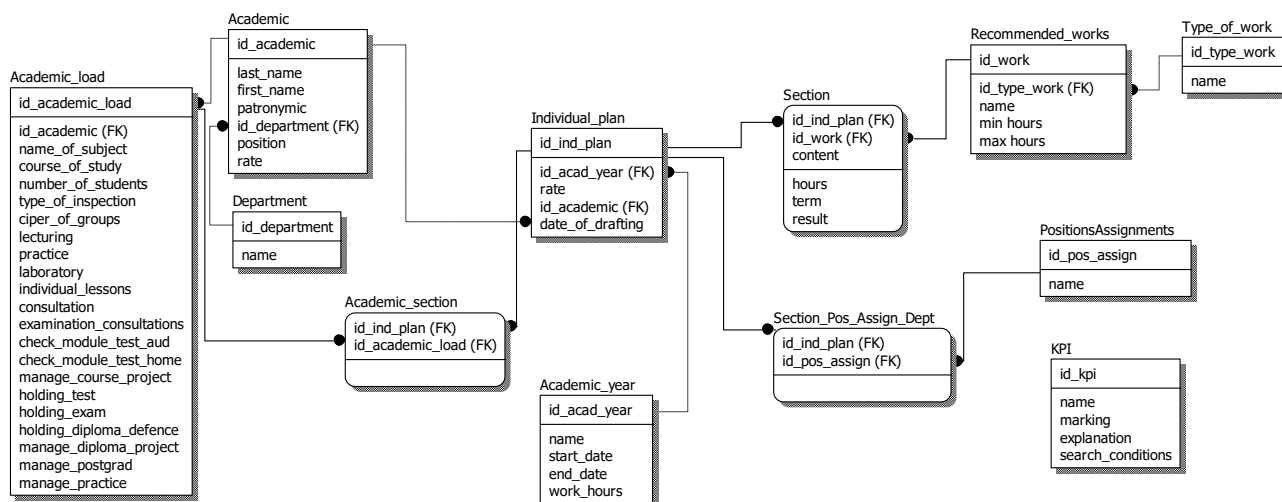


Рисунок 4.1 - Структурний опис функціонального завдання «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» у вигляді діаграми «сутність – зв'язок»

Таблиця 4.2 - Безліч описів сутностей функціонального завдання

Числовий ідентифікатор	Найменування
1	Academic_load
2	Academic
3	Department
4	Individual_plan
5	Academic_section
6	Academic_year
7	Section
8	Recommended_works
9	Type_of_work
10	Section_Pos_Assign_Dept
11	PositionsAssignments
12	KPI

У табл. 4.1 як числові ідентифікатори вказані номери сутностей, сформовані за допомогою CASE-засобу AllFusion ERwin Data Modeler в ході розробки показаної на рис. 4.1 діаграми «сутність – зв'язок» та імпортовані в CASE-засіб AllFusion Process Modeler для зв'язку діаграми «сутність – зв'язок» з діаграмою потоків даних завдання [19].

Результати визначення параметрів окремих функцій завдання, а також ідентифікатори сутностей, використаних для опису робіт, вхідних та вихідних потоків даних, наведено у табл. 4.3.

Таблиця 4.3 - Вихідна інформація для застосування методу FPA IFPLUG

№ функції	Ідентифікатор підмножини	№ елемента / Позначення параметра методу	Безліч ідентифікаторів сутностей, що описують елемент
1	1	1/ILF	{1, 2, 3, 4, 5, 6}
	2	1/EIF	{1, 2, 3}
	3	2/EIF	{1, 2, 4, 5, 6}
2	1	2/ILF	{1, 2, 3, 4, 5, 6, 7, 8, 9}
	2	2/EIF	{2, 3}
		3/EI	{2, 4, 6, 7, 8, 9}
		5/EIF	{8, 9}
		8/EIF	{1, 2, 4, 6, 7, 8, 9}
		12 / EIF	{1, 2, 4, 5, 6, 7}
3	3/EIF	{2, 4, 6, 7, 8, 9}	
3	1	3/ILF	{1, 2, 3, 4, 5, 6, 7, 8, 9}
	2	2 / EIF	{2, 3}
		3/EI	{2, 4, 6, 7, 8, 9}
		5 / EIF	{8, 9}
		9 / EIF	{2, 4, 6, 7, 8, 9}
		12 / EIF	{1, 2, 4, 5, 6, 7}
3	4 / EIF	{2, 4, 6, 7, 8, 9}	
4	1	4/ILF	{1, 2, 3, 4, 5, 6, 7, 8, 9}
	2	2 / EIF	{2, 3}
		3/EI	{2, 4, 6, 7, 8, 9}
		5 / EIF	{8, 9}
		10 / EIF	{2, 4, 6, 7, 8, 9}
		12 / EIF	{1, 2, 4, 5, 6, 7}
3	5 / EIF	{2, 4, 6, 7, 8, 9}	
5	1	5/ILF	{2, 4, 6, 7, 10, 11}
	2	4/EI	{2, 4, 6, 7, 10, 11}
		11 / EIF	{2, 4, 6, 7, 10, 11}
3	6 / EIF	{2, 4, 6, 7, 10, 11}	
6	1	6 / ILF	{8, 9}
	2	5/EI	{8, 9}
	3	1 / EIF	{8, 9}
7	1	7/ILF	{12}
	2	6 / EI	{12}
	3	7 / EIF	{12}

