

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження інтуїтивних методів оцінювання змін функціональних вимог
у ІТ-проекті
(тема)

Виконав:

здобувач 2 року навчання,
групи ІУСТзм-23-1

Міщенко Олександр Анатолійович
(прізвище, ініціали)

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

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

Освітня програма Інформаційні управляючі
системи та технології
(повна назва освітньої програми)

Керівник проф. каф. ІУС Євланов М.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри


(підпис)


Костянтин ПЕТРОВ
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Навчально-науковий центр заочної форми навчанняКафедра Інформаційних управляючих системРівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)Освітня програма Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)« 09 » грудня 2024 р.**ЗАВДАННЯ**
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Міщенко Олександр Анатолійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження інтуїтивних методів оцінювання змін функціональних вимог у ІТ-проєктізатверджена наказом університету від 03 грудня 2024 р. № 205 Стз2. Термін подання здобувачем роботи до екзаменаційної комісії 16 січня 2025 р.3. Вихідні дані до роботи Фреймворк управління змінами, існуючі методи інтуїтивного оцінювання ІТ-проєктів, ІТ-проєкт розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»4. Перелік питань, що потрібно опрацювати в роботі Огляд і аналіз особливостей оцінювання змін у ІТ-проєкті; вдосконалення інтуїтивного методу оцінювання змін функціональних вимог у ІТ-проєкті; розробка інформаційної технології оцінювання змін функціональних вимог до інформаційної системи; практичне використання ансамблевого методу Size Ordering Rule.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз фреймворку управління змінами	09.12.2024 р. – 11.12.2024 р..	Виконано
2	Аналіз систем класифікації запитів на зміни	12.12.2024 р. – 15.12.2024 р.	Виконано
3	Аналіз методів оцінювання запитів на зміни	16.12.2024 р. – 19.12.2024 р.	Виконано
4	Аналіз існуючих методів інтуїтивного оцінювання	20.12.2024 р. – 25.12.2024 р.	Виконано
5	Розробка ансамблевого методу Size Ordering Rule	26.12.2024 р. – 29.12.2024 р.	Виконано
6	Розробка інформаційної технології оцінювання	30.12.2024 р. – 05.01.2025 р.	Виконано
7	Апробація отриманих результатів	06.01.2025 р. – 08.01.2025 р.	Виконано
8	Оформлення пояснювальної записки та графічного матеріалу	09.01.2025 р. – 10.01.2025 р.	Виконано
9	Підготовка презентації	11.01.2025 р. – 12.01.2025 р.	Виконано
10	Подання роботи для перевірки на плагіат	13.01.2025 р.	Виконано
11	Надання роботи на підпис керівнику	14.01.2025 р.	Виконано
12	Попередній захист роботи	14.01.2025 р.	Виконано
13	Надання роботи на рецензію	15.01.2025 р.	Виконано
14	Надання роботи на підпис завідувачу кафедри	16.01.2025 р.	Виконано
15	Захист кваліфікаційної роботи	17.01.2025 р.	Виконано

Дата видачі завдання 09 грудня 2024 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Євланов М.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 173 стор., 18 рис., 19 табл., 34 джерела, 5 додатків.

ЗАПИТ НА ЗМІНУ, ЕКСПЕРТ, ІНТУЇТИВНЕ ОЦІНЮВАННЯ, ІНФОРМАЦІЙНА СИСТЕМА, МЕТОД BIG/SMALL/UNCERTAIN, МЕТОД ORDERING RULE, ФУНКЦІОНАЛЬНА ВИМОГА.

Метою магістерської кваліфікаційної роботи є розробка інформаційної технології оцінювання змін функціональних вимог до системи у ІТ-проєкті створення ІС.

Об'єктом дослідження в рамках даної роботи є робота «Окресліть масштаби змін» підпроцесу «Формулювання змін» процесу «Управління змінами».

Предметом дослідження є методи інтуїтивного оцінювання змін функціональних вимог у ІТ-проєкті.

В ході виконання роботи розроблено ансамблевий метод інтуїтивного оцінювання Size Ordering Rule, який вдосконалює існуючі методи інтуїтивного оцінювання. Розроблено також інформаційну технологію оцінювання змін функціональних вимог до інформаційних систем управління підприємствами та організаціями. Проведено практичну апробацію отриманих результатів на прикладі ІТ-проєкту розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри».

Отримані результати можуть бути використані для створення спеціалізованих інформаційних систем та технологій управління ІТ-проєктами розробки та супроводження інформаційних систем управління підприємствами та організаціями.

ABSTRACT

The explanatory note to the qualification work: 173 p., 18 fig., 19 tabl., 34 sources, 5 appendixes.

REQUEST FOR CHANGE, EXPERT, INTUITIVE ESTIMATION, INFORMATION SYSTEM, BIG/SMALL/UNCERTAIN METHOD, ORDERING RULE METHOD, FUNCTIONAL REQUIREMENT.

The purpose of this work is to develop an information technology for estimating changes in functional requirements for the system in an IT project for creating an IS.

The object of research within the framework of the master's qualification work is the work «Describe the scope of changes» of the sub process «Formulation of changes» of the process «Change management».

The subject of the research is methods of intuitive estimation of changes in functional requirements in the IT project.

During the work, an ensemble method of intuitive estimation Size Ordering Rule was developed, which improves existing methods of intuitive estimation. An information technology for estimating changes in functional requirements for information systems of management of enterprises and organizations was also developed. Practical testing of the obtained results was carried out on the example of an IT project for developing a functional task «Formation and maintenance of an individual plan of a scientific and pedagogical employee of the department».

The results obtained can be used to create specialized information systems and technologies for managing IT projects, developing and maintaining information systems for managing enterprises and organizations.

ЗМІСТ

	С.
Скорочення та умовні позначки.....	9
Вступ.....	10
1 Огляд і аналіз особливостей оцінювання змін у ІТ-проєкті.....	12
1.1 Аналіз актуального фреймворку управління змінами.....	12
1.2 Дослідження і аналіз класифікацій запитів на зміни у ІТ-проєктах.....	18
1.3. Аналіз сучасних досліджень в галузі оцінювання запитів на зміни.....	28
1.4 Мета і задачі магістерської класифікаційної роботи.....	35
2 Вдосконалення інтуїтивного методу оцінювання змін	
функціональних вимог у ІТ-проєкті.....	38
2.1 Порівняльний аналіз інтуїтивних методів оцінювання	
зусиль на реалізацію функціональних вимог до	
інформаційної системи.....	38
2.1.1 Дослідження особливостей інтуїтивного	
оцінювання функціональних вимог до	
інформаційної системи.....	38
2.1.2 Метод T-Shirt Sizes.....	39
2.1.3 Метод Planning Poker.....	40
2.1.4 Метод Bucket System.....	42
2.1.5 Метод Dot-voting.....	43
2.1.6 Метод Maximum Size or Less.....	43
2.1.7 Метод Big/Small/Uncertain.....	44
2.1.8 Метод Ordering Rule.....	45
2.1.9 Результати порівняльного аналізу розглянутих методів	
інтуїтивного оцінювання за обраними критеріями.....	46
2.2 Вдосконалення методу Big/Small/Uncertain для оцінювання	