Кінець таблиці 4.3

№ функції	Ідентифікатор підмножини	№ елемента / Позначення параметра методу	Безліч ідентифікаторів сутностей, що описують елемент
8	1	8/ILF	{1, 2, 4, 5, 6, 7, 8, 9, 12}
	2	8 / EIF	{1, 2, 4, 6, 7, 8, 9}
	3	9 / EO	{1, 2, 4, 5, 6, 7, 12}
9	1	9/ILF	{1, 2, 4, 5, 6, 7, 8, 9}
	2	7 / EIF	{1, 2, 4, 5, 6}
		8 / EIF	{1, 2, 4, 6, 7, 8, 9}
		9 / EIF	{2, 4, 6, 7, 8, 9}
		10 / EIF	{2, 4, 6, 7, 8, 9}
	3	8 / EO	{1, 2, 4, 5, 6, 7}
10	1	10/ILF	{1, 2, 4, 5, 6, 7, 8, 9, 10, 11}
	2	7 / EIF	{1, 2, 4, 5, 6}
		8 / EIF	{1, 2, 4, 6, 7, 8, 9}
		9 / EIF	{1, 4, 6, 7, 8, 9}
		10 / EIF	{2, 4, 6, 7, 8, 9}
		11 / EIF	{2, 4, 6, 7, 10, 11}
	3	10 / EO	{1, 2, 4, 5, 6, 7, 8, 9, 10, 11}

У табл. 4.1 ідентифікатор підмножини набуває наступних значень:

- 1 – підмножина опису роботи діаграми потоків даних;
- 2 - підмножина опису вхідних потоків діаграми потоків даних;
- 3 – підмножина опису вихідних потоків діаграми потоків даних.

Вміст осередків "Номер елемента" в табл. 4.1 відповідає номерам робіт (для підмножин 1), номерам вхідних потоків (для підмножини 2) та номерам вихідних потоків (для підмножини 3), взятих з табл. 2.1. Як ідентифікатори сутностей використано їх номери з табл. 2.2 [19].

У ході виконання Етапу 3 модифікованого методу було визначено кластери функцій, чії структурні описи хоча б частково подібні один до одного. Результатом виконання цього етапу є дендрограма функцій як конфігураційних елементів задачі, наведена на рис. 4.2 [19].

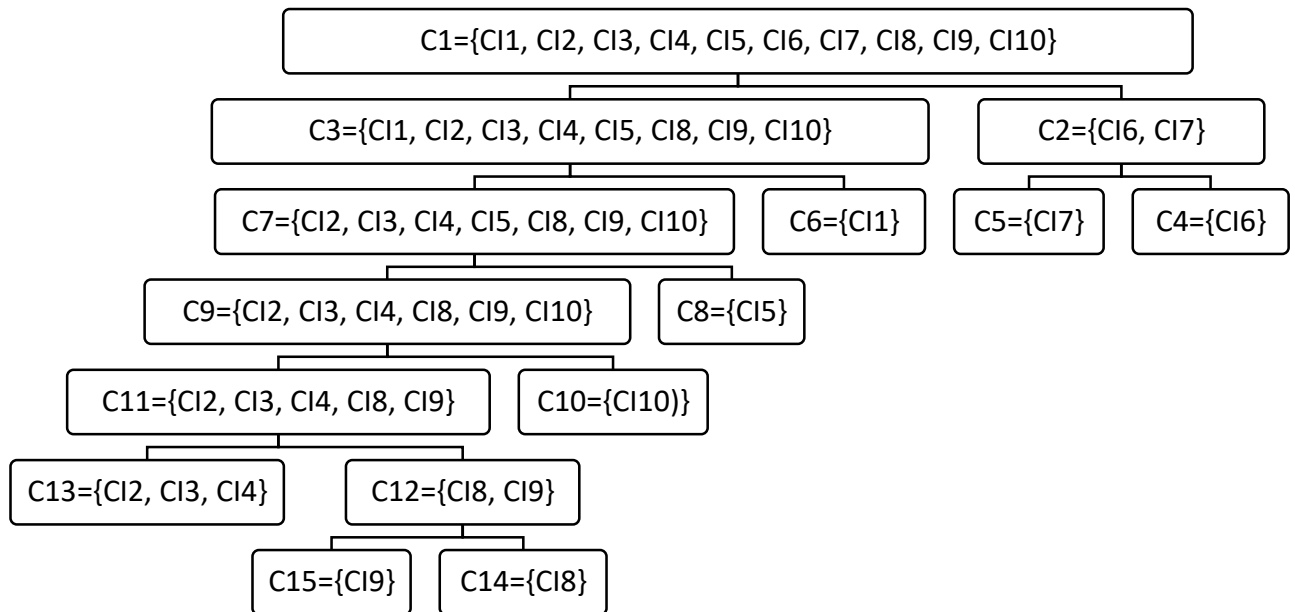


Рисунок 4.2 - Дендрограма кластерів конфігураційних елементів задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»

Номери конфігураційних елементів (CI) наведені на рис. 4.2, збігаються з номерами робіт у табл. 4.1 та номерами функцій у табл. 4.2 Детальний опис методу формування цієї дендрограми викладено у [19].

#### 4.2 Апробація модифікованого методу аналізу функціональних точок

Розглянемо апробацію запропонованого модифікованого методу FPA на прикладі оцінювання зусиль на виконання ІТ-проекту розробки функціональних задач в умовах ре-використання окремих функцій цієї задачі.

У ході виконання Етапу 3 модифікованого методу було визначено кластери функцій, чий структурні описи хоча б частково подібні один до одного. Результатом виконання цього етапу є дендрограма функцій як конфігураційних елементів задачі, наведена на рис. 4.2 [19].

Аналіз дендрограми дозволив виділити для прийняття рішення про ре-

використання підмножина функцій f2 «Формування розділу «Наукова робота», f3 «Формування розділу «Методична робота» та f4 «Формування розділу «Організаційна робота». Структурні описи та найменування цих функцій максимально збігаються один з одним. Даний варіант пропонується описати кортежем <f2, (f3, f4)>. Згідно з цим описом, функція f2 буде ре-використана для створення функцій f3 і f4.

На початку було розраховано кількість функціональних точок, яким оцінюються зусилля створення функції f2. Для цього використані рекомендовані в [5] табличні форми подання ходу та результатів розрахунків (табл. 4.4, 4.5).

Таблиця 4.4 - Визначення параметрів ILF, EIF та EI для функції «Формування розділу «Наукова робота»

Функції даних		RETs	DETs	Функціональна складність
№ з/п	Найменування			
Внутрішні логічні файли				
1	Формування розділу «Наукова робота»	7	60	Висока
Зовнішні файли інтерфейсу				
1	Інформація про викладача	2	9	Низька
2	Інформація про рекомендовані до виконання роботи	2	7	Низька
3	Інформація з розділу ІІІ «Наукова робота»	6	51	Висока
4	Залишок годинника	5	48	Середня
Транзакційні функції		FTRs	DETs	Функціональна складність
№ з/п	Найменування			
Зовнішні входи				
1	Інформація про заплановані для виконання роботи	2	111	Висока

Таблиця 4.5- Розрахунок функціонального розміру проекту розробки функції «Формування розділу «Наукова робота»

Тип функції	Кількість	Функціональна складність		Складність всього	Цілком типи функцій
ILF	0	Низька	x 7 =	0	15
	0	Середня	x 10 =	0	
	1	Висока	x 15 =	15	
EIF	2	Низька	x 5 =	10	27
	1	Середня	x 7 =	7	
	1	Висока	x 10 =	10	
EI	0	Низька	x 3 =	0	6
	0	Середня	x 4 =	0	
	1	Висока	x 6 =	6	
Повний функціональний розмір					48

Потім було розраховано кількість функціональних точок, яким оцінюються зусилля створення функції f3. На початку були визначені параметрів ILF, EIF та EI для функції «Формування розділу «Наукова робота». Результати цієї операції наведено у табл. 4.6.