змін функціональних вимог до інформаційної системи.....	50
2.3 Висновки до другого розділу.....	54
3 Розробка інформаційної технології оцінювання змін	
функціональних вимог до інформаційної системи.....	55
3.1 Розробка моделі потоків даних інформаційної технології.....	55
3.1.1 Визначення вимог до інформаційної технології в цілому.....	55
3.1.2 Визначення функціональних вимог до інформаційної	
технології.....	59
3.1.3 Визначення вимог до видів забезпечень інформаційної	
технології.....	63
3.2 Розробка елементів інформаційного забезпечення	
інформаційної технології.....	66
3.3 Розробка елементів алгоритмічного забезпечення	
інформаційної технології.....	83
3.4 Розробка елементів програмного забезпечення інформаційної	
технології.....	89
4 Практичне використання ансамблевого методу Size Ordering Rule.....	101
4.1 Опис основних особливостей змінюваної функціональної задачі.....	101
4.2 Опис запитів на зміни функцій задачі.....	103
4.3 Хід та результати оцінювання запропонованих запитів	
на зміни функціональних вимог до задачі.....	105
4.4 Висновки до четвертого розділу.....	117
Висновки.....	118
Перелік джерел посилання.....	120
Додаток А Діаграма декомпозиції другого рівня моделі потоків	
даних інформаційної технології оцінювання змін	
функціональних вимог до інформаційної системи.....	125
Додаток Б Фрагменти логічної схеми даних інтелектуальної	
інформаційної технології прискореної розробки	

інформаційної системи.....	128
Додаток В Публікації функціональних вимог до прикладного програмного забезпечення інформаційної технології оцінювання змін функціональних вимог до інформаційної системи.....	130
Додаток Г Опис архітектури функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри».....	139
Додаток Д Графічний матеріал кваліфікаційної роботи.....	150

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ІС – інформаційна система

ІТ- інформаційна технологія

ПрО – предметна область

СУБД – система управління базами даних

DFD – data flow diagram

ERD – entity-relationship diagram

KAMP – Karlsruhe Architectural Maintainability Prediction

ML – machine learning

RFC – request for change

ВСТУП

Експлуатація сучасних інформаційних систем (ІС) управління підприємствами та організаціями вимагає виконання великого обсягу робіт із супроводження та вдосконалення таких систем. Особливо важливими ці роботи стають у випадку застосування цих ІС на об'єктах із нестабільними бізнес-процесами, правила виконання яких можуть змінюватися з часом. Подібна ситуація стає розповсюдженою в умовах воєнного стану навіть для тих підприємств та організацій, бізнес-процеси яких донедавна вважались стабільними.

Тому виникає необхідність управління значним обсягом робіт, пов'язаних із визначенням, плануванням та управлінням реалізацією запитів на зміни (Requests For Changes, RFC), які виникають внаслідок зміни вимог, що висуваються до ІС. Серед таких RFC особливу увагу приділяють RFC функціональних вимог до ІС, оскільки реалізація таких RFC може викликати велику кількість змін не тільки у тій функції, до якої висувається RFC, а й у інших функціях змінюваної ІС.

Необхідно визнати, що RFC можуть виникати не тільки під час експлуатації ІС управління підприємствами та організаціями, а й під час розробки таких ІС в ході виконання відповідних ІТ-проектів. Тому виникає необхідність дослідити методи та інформаційні технології (ІТ) вирішення задачі оцінювання RFC, які могли б використовуватися одночасно і як доповнення сучасних ІС та ІТ управління ІТ-проектами, і як самостійні рішення, що можуть бути застосовані під час безпосередньої експлуатації ІС управління підприємствами та організаціями.

Оскільки переважна більшість сучасних ІТ-проектів використовує принципи методології Agile, основну увагу рекомендовано приділяти так званим інтуїтивним методам оцінювання, які спираються на використання інтуїції та

досвіду фахівців, що виступають як експерти під час вирішення відповідних задач оцінювання. Застосування інтуїтивних методів забезпечує мінімальні витрати часу та зусиль на вирішення задачі оцінювання RFC із гарантією отримання достатньо точних результатів оцінювання.

Але проблеми застосування інтуїтивних методів оцінювання для визначення основних характеристик RFC функціональних вимог до ІС управління підприємствами та організаціями на даний час залишаються майже недослідженими. Тому проведення дослідження, присвяченого вирішенню задачі вибору методу інтуїтивного оцінювання, який найкраще підходить для вирішення задачі оцінювання RFC функціональних вимог до ІС, та розробки відповідної ІТ є актуальним з теоретичної та прикладної точок зору.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є робота «Окресліть масштаби змін» підпроцесу «Формулювання змін» процесу «Управління змінами». Предметом дослідження є методи інтуїтивного оцінювання змін функціональних вимог у ІТ-проєкті.

Метою роботи є розробка ІТ оцінювання змін функціональних вимог до системи у ІТ-проєкті створення ІС, яка дозволила б підвищити точність оцінювання обсягу змін функціональних вимог, що пропонуються під час виконання ІТ-проєкту.

Магістерська кваліфікаційної роботи виконана згідно з методичними вказівками 2021 року щодо розробки та оформлення магістерської кваліфікаційної роботи за спеціальністю 122 Комп'ютерні науки та освітньою програмою «Інформаційні управляючі системи та технології» [1].

1 ОГЛЯД І АНАЛІЗ ОСОБЛИВОСТЕЙ ОЦІНЮВАННЯ ЗМІН У ІТ-ПРОЄКТІ

1.1 Аналіз актуального фреймворку управління змінами

Збереження актуальності в сучасному бізнес-середовищі є фундаментальним завданням для всіх організацій. Актуальність передбачає здатність реагувати на потреби та бажання стейкхолдерів. Для цього потрібно постійно оцінювати пропозиції на користь стейкхолдерів, швидко реагувати на зміни та діяти як агенти змін [2]. Прикладом таких агентів змін є проєкти.