Таблиця 4.6 - Визначення параметрів ILF, EIF та EI для функції «Формування розділу «Методична робота»

Функції даних		RETs	DETs	Функціональна складність
№ з/п	Найменування			
Внутрішні логічні файли				
1	Формування розділу «Методична робота»	7	60	Висока
Зовнішні файли інтерфейсу				
1	Інформація про викладача	2	9	Низька
2	Інформація про рекомендовані до виконання роботи	2	7	Низька
3	Інформація з розділу ІІІ «Методична робота»	6	51	Висока
4	Залишок годинника	5	48	Середня
Транзакційні функції		FTRs	DETs	Функціональна складність
№ з/п	Найменування			
Зовнішні входи				
1	Інформація про заплановані для виконання роботи	2	111	Висока

Як впливає з виразу (9), необхідно спочатку визначити кількість елементів у різницях множин (RET<sub>ija</sub>-RET<sub>ia</sub>), (DET<sub>ija</sub>-DET<sub>ia</sub>), (RET<sub>ijb</sub>-RET<sub>ib</sub>), (DET<sub>ijb</sub>-DET<sub>ib</sub>), (FTR<sub>ijc</sub>-FTR<sub>ic</sub>) та (DET<sub>ijc</sub>-DET<sub>ic</sub>). Однак кількість елементів у різницях множин (RET<sub>ija</sub>-RET<sub>ia</sub>), (RET<sub>ijb</sub>-RET<sub>ib</sub>) і (FTR<sub>ijc</sub>-FTR<sub>ic</sub>) для функцій  $f_{ij}=f_3$  і  $f_i=f_2$  дорівнює 0, це означає, що значення  $FC_{ILF_{ija}}$ ,  $FC_{EIF_{ijb}}$  і  $FC_{EI_{ijc}}$  для функції  $f_3$  дорівнюють нулю. Отже, перша частина виразу (10) також дорівнює 0. Це означає, що функція  $f_3$  на рівні її структурного опису виділеними на рис. 4.1 сутностями повністю збігається з аналогічним структурним описом функції  $f_2$ .

Для розрахунку значення другої частини виразу (10) необхідно виконати стемінг найменувань функцій  $f_3$  та  $f_2$  з подальшим видаленням стоп-слів. Результати цих операцій із застосуванням стеммера Портера наведено у табл. 4.7.

Таблиця 4.7 - Результати попередньої обробки найменувань функцій  $f_3$  та  $f_2$

Стадія попередньої обробки	Найменування функції $f_3$	Найменування функції $f_2$
Вихідне найменування кадру	Формування секції "Methodical work"	Формування розділу "Scientific work"
Результати застосування стеммера Портера	Формат розділу "Метод роботи"	Формат розділу "Scientif work"
Результати видалення стоп-слів	Формат секції "Method work"	Формат секції "Scientif work"

Наведені у табл. 4.7 результати дозволяють визначити наступні значення елементів другої частини виразу (10):

- name3=(Format,section,Method,work);
- name2=(Format,section,Scientif,work);
- | Name2 | = 4;
- | Name3-Name2|=1.

Таким чином, остаточна оцінка зусиль зі створення функції  $f_3$  з урахуванням ре-використання функції  $f_2$  становить:

$$FP_{f_3} = 0 + 48 \times \frac{1}{4} = 12 \text{ (function points)}. \quad (12)$$

Слід зазначити, що у разі застосування вихідного методу FPA IFPLUG значення оцінки зусиль створення функції f3 було б визначено як  $FP_{f_3} = 48$  функціональних точок. Дана оцінка для випадку використання функції f2 в ході створення функції f3 є надмірно завищеною.

Потім було розраховано кількість функціональних точок, яким оцінюються зусилля створення функції f4. На початку були визначені параметрів ILF, EIF та EI для функції «Формування розділу «Організаційна робота». Результати цієї операції наведено у табл. 4.8.

Таблиця 4.8 - Визначення параметрів ILF, EIF та EI для функції «Формування розділу «Організаційна робота»

Функції даних		RETs	DETs	Функціональна складність
№ з/п	Найменування			
Внутрішні логічні файли				
1	Формування розділу «Організаційна робота»	7	60	Висока
Зовнішні файли інтерфейсу				
1	Інформація про викладача	2	9	Низька
2	Інформація про рекомендовані до виконання роботи	2	7	Низька
3	Інформація з розділу III «Організаційна робота»	6	51	Висока
4	Залишок годинника	5	48	Середня
Транзакційні функції		FTRs	DETs	Функціональна складність
№ з/п	Найменування			
Зовнішні входи				
1	Інформація про заплановані для виконання роботи	2	111	Висока

За аналогією з розрахунками для функції f3, остаточна оцінка зусиль зі створення функції f4 з урахуванням ре-використання функції f2 становить  $FP_{f_4} = 12$  функціональних точок. У разі застосування вихідного методу FPA

IFPLUG значення оцінки зусиль створення функції  $f_4$  було б визначено як  $FP_{f_4} = 48$  функціональних точок.