Управління змінами – це комплексний, циклічний і структурований підхід до зміни окремих осіб, груп і організації з поточного стану в майбутній стан із передбачуваними бізнес-вигодами [2], [3]. Цей підхід зараз описано у вигляді фреймворку, схему якого наведено на рис. 1.1 [3]. Під фреймворком в даному випадку будемо розуміти конвенції, принципи і методи управління змінами в галузі управління програмами, портфелями проєктів та окремими проєктами. Цей фреймворк охоплює тему стратегічної гнучкості як відображення здатності організації відчувати зовнішні або внутрішні загрози і можливості та реагувати на них. Стратегічна гнучкість визначається як здатність бізнесу завчасно оцінювати і реалізувати зміни бізнес-середовища та демонструвати стійкість до непередбачених змін [3].

Фреймворк управління змінами розглядає перехід з поточного стану в майбутній стан із передбачуваними бізнес-вигодами як процес «Управління змінами» із такими підпроцесами [3]:

а) формулювання змін – підпроцес, який здійснюється шляхом виявлення та уточнення потреби у змінах, оцінки готовності до змін та окреслення масштабу змін;

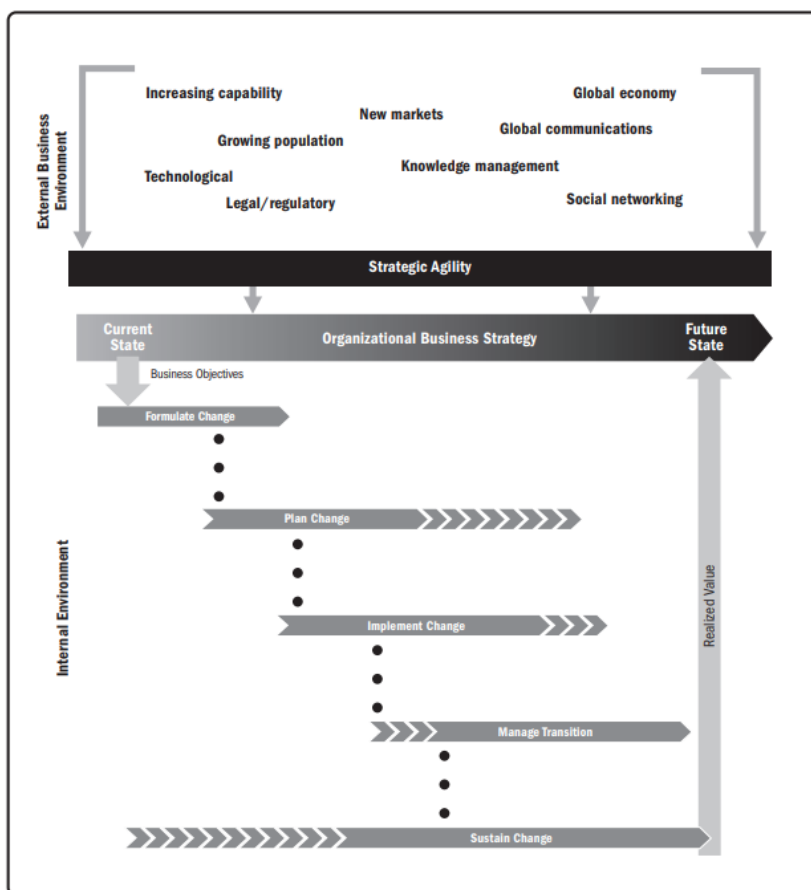


Рисунок 1.1 – Схема актуального фреймворку управління змінами

б) планування змін – підпроцес, який здійснюється шляхом визначення підходу до змін і планування залучення зацікавлених сторін як переходів та інтеграції;

в) впровадження змін – підпроцес, який здійснюється шляхом підготовки організації до змін, мобілізації зацікавлених сторін та досягнення результатів проекту;

г) управління переходом до змін – підпроцес, який здійснюється шляхом переведення результатів у бізнес-операції, вимірювання рівня впровадження та результатів і переваг змін, а також коригування плану для усунення розбіжностей;

д) підтримка змін на постійній основі – підпроцес, який здійснюється через спілкування, консультації та представництво зацікавлених сторін, проведення смислотворчої діяльності та вимірювання реалізації вигод.

Схему життєвого циклу управління змінами наведено на рис. 1.2 [3]. За цією схемою кожен з розглянутих підпроцесів управління змінами у [3] запропоновано розглядати як множину окремих робіт, кожна з яких виконується у встановленому фреймворком порядку (але не обов'язково суворо послідовно одна за одною). Весь процес управління змінами та його підпроцеси запропоновано описувати з використанням ітераційної моделі життєвого циклу з врахуванням можливості постійного виникнення адаптивних змін у відповідь на зміни обставин [3].

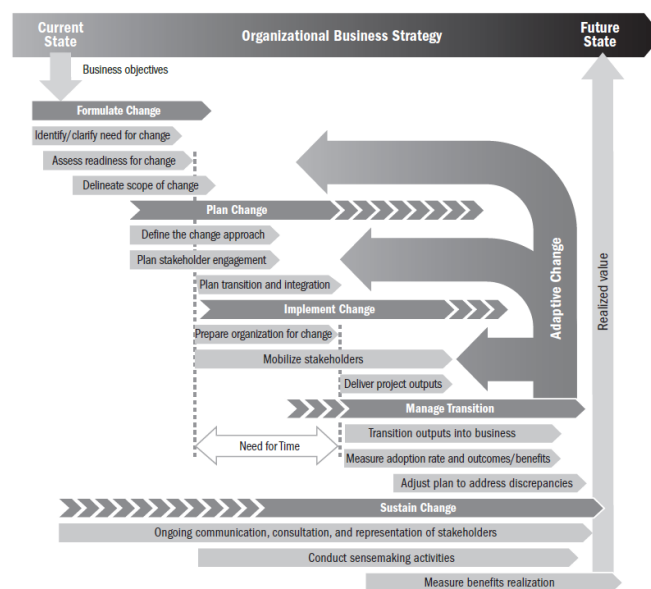


Рисунок 1.2 – Схема життєвого циклу управління змінами

Підпроцес «Формулювання змін» представлено як множину таких робіт [3]:

– робота «Визначте/уточніть потребу в змінах» (встановить потребу в змінах і внесок, необхідний для постійного зростання та сталої конкурентної переваги проєкту та продукту);

– робота «Оцініть готовність до змін» (оцінить готовність до змін організаційних систем, структур, культури та людей, на яких впливають ці зміни або яких вони потребують);

– робота «Окресліть масштаби змін» (уточніть очікувані результати змін і визначте обсяг і заходи, необхідні для успішних змін).

Підпроцес «Планування змін» представлено як множину таких робіт [3]:

– робота «Визначте підхід до змін» (узгодьте підхід до змін із культурою організації);

– робота «Плануйте залучення зацікавлених сторін» (визначте та залучіть усі зацікавлені сторони, внутрішні і зовнішні, на які впливають зміни або які зацікавлені в результатах, і сплануйте їхню постійну участь);

– робота «Плануйте перехід та інтеграцію» (розробіть план, який включає всі дії, необхідні для досягнення цілей та інтеграції з бізнес-операціями).

Підпроцес «Впровадження змін» представлено як множину таких робіт [3]:

– робота «Підготуйте організацію до змін» (визначте сфери, де потрібна конкретна підтримка, і запровадьте заходи підтримки);

– робота «Мобілізація зацікавлених сторін» (повідомте зацікавленим сторонам про кінцеві цілі змін і дозвольте їм брати активну участь у прийнятті рішень, що впливають на зміни);

– робота «Надавати результати проєкту» (організуйте отримання продуктів, послуг та інших результатів як результатів ініціації змін).

Підпроцес «Управління переходом до змін» представлено як множину таких робіт [3]:

– робота «Перетворення результатів у бізнес» (під час процесу змін, у міру досягнення результатів, продовжуйте поточний процес переходу, інтегруючи нові можливості у звичайний бізнес);

– робота «Виміряйте швидкість адаптації та результати/вигоди» (вимірюйте результати на бізнес-рівні не лише з точки зору результатів продукту, але й з точки зору результатів ефективності);

– робота «Відкоригуйте план, щоб усунути розбіжності» (команді необхідно регулярно коригувати план, щоб врахувати обставини, що змінюються або розвиваються).

Підпроцес «Підтримка змін на постійній основі» представлено як множину таких робіт [3]:

- робота «Постійне спілкування, консультації та представництво зацікавлених сторін» (успіх змін підкріплюється двостороннім спілкуванням і консультаціями із зацікавленими сторонами);

- робота «Проводьте сенсосмислення» (підтримуйте постійні розмови та виконуйте соціальні практики, які дозволяють людям зрозуміти, що відбувається під час змін);

- робота «Вимірювання реалізації переваг» (виміряйте успіх змін через їхній вплив на організацію).

Цей фреймворк є загальним і повинен уточнюватися згідно з кожним з варіантів проєктного управління, а саме: управління окремими проєктами, управління портфелями проєктів та управління програмами. Так, управління змінами в управлінні окремими проєктами розглядається як балансування між двома групами факторів, які впливають на проєкт. До першої групи цих факторів належать межі проєкту та продукту, моделі, методи та механізми управління проєктом. До другої групи факторів належать необхідний рівень гнучкості проєкту та продукту, адаптація проєкту та продукту до зовнішніх та внутрішніх обставин, що постійно змінюються [3].

Діяльність з управління змінами окремого проєкту охоплює всі фази життєвого циклу управління змінами, показано на рисунку 1.2. Якщо проєкт не є частиною великої програми, існує прямиий інтерфейс із керуванням портфелем окремих проєктів, які виконуються підприємством, під час виконання підпроцесу «Формулювання змін» [3].

Застосування розглянутого загального фреймворку управління змінами під час управління таким різновидом проєктів, як ІТ-проєкти, вимагає приділення особливої уваги підпроцесу «Формулювання змін» і, зокрема, роботі «Визначте/уточніть потребу в змінах». Ця увага обумовлена тим, що саме ця робота є першою роботою, в результаті виконання якої ІТ-проєкт та система

управління цим проектом отримує інформацію про необхідність зміни та перший опис цієї зміни.

Але, на жаль, існуючий фреймворк управління змінами, розглянутий у [3], та стандарти і зведення знань з управління проектами та, зокрема, ІТ-проектами [3], [4] не дають конкретних рекомендацій з виконання підпроцесу «Формулювання змін». Так, у [3] вказано два таких варіанти виконання цього підпроцесу:

а) якщо проект є частиною програми, визначення та роз'яснення потреби в змінах повинно бути добре розробленою і сформульованою частиною статуту проекту;

б) якщо проект не є частиною програми, необхідно під час управління портфелем окремих проектів підприємства виконати відповідні дії, щоб переконатися, що уточнення потреб в змінах виконано до завершення статуту проекту.

Таким чином, незалежно від особливостей ІТ-проекту, потреби в змінах та перші описи можливих змін за існуючим фреймворком управління змінами повинні бути сформульовані до завершення робіт зі складання та ухвалення статуту ІТ-проекту. При цьому роботу «Окресліть масштаби змін» пропонується розглядати як різновид робіт з формування вимог, які висуваються до ІТ-продукту та ІТ-проекту перед початком планування та виконання цього проекту [3]. Таке позиціонування підпроцесу «Формулювання змін» і роботи «Окресліть масштаби змін» значно обмежує можливість внесення змін до ІТ-проекту під час його планування та виконання і, відповідно, зменшує гнучкість проекту та можливості його адаптації до обставин, що постійно змінюються.

Слід зазначити, що існуючий фреймворк управління змінами передбачає можливість виникнення змін під час виконання проекту. Зокрема, за цим фреймворком, керівникам проектів може знадобитися виконати завдання, які виходять за межі встановлених меж цього проекту. В ідеалі ці додаткові завдання мають бути частиною узгодженого обсягу проекту, щоб забезпечити успішне

виконання змін. Інакше керівник проєкту повинен попередити спонсора проєкту та вимагати зміни обсягу проєкту [3]. Таким чином, виходить замкнене коло: керівник проєкту повинен прийняти рішення про виконання роботи «Окресліть масштаби змін», базуючись на оцінках обсягу ІТ-проєкту до та після внесення зміни, яка пропонується. Але ці оцінки неможливо розрахувати, не знаючи зміст зміни, яка пропонується, тобто не виконавши роботу «Окресліть масштаби змін».

Проведений аналіз актуального фреймворку управління змінами дозволяє зробити такі висновки:

- він може бути використаний як традиційними, так і Agile-методологіями управління ІТ-проєктами;
- він визнає підпроцес «Формулювання змін» як сукупність дій, які забезпечують формування першого уявлення про зміну, що пропонується;
- він рекомендує розглядати роботу «Окресліть масштаби змін» як елемент робіт з формування та узгодження статуту проєкту під час ініціації ІТ-проєкту;
- він не дає жодних рекомендацій з визначення особливостей роботи «Окресліть масштаби змін» на подальших стадіях життєвого циклу проєкту.

1.2 Дослідження і аналіз класифікацій запитів на зміни у ІТ-проєктах

Для того, щоб можна було оцінити зміну обсягу ІТ-проєкту після внесення зміни, яка пропонується, треба визначити, якої саме частини проєкту стосується ця зміна. Тому виникає необхідність дослідити і проаналізувати існуючі класифікації запитів на зміни (request for change, RFC), які можуть локалізувати місце впливу RFC на обсяг ІТ-проєкту.

Подібне дослідження було проведено у [5]. Результати цього дослідження можна розділити на три групи.