Таким чином, сумарна оцінка зусиль на створення функцій  $f_2$ ,  $f_3$  і  $f_4$  дорівнює:

- у разі застосування модифікованого методу для кортежу  $\langle f_2, (f_3, f_4) \rangle$   $48+12+12=72$  функціональні точки;

- у разі застосування вихідного методу FPA IFPLUG  $48+48+48=144$  функціональні точки.

Можна стверджувати, що застосування модифікованого методу у разі підвищило точність оцінки зусиль створення функцій  $f_2$ ,  $f_3$  і  $f_4$  вдвічі.

#### 4.3 Аналіз переваг і недоліків методу

Існуючі особливості ре-використання окремих функцій ІС визначили напрям удосконалення правилрозрахунку значень параметрів оцінювання кількості функціональних точок окремої функції ІС в умовах ре-використання її елементів. Удосконалені правила представлені виразами (9), (10). Отримані результати пояснюються обліком у ході оцінювання збігів і відмінностей у структурних описах та найменуваннях функції  $f_{ij}$  і ре-використовується функції  $f_i$ .

Отримані результати дозволили модифікуватиметод FPA IFPUG для оцінювання зусиль ІТ-проекту створення ІС в умовах ре-використання її елементів. Суть запропонованої модифікації полягає в наступному:

- додано новий етап (Етап 3), в рамках якого здійснюється вибір та прийняття рішення про доцільність ре-використання окремої функції ІС;

- розширено набори правил розрахунку функціональної складності оцінюваної функції в Етапах 4 та 5 методу за рахунок додавання правил,

описаних виразом (9);

- розширено набір правил розрахунку повного функціонального розміру функції, що оцінюється в Етапі 6 методу за рахунок додавання правила, описаного виразом (10).

Внаслідок цієї модифікації стало можливим підвищити точність кількісного оцінювання зусиль ІТ-проектів створення ІС. Це підвищення точності пояснюється своєчасним виділенням ре-використовуваних функцій і врахуванням результатів їх ре-використання в ході створення інших функцій ІВ.

Модифікований метод був протестований під час оцінювання зусиль виконання ІТ-проекту створення функціональної завдання «Формування і ведення індивідуального плану науково-педагогічного працівника кафедри». У ході оцінювання було запропоновано ре-використовувати функцію «Формування розділу «Наукова робота» в ході створення наступних функцій:

- «Формування розділу «Методична робота»;
- "Формування розділу "Організаційна робота".

З урахуванням запропонованого рішення щодо ре-використання були розраховані оцінки зусиль на створення відповідних функцій із застосуванням модифікованого методу та вихідного методу FPA IFPLUG. Показано, що застосування оцінювання зусиль вихідного методу FPA IFPLUG призводить до значного завищення оцінок. Для розглянутого випадку оцінки порівняно з оцінками, отриманими в результаті застосування модифікованого методу, було завищено вдвічі. Отриманий результат пояснюється повним збігом структурних описів та мінімальними відмінностями найменувань функцій, що оцінюються.

Слід зазначити, що можливість оцінювання зусиль адаптацію раніше розробленого програмного забезпечення реалізації елемента ІС властива й іншим методам параметричного оцінювання ІТ-проекта. Найбільш повно цю можливість реалізовано у комплексі параметричних моделей СОСОМО II. Однак у моделях СОСОМО II для проведення оцінки зусиль на адаптацію програмного забезпечення, що використовується, необхідно використовувати як вихідні значення [3]:

- кількість тисяч рядків вихідного коду, який необхідно створити "з нуля";
- кількість тисяч рядків вихідного коду, що підлягає адаптації;
- відсоток вихідного коду, що підлягає автоматичній трансляції чи компіляції;
- значення інтегрального показника, що оцінює відсоток модифікації структури і змісту, а також витрат на інтеграцію ре-використовується коду;
- оцінку витрат на розуміння ре-використовується коду;
- значення показника оцінювання та асиміляції ре-використовується коду;
- значення показника рівня роз'єднаності проектної команди.

На відміну від моделей СОСМО II, запропонований модифікований метод дозволяє використовувати для оцінки не тільки логічні, а й концептуальні описи структурних та транзакційних елементів функцій ІС, що оцінюються. Такий підхід значно спрощує вирішення завдання щодо оцінювання зусиль на виконання ІТ-проектів створення ІС. При цьому точність як моделей СОСОМО II, так і вихідного методу FPA IFPLUG на ранніх стадіях подібних ІТ-проектів приблизно однакова внаслідок відсутності точних відомостей про параметри ІТ-проекту. У цілому ряді випадків на ранніх стадіях ІТ-проекту точність оцінювання методом FPA IFPLUG та його запропонованою модифікацією може бути навіть вищою за точність оцінювання на основі моделей СОСОМО II. Причиною є необхідність витратити на уточнення описів структурних і транзакційних елементів оцінюваної функції менше зусиль, ніж оцінювання розмірів вихідного програмного коду.

Проведена модифікація заснована на припущенні про можливість багаторазового використання структур даних окремої функції в ході створення інших функцій, опис яких повністю або частково збігаються з описами даної функції. Запропоноване рішення не залежить від виду ІТ-проекту і може розглядатися як поведінковий патерн оцінювання зусиль на створення ІВ в умовах ре-використання її елементів.

Основними обмеженнями застосування отриманих результатів дослідження у подальших прикладних роботах і теоретичних дослідженнях є:

- використання в якості основних джерел відомостей про функції і структури даних створюваної ІС діаграм потоків даних і «сутність - зв'язок»;
- використання для прийняття рішення про доцільність ре-використання функції задачі результатів рішення задачі кластеризації конфігураційних елементів і, зокрема, запропонованого в [19] способу визначення відстані між конфігураційними елементами;
- необхідність створення хоча б на концептуальному рівні діаграми "сутність - зв'язок" для аналізованого ІТ-продукту.

Головним недоліком цього дослідження є його залежність від повноти описів функціональних вимог, виявлених у ході збору та аналізу вимог до ІС. Найбільш точними оцінки із застосуванням модифікованого методу FPA будуть у разі наявності каталогу вимог до ІВ з детальним описом атрибутів кожної сутності предметної області та створюваної системи. Ця ситуація можлива, головним чином, у разі вибору для ІТ-проекту створення ІВ водоспадної моделі його життєвого циклу. Однак вибір такої моделі призводить до серйозного збільшення витрат часу створення ІС. У разі вибору спіральної або гібридної моделей життєвого циклу ІТ-проекту точність оцінювання запропонованої модифікації методу знизиться через відсутність детальних описів структурних і транзакційних елементів оцінюваних функцій ІС.

Подальший розвиток цього дослідження може здійснюватися у кількох напрямках. Так, залишається відкритим питання щодо застосування модифікованого методу в ході етапу ініціації та на ранніх етапах планування ІТ-проекту. Складність досліджень даного питання пов'язана з поданням даних про створювану ІВ та її функції у вигляді функціональних вимог. Ці вимоги можуть бути неповними, суперечити одна одній, а також можуть бути представлені у вигляді слабо формалізованих текстів природною мовою. Можливі спроби подолати зазначені труднощі можуть бути пов'язані із застосуванням засобів штучного інтелекту, що може призвести до значного збільшення витрат на застосування модифікованого методу та зробити подібне застосування недоцільним з економічної точки зору.

Іншим напрямом досліджень слід визнати можливість застосування модифікованого методу на основі не лише описів елементів бази даних ІВ, але й на основі описів структурних елементів програмного забезпечення цієї ІВ. Неясною залишається також можливість застосування модифікованого методу для оцінювання ІТ-проекту рефакторингу експлуатованої ІВ (зокрема, для випадків додавання нових, модифікації та розвитку існуючих функцій). Головна складність, що ускладнює дослідження в цих напрямках, полягає у виборі критерію або групи критеріїв, що використовуються для прийняття рішення про доцільність ре-використання окремих функцій ІС.

Окремим напрямком подальших досліджень слід визнати можливість оцінювання зусиль ІТ-проекту зі створення ІС на основі ре-використання елементів раніше виконаних ІТ-проектів. Запропонований варіант модифікації методу FPA IFPLUG ґрунтується на неявному припущенні про незмінність складу команди виконавців робіт із створення виділеної підмножини функцій ІС. Для випадку ре-використання елементів раніше виконаних ІТ-проектів дане припущення не завжди може бути дотримано.

Тож підсумовуючи даний розділ можемо сказати, що при повторному використанні методу функціональних точок

Зробивши висновки можемо сказати що в [20] досить відкрито розібрали даний метод та головне, що метод таки був використаний повторно та це дає в двічі кращі результати ніж звичайне його використання. Щодо переваг та недоліків повторне використання класичного методу функціональних точок значно підвищило точність. Основним недоліком є його залежність від повноти описів функціональних вимог, виявлених у ході збору та аналізу вимог до ІС.