Перша група – це класифікації RFC, які виникли та виникають внаслідок узагальнення досвіду управління змінами окремих українських ІТ-компаній. Такі класифікації є спробою формалізувати різновиди RFC, базуючись, зазвичай, на таких результатах [5]:

- а) досвід роботи окремої української ІТ-компанії;
- б) досвід управління змінами в ІТ-проєктах, які належать до одного й того ж різновиду;
- в) досвід управління змінами окремих фахівців з менеджменту ІТ-проєктів, аналітиків ІТ-проєктів та бізнес-аналітиків.

Типовим прикладом класифікації з першої групи є класифікація, запропонована у [6]. Слід зазначити, що автор цієї класифікації позиціонує себе як бізнес-аналітик, який має сім років досвіду розробки програмних продуктів. За цією класифікацією, уся множина RFC поділяється на такі класи:

- зміни системної логіки/поведінки;
- зміни скоупу (під цим терміном тут слід розуміти обсяг, межі або масштаб ІТ-проєкту або ІТ-продукту);
- зміни в назвах/формулюваннях;
- зміни в інтерфейсі користувача;
- технічні зміни (інтеграції, мови програмування тощо).

Як можливий окремий клас RFC у [6] пропонується розглядати запит на видалення елементів зі скоупу.

Але запропонована у [6] класифікація характеризується тим, що вона не використовується навіть самим автором під час визначення особливостей виконання робіт процесу «Управління змінами». Так, у [6] представлено візуальну модель кастомного процесу управління змінами, адаптованого під команди, з якими працював автор. Але ця модель визнає головною проблемою прийняття рішення щодо подальшого управління зміною ІТ-проєкту проблему відділення потреб зацікавлених сторін проєкту у змінах від потреб тих самих сторін у ліквідації знайдених програмних помилок. Далі пропонуються описи дій

з управління RFC, які незначно відрізняються від загальних рекомендацій загального фреймворку управління змінами за ступенем деталізації. Тому можна зробити висновок, що запропонований варіант класифікації є декларативним та суб'єктивним, тому що він не впливає на виконання робіт процесу управління змінами та підпроцесу «Формулювання змін» [5].

Друга група результатів – це класифікації RFC, які виникли та виникають внаслідок узагальнення досвіду закордонних ІТ-компаній. На відміну від результатів першої групи, ці класифікації є результатом наукового узагальнення одного з таких варіантів досвіду [5]:

- а) досвід управління змінами на окремій стадії ІТ-проектів;
- б) досвід управління змінами в ІТ-проектах, які належать до одного й того ж різновиду;
- в) узагальнений досвід управління змінами різних ІТ-проектів.

Як приклад класифікації RFC, яка використовується для управління змінами на окремій стадії ІТ-проекту, пропонується розглянути запропоновану у [7] класифікацію технічних (або інженерних) змін. Така зміна у [8] визначається як зміна деталей, креслень або програмного забезпечення, які вже були випущені в процесі розробки продукту. Ця зміна може бути будь-якого розміру або типу; вона може включати будь-яку кількість людей і тривати будь-який час [7], [8]. Запит на технічну зміну визначається як RFC, створений для інформування про вихід проекту або продукту за рамки цільових значень [9]. За іншим визначенням, запит на технічну зміну – це запит на технічну зміну стану моделі продукту, яка вже була випущена для подальшої розробки або виробництва [7].

Для вирішення задачі класифікації авторами у [7] було розроблено атрибутивну модель запиту на технічну зміну за результатами дослідження літературних джерел та великої кількості текстів запитів на технічні зміни. Цю модель наведено на рис. 1.3.

	Attribute	Characteristic value					
Object of change	Concerned object	Document	Component	Process	Module	Equipment	Product
	Concerned object specification	Physical Specification	Functionality	Software	Procedures	Information	
	Extend of change*	Document	Component	Process	Module	Equipment	Product
	Volume of change	Single position		Group		All	
Properties	Phase of life-cycle	Development	Prototyping / Series start-up	Series production	Use phase		
	Architecture	Modular		Partly integrated	Integrated		
	Complexity	Low		Medium	High		
Influencing factors	Root of change	Customer requirement	Law / Certification	Safety	Cost	Technology	Productivity
	Reason of change	Current situation faulty		Optimization of the current situation	Target requirements changed	Incorrect target requirements	
	Urgency	Change to next release		Important, short-term change		Immediate change	
	Concerned Stakeholder	R&D	R&D and other departments		Involvement of the management	Involvement of external stakeholders	
Effect	Effect on cost	Decreasing		Slightly decreasing	Slightly increasing		Increasing
	Effect on time schedules	Low		Medium	High		
	Effect on the product	None	Necessary correction	Fulfillment of individual customer requirements	Improvement of functionality	Quality improvement	

Descriptive attributes
 Evaluating attributes (expert knowledge required)

*The CVs of concerned object and extend of change are the same. The difference is, that concerned object identifies the objects from where the change originates, while extend of the change describes objects that are impacted by the change of the object. Example: If the dimension of a component (=object) is changed, most likely the drawing document will be changed too (=extend of change)

Рисунок 1.3 – Атрибутивна модель запиту на технічну зміну

За моделлю, наведеною на рис. 1.3., кожен запит на технічну зміну може бути описаний множиною атрибутів, які групуються у такі підмножини: «Об'єкт зміни», «Властивості [об'єкту зміни]», «Фактори, які впливають [на технічну зміну]» та «Ефекти [технічної зміни]». У кожній з цих підмножин виділяються два типи атрибутів: описові та оціночні. Описові атрибути можна чітко ідентифікувати за допомогою тексту запиту на технічні зміни, тоді як оціночним атрибутам потрібно призначити контекст або встановити їхні значення за допомогою експертів [7].

Для використання моделі, наведеної на рис. 1.3, у [7] було розроблено рішення задачі автоматичної класифікації запитів на технічні зміни. Загальне графічне представлення цієї задачі наведено на рис. 1.4 [7].

Для реалізації цієї задачі у [7] запропоновано формалізувати опис запиту на технічну зміну як кортеж таких текстових атрибутів:

- а) «Причина зміни»;
- б) «Компонент/агрегат»;
- в) «Опис поточної ситуації»;
- г) «Вигода/ціль зміни»;
- д) «Запропонована реалізація».

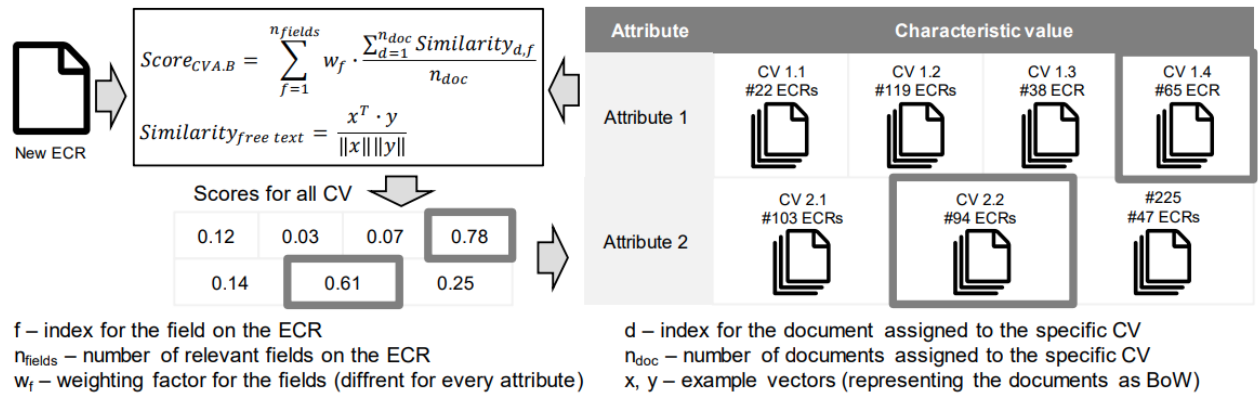


Рисунок 1.4 – Графічне представлення задачі автоматичної класифікації запитів на технічні зміни

Апробацію інформаційної технології (ІТ), яка реалізувала цю задачу класифікації, проводили на 40 запитах на технічні зміни. За результатами апробації було встановлено, що ця технологія досягла загальної точності вирішення задачі класифікації запитів на технічні зміни 79 %, а максимальна точність дорівнювала 90 % [7].

Як інший приклад класифікації RFC, яка використовується для управління змінами на окремій стадії ІТ-проєкту, пропонується розглянути запропоновану у [10] онтологічну модель. Ця модель описує класи RFC, які виникають під час змін у системних вимогах, висунутих до створюваного ІТ-продукту. Модель представлено у вигляді діаграми класів, яка відображає ієрархію класів RFC системних вимог (див. рис. 1.5). Специфікації значень важливих елементів цієї онтологічної моделі наведено у табл. 1.1.

Як приклад класифікації RFC, яка використовується для управління змінами в ІТ-проєктах, які належать до одного й того ж різновиду, пропонується розглянути класифікацію, яку запропонували використовувати під час управління змінами в ІТ-проєктах розробки мобільних застосунків. Схема життєвого циклу розробки релізу мобільного застосунку із впровадженою технологією управління змінами наведено на рис. 1.6 [11]. На цьому рисунку синьою пунктирною лінією виділено основу технології управління змінами та її інтерфейси, які передають встановлені зміни до інструментальних засобів, що використовуються на різних стадіях відповідного ІТ-проєкту. Схему підходу, який знаходиться в основі цієї технології, наведено на рис. 1.7 [11].

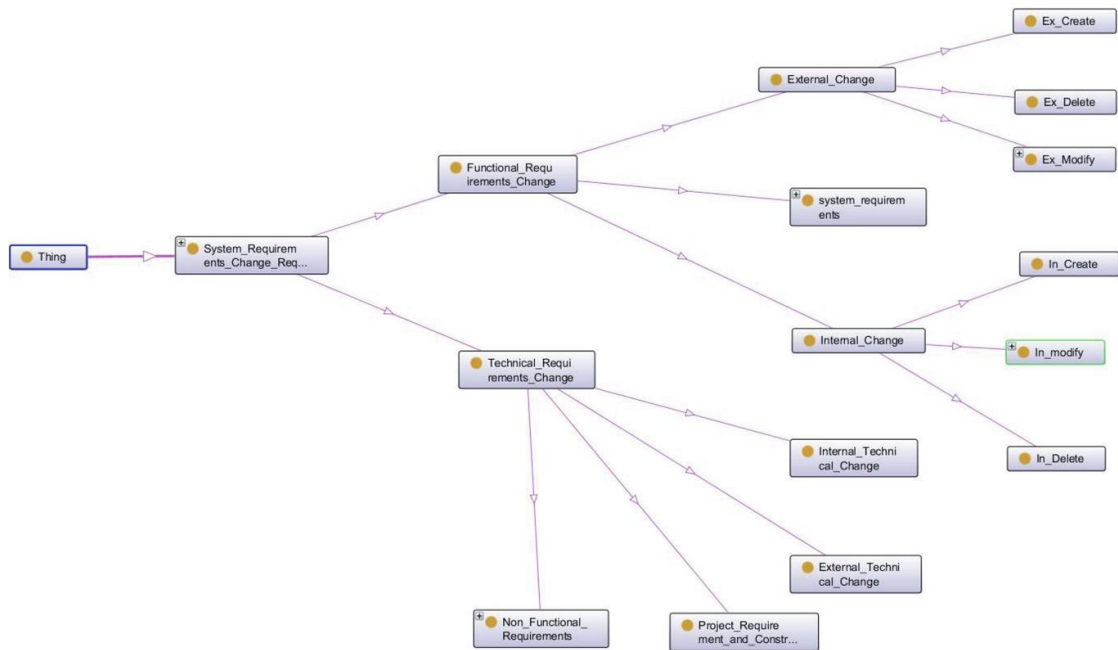


Рисунок 1.5 – Онтологічна модель класифікації запитів на зміни системних
ВИМОГ

Таблиця 1.1 – Специфікації класів онтологічної моделі

Клас	Опис
System requirements change requests (RFC системних вимог)	Запити майбутніх користувачів на зміни системних вимог зі сховища вимог
Functional change (Функціональна зміна)	Запити майбутніх користувачів на зміни, які впливають на функціональні вимоги користувачів
Technical change (Технічна зміна)	Запити майбутніх користувачів на зміни, які впливають на нефункціональні вимоги або проєктні вимоги та обмеження
System requirements (Системні вимоги)	RFC комунікаторів (внутрішніх (системні аналітики, розробники, проєктувальники тощо) або зовнішніх (оператори, користувачі, керівники тощо))

Кінець табл. 1.1

Клас	Опис
Delete (Видалити)	Містить вимоги, які необхідно видалити
Modify (Модифікувати)	Містить вимоги, які необхідно модифікувати
Create (Створити)	Містить вимоги, які необхідно створити