Модифікований метод FPA IFPLUG дозволяє більш точно оцінювати зусилля ІТ-проектів завдяки ре-використанню елементів з раніше виконаних проектів. Використання цього методу значно підвищує точність оцінок, знижуючи зусилля на оцінювання, оскільки він дозволяє оцінювати не тільки логічні, але й концептуальні описи структурних та транзакційних елементів функцій. Модифікація методу включає додатковий етап, який дозволяє приймати

рішення щодо доцільності ре-використання окремих функцій, що робить метод більш гнучким та придатним для різних типів ІТ-проектів. Водночас, ефективність методу залежить від повноти описів функціональних вимог, зібраних на етапі аналізу. Загалом, модифікація методу FPA IFPLUG демонструє покращення точності оцінювання зусиль на створення інформаційних систем шляхом ре-використання існуючих елементів, хоча й залежить від якості та повноти описів функціональних вимог.

Отже, модифікація методу FPA IFPLUG демонструє значне покращення точності оцінювання зусиль на створення інформаційних систем шляхом ре-використання існуючих елементів. Проте, ефективність цього методу залишається залежною від якості та повноти описів функціональних вимог, зібраних на етапі аналізу.

## ВИСНОВКИ

У процесі виконання магістерської кваліфікаційної роботи було проведено дослідження класичного методу функціональних точок для використання в ІТ проектах заснованих на методології повторного використання розроблених раніше рішень, а також метод функціональних точок. Удосконалено правила розрахунку значень параметрів оцінювання кількості функціональних точок окремої функції ІС в умовах ре-використання її елементів. Суть даного удосконалення полягає у виключенні з подальшого розгляду описів структурних елементів ре-використовується функції, присутніх в описах оцінюваної функції системи. Крім того, запропонована кількісна оцінка зусиль на адаптацію ре-використовується функції до особливостей предметної області оцінюваної функції. Ця оцінка ґрунтується на результатах аналізу найменувань відповідних функцій.

Проведено аналіз особливостей процесу оцінювання та контролю проекту, аналіз зусиль ІТ – проектів та підходів удосконалення методу аналізу функціональних точок в 1 розділі.

В 2 розділі було зроблено опис методу аналізу функціональних точок, теоретичної основи для модифікації методу та сама модифікація методу функціональних точок.

В 3 розділі проведено аналіз особливостей застосування методу, а саме схема алгоритму та методика використання методу функціональних точок.

В 4 розділі проведена апробація модифікованого методу функціональних точок, загальні відомості про метод та аналіз переваг та недоліків.

Запропоновані удосконалення дозволяють підвищити точність оцінювання зусиль на створення окремих функцій у рамках ІТ-проекту створення ІС при ре-використанні окремих функцій цієї системи.

Модифіковано метод FPA IFPLUG (version 4.3.1). Суть даного методу полягає, перш за все, у додаванні спеціального етапу формування та прийняття

рішення про доцільність ре-використання окремих функцій для зменшення зусиль на створення низки інших функцій ІВ. З іншого боку, підвищення точності оцінювання зусиль створення окремих функцій ІВ, у наступних етапах методу запропоновано використовувати вдосконалені правила розрахунку значень параметрів оцінювання кількості функціональних точок. Отримана модифікація методу дозволяє підвищити точність оцінювання зусиль ІТ-проекту створення ІС в умовах ре-використання її окремих функцій.

Проведено експериментальну перевірку розробленої модифікації методу. Дана перевірка здійснена під час планування ІТ-проекту розробки функціонального завдання «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри». Як опис функцій завдання були використані такі візуальні моделі функціонального завдання, як діаграма потоків даних та діаграма «сутність – зв'язок». Отримано сумарну оцінку зусиль створення трьох функцій, описи яких майже повністю збігаються. Ця оцінка дорівнює 72 функціональних точках. Також було проведено сумарну оцінку тих самих функцій із застосуванням вихідного методу FPA IFPLUG, що становить 144 функціональних точки. Результати перевірки дозволяють стверджувати, що точність оцінювання окремих функцій завдання, які планується створити шляхом ре-використання раніше розробленої функції, підвищується вдвічі.