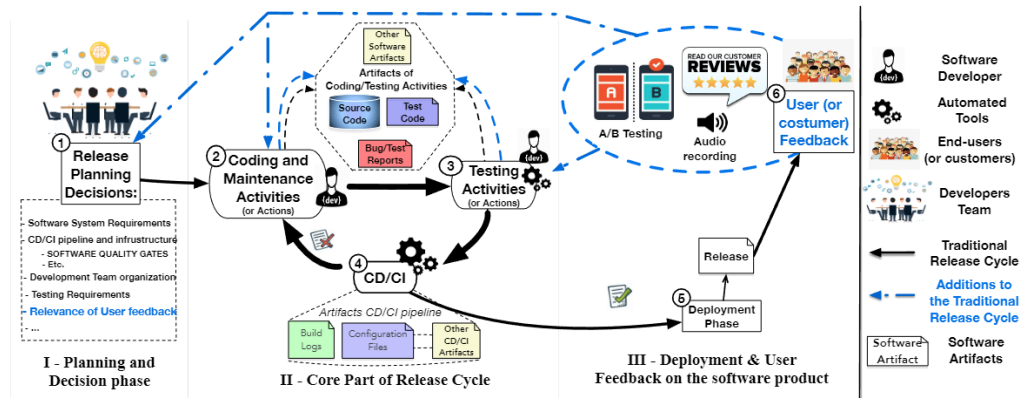


Рисунок 1.6 – Схема життєвого циклу розробки релізу мобільного застосунку із вбудованою системою управління змінами

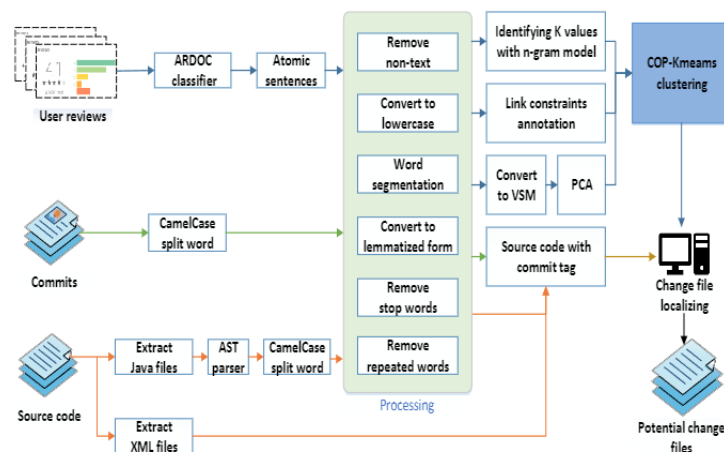


Рисунок 1.7 – Схему підходу, який знаходиться в основі запропонованої у [11] технології управління змінами

Основними обмеженнями системи управління змінами на основі запропонованої на рис. 1.6 схеми життєвого циклу є [11]:

- низька точність класифікації та кластеризації через низьку якість відгуків користувачів;
- труднощі в роботі з різними словниками, які використовуються для опису досвіду використання користувачами застосунків;
- існування розривів в словниках ІТ-проекту між термінами з відгуків користувачів і позначеннями артефактів програмного забезпечення.

Тому в запропонованій технології (див. рис. 1.7) для подолання цих обмежень запропоновано поділити усі можливі відгуки користувачів на чотири такі категорії [11]:

- а) «Надання інформації» (речення, які інформують або оновлюють інформацію для користувачів або розробників щодо аспекту, пов'язаного з програмою);
- б) «Пошук інформації» (інформація про тих, хто намагається отримати інформацію або допомогу);
- в) «Запит на функцію» (ідеї, пропозиції або потреби щодо покращення функціональності та продуктивності програми);
- г) «Виявлення проблеми» (речення, що описують проблеми з програмами або їхню несподівану поведінку).

Для перетворення інформації з відгуків користувачів у RFC у [11] запропоновано вирішити задачу класифікації відгуків користувачів, виділяючи для подальшої обробки тільки ті відгуки, які слід віднести до третьої або четвертої з наведених вище категорій. Для цього застосовано ARDOC-класифікатор, який перетворює відгуки користувачів на окремі речення, а потім класифікує ці речення в одну з чотирьох вищезгаданих категорій.

Недоліком такої класифікації є надто примітивний поділ усіх RFC на два великих класи. Тому підмножина відгуків користувачів, які за допомогою ARDOC-класифікатора визнали такими, що належать до третьої або четвертої з розглянутих вище категорій, далі обробляється шляхом вирішення задачі кластеризації. Результатом вирішення цієї задачі є розподіл усіх відгуків

користувачів з вихідної підмножини на окремі кластери, які визначають актуальні для поточної ітерації життєвого циклу релізу мобільного застосунку класи RFC. Для вирішення задачі кластеризації у [11] запропоновано використовувати алгоритми сімейства k-means із додатковими обмеженнями.

Як приклад класифікації RFC, яка враховує узагальнений досвід управління змінами різних ІТ-проектів, пропонується розглянути класифікацію RFC, що формуються під час ревіювання програмного коду. Для цього у [12] було проаналізовано 10 різних ІТ-проектів, які виконувалися за період 2012-2017 рр. за різноманітними замовленнями та розрізнялися значеннями характеристик. Крім того, автори [12] спиралися на результати визначення таксономії RFC, яку було розроблено у [13]. Результатом дослідження [12] є скоригована модель змін, які виникають під час ревіювання коду. Цю модель (з виключенням розгорнутих описів класів RFC) наведено у табл. 1.2.

Таблиця 1.2 – Скоригована модель змін, які виникають під час ревіювання коду

Artifact	Activity	Category	Topic	Detailed Change
1	2	3	4	5
Production & Test Code	Maintainability & Perfective Maintenance	Documentation (D)	- Textual Documentation	(D.1) - Naming. (D.2) - Comments. (D.3) - License Header. (D.4) - Typos. (D.5) - Other.
			- Language Supported Documentation	(D.6) - Immutability. (D.7) - Visibility.
		Style (S)	- Style	(S.1) – Brackets & Braces. (S.2) – Indentation. (S.3) – Blank Lines. (S.4) – Long Lines. (S.5) – Whitespace Usage. (S.6) – Grouping. (S.7) – Commented out code.
Production & Test Code	Maintainability & Perfective Maintenance	Structure (STR)	- Re-implementation	(STR.1) – Semantic Duplication. (STR.2) – Semantic Dead Code. (STR.3) – Change Function. (STR.4) – Standard Coding Conventions. (STR.5) – New Functionality. (STR.6) – Strings (Wording). (STR.7) – Logging. (STR.8) – Testing.