Результати роботи були опубліковані у [20].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Institute P. M. Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition. Project Management Institute, 2008.
2. Настанова до Зводу знань з управління проєктами. Настанова PMBOK. Сьоме видання. Стандарт з управління проєктами (2021). Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 370.
3. COCOMO II Model Definition Manual. Available at: [https://athena.ecs.csus.edu/~buckley/CSc231\\_files/Cocomo\\_II\\_Manual.pdf](https://athena.ecs.csus.edu/~buckley/CSc231_files/Cocomo_II_Manual.pdf).
4. C.E. Carbonera, K. Farias, V. Bischoff (2020). Software development effort estimation: a systematic mapping study. IET Software, 14 (4), 328-344. DOI: <https://doi.org/10.1049/iet-sen.2018.5334>
5. Timp, A. (Ed.) (2010). International Function Point Users Group (IFPUG), Function point counting practices manual, release 4.3.1. Equens SE, Netherlands, Counting Practices Committee. 546.
6. de Freitas Junior, M., Fantinato, M., Sun, V. (2015). Improvements to the Function Point Analysis Method: A Systematic Literature Review. IEEE Transactions on Engineering Management, 62(4), 495-506. doi:10.1109/TEM.2015.2453354.
7. Lavazza, L. (2017). On the Effort Required by Function Point Measurement Phases. International Journal on Advances in Software, 10 (1&2),108-120.
8. Shah, J., Kama, N., & Ismail, S.A. (2018). An Empirical Study with Function Point Analysis for Software Development Phase Method. International Conference on Software and Information Engineering. 7-11. <https://doi.org/10.1145/3220267.3220268>
9. Shah, J., & Kama, N. (2018). Extending Function Point Analysis Effort Estimation Method for Software Development Phase. Proceedings of the 2018 7<sup>th</sup> International Conference on Software and Computer Applications. 77-81. <https://doi.org/10.1145/3185089.3185137>

10. Júnior, O.D., Farias, P.P., & Belchior, A.D. (2018). A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments. *CLEI Electronic Journal*, 5 (2).
11. Zhang, K., Wang, X., Ren J., Liu, C. (2021). Efficiency Improvement of Function Point-Based Software Size Estimation With Deep Learning Model. *IEEE Access*, 9, 107124-107136. doi: 10.1109/ACCESS.2020.2998581.
12. Rohayani, H., Gaol, F.L., Soewito, B., Spits Warnars, H.L.H. (2017). Estimated measurement quality software on structural model academic system with Function Point Analysis. 2017 International Conference on Applied Computer and Communication Technologies (ComCom), Jakarta, Indonesia, 1-5. doi:10.1109/COMCOM.2017.8167085.
13. de Freitas, M., Fantinato, M., Sun, V., Thom, L.H., Garaj, V. (2020). Function Point Tree-Based Function Point Analysis: Improving Reproducibility Whilst Maintaining Accuracy in Function Point Counting. Filipe, J., Śmiałek, M., Brodsky, A., Hammoudi, S. (eds) *Enterprise Information Systems. ICEIS 2019. Lecture Notes in Business Information Processing*, vol 378. Springer, Cham. 182-209.
14. Prokopova, Z., Silhavy, P., Silhavy, R. (2019). Influence Analysis of Selected Factors in the Function Point Work Effort Estimation. Silhavy, R., Silhavy, P., Prokopova, Z. (eds) *Intelligent Systems in Cybernetics and Automation Control Theory. CoMeSySo 2018. Advances in Intelligent Systems and Computing*, vol 860. Springer, Cham. 112-124. [https://doi.org/10.1007/978-3-030-00184-1\\_11](https://doi.org/10.1007/978-3-030-00184-1_11)
15. Alkazemi B.Y., Nour M.K., Sahraoui Abd-El-K. (2014). Guidelines for Designing Reusable Software Components. *International Journal of Computer and Information Technology*, 03 (06), 1356-1361.
16. Meyer M.H., Osiyevskyy O., Libaers D., van Hugten M. (2018). Does product platforming pay off? *Journal of Product Innovation Management*, 35(1), 66-87. <https://doi.org/10.1111/jpim.12378>
17. Systems and software engineering – Software life cycle processes (ISO/IEC/IEEE 12207:2017). Available at: <https://www.iso.org/ru/standard/63712.html>

18. Ievlanov, M., Solovieva, E. (2014). Unification of methods for estimating the cost of creating modern information systems. Scientific journal &quot;Transactions of Kremenchuk Mykhailo Ostrohradskyi National University, 5/2014(88), 62-67.

19. Ievlanov, M., Vasilcova, N., Neumyvakina, O., Panforova, I. (2022). Development of a method for solving the problem of it product configuration analysis. Eastern-European Journal of Enterprise Technologies, 6/2(120), 6-19.  
<https://doi.org/10.15587/1729-4061.2022.269133>

20. Estimation of IT-project efforts for information system creation in the conditions of re-use of its functions / V. Levykin et al. *Information technology. Industry control systems*. 2024.Vol. 2, no. 2 (128). P. 6–19.  
<https://doi.org/10.15587/1729-4061.2024.301227> .

21. Методичні вказівки щодо розробки та оформлення магістерської атестаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проектами в галузі інформаційних технологій» освітньо-кваліфікаційного рівня «магістр» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2019. – 24 с.

22. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

23. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.