Кінець таблиці 1.2

1	2	3	4	5
Production & Test Code	Maintainability & Perfective Maintenance	Structure (STR)	- Organization	(STR.9) – Imports. (STR.10) – Move Functionality. (STR.11) – Long Sub Routine. (STR.12) – Dead Code. (STR.13) – Duplication / Redundant Code. (STR.14) – Complex Code/ Simplification. (STR.15) – Statement Issue. (STR.16) – Consistency. (STR.17) – Architectural Changes.
	Functionality / Corrective Maintenance	Interface (I)	- Interface	(I.1) – Function Call. (I.2) – Parameter.
		Logic (L)	- Logic	(L.1) – Compare. (L.2) – Computation. (L.3) – Wrong Location. (L.4) – Algorithm / Performance.
		Resource (R)	- Resource	(R.1) – Variable Initialization. (R.2) – Memory Management. (R.3) – Data & Resource Manipulation. (R.4) – Security. (R.5) – Concurrency.
		Check (C)	- Check	(C.1) – Check Function. (C.2) – Check Variable. (C.3) – Check User Input.
		Larger Defects (LD)	- Larger Defects	(LD.1) – Completeness. (LD.2) – GUI. (LD.3) – Check outside code / Domino Effects.
Other Changes	–	–	–	(O.1) – Commit Message. (O.2) – Continuous Integration / Continuous Delivery configurations. (O.3) – Automated Static Analysis Tools configurations. (O.4) – Language and Framework specific. (O.5) – External Software Documentation. (O.6) – Runtime Configurations. (O.7) – Other.

Результати порівняльного аналізу розглянутих у цьому підрозділі варіантів класифікацій RFC дозволяють зробити такі висновки [5]:

а) єдиної сталої системи класифікації RFC, яка охоплювала б увесь життєвий цикл ІТ-проєкту, не існує;

б) кожен різновид ІТ-проєкту може мати власну множину типів RFC, яка, в свою чергу, може розглядатися як підмножина загальних моделей класифікації таких RFC;

в) в будь-якому випадку загальний поділ RFC на окремі підмножини виділяє функціональні і технічні RFC;

г) кількість класів RFC може значною мірою залежати від таких характеристик ІТ-проекту:

- розмір ІТ-проекту;
- час, який відведено на виконання ІТ-проекту;
- необхідна швидкість реагування на потреби у змінах;
- рівень кваліфікації учасників ІТ-проекту, які мають право визначати потреби у змінах та безпосередньо формувати RFC;
- кількість очікуваних або вже наявних RFC, які вимагають класифікації та ранжування;

д) переважну більшість розглянутих класифікацій RFC побудовано на основі результатів обробки і аналізу вже існуючих RFC, без врахування особливостей застосування методів збирання цих запитів

1.3. Аналіз сучасних досліджень в галузі оцінювання запитів на зміни

Для проведення аналізу сучасних досліджень в галузі дослідження методів оцінювання змін було здійснено пошук наукових публікацій у наукометричній базі даних Scopus. Пошук проводився за ключовими словами «request», «change», «estimate», «project», «ІТ», «information», «system». Під час пошуку було використано такі фільтри:

- ключові слова повинні бути присутніми в назві публікації, переліку ключових слів або в тексті анотації цієї публікації;
- під час пошуку відбиралися тільки публікації у виданнях, які мають безпосереднє відношення до комп'ютерних наук (Computer Sciences).

В результаті пошуку було відібрано 54 наукові публікації, проіндексовані в наукометричній базі Scopus. Але слід зазначити, що серед цих публікацій були такі, що не повністю відповідали умовам пошуку. Тому після вивчення

матеріалів цих публікацій для подальшого аналізу було відібрано 10 робіт за період 2010-2023 рр.

У [14] вказано, що оцінювання змін розглядалося у 2010 р. як частина процесу спільного вирішення проблем. Інформацію про цей процес задокументовано в журналі змін контролю версій та в системах відстеження проблем, які фіксують вирішення проблем, запити на функції або інші завдання розробки. Виділення робіт із визначення та оцінювання змін у окремий процес в цей період не відбувалося. Вирішення питань оцінювання окремих змін у поточній версії ІТ-продукту виконувалося в ті часи як окрема задача групи процесів управління проблемами, яку, в свою чергу було визначено чинною на той час версією стандарту ISO 20000.

Період часу з 2010 р. до 2018 р. можна охарактеризувати як період, протягом якого для оцінювання змін дослідники намагалися застосовувати доволі прості засоби формального опису як самої зміни, так і створюваного ІТ-продукту (зокрема, інформаційних систем (ІС)). Як такі засоби формального опису було запропоновано:

а) структуровані текстові документи, які описують RFC, вихідний програмний код та додаткову контекстну інформацію (трасування виконання та початкову сутність вихідного коду, перевірену на наявність змін) [15];

б) профілі прототипів програмного забезпечення, які використовувалися для виявлення і документування функціональних вимог до створюваного ІТ-продукту [16];

в) спеціальні структурні моделі [17].

Для вирішення задачі оцінювання змін в цих дослідженнях пропонувалися, відповідно, такі методи:

а) комбінація методів пошуку інформації, динамічного аналізу та інтелектуального аналізу даних попередніх комітів вихідного коду [15];

б) метод оцінки за аналогією шляхом аналізу історичних даних з попередніх проєктів [16];

в) методи аналізу запитів на зміну історичних даних, пов'язаних із обслуговуванням і розвитком програмного забезпечення [17].

Своєрідним підсумком наукових розробок цього періоду часу є результати, опубліковані у [18], [19]. Ці результати було отримано саме для управління змінами в межах усього життєвого циклу сучасних ІС управління бізнес-процесами підприємства. Аналізуючи досвід попередніх дослідників та власні доробки у даному напрямі, автори [18], [19] формулюють такі передумови вирішення задачі оцінювання змін та їхнього впливу на ІС:

- основним артефактом, який відображає проєктні рішення та вплив проєктних рішень на технічні артефакти та організаційні обов'язки, є архітектура програмного забезпечення ІС;

- для прийняття рішень щодо затвердження змін важливо передбачити вплив RFC (наприклад, зміну інтерфейсу) на архітектуру програмного забезпечення;

- зусилля для реалізації RFC ІС може оцінити архітектор програмного забезпечення ІС, вивчаючи відповідні завдання реалізації.

Як головні недоліки існуючих підходів до оцінювання змін дослідники у [18] визначили те, що ці підходи або обмежуються артефактами процесу розробки програмного забезпечення, або не використовують офіційні описи архітектури.

Для подолання цих недоліків дослідники у [18] представили Karlsruhe Architectural Maintainability Prediction (КАМР) – інструментальний засіб дослідження змін та оцінювання їхнього впливу на архітектуру програмного забезпечення ІС. У [19] дослідники на основі КАМР представили інструментальний підхід для оцінки поширення змін, спричинених запитом на зміну в бізнес-процесах або програмних системах, на основі архітектури програмного забезпечення ІС та дизайну процесу. Приклад застосування цього підходу для ІС, архітектура якої базується на гібридному клоуд-базованому

Common Component Modeling (CoCoMe) Example, та бізнес-процесі, що реалізує сценарій продажу, наведено на рис. 1.8.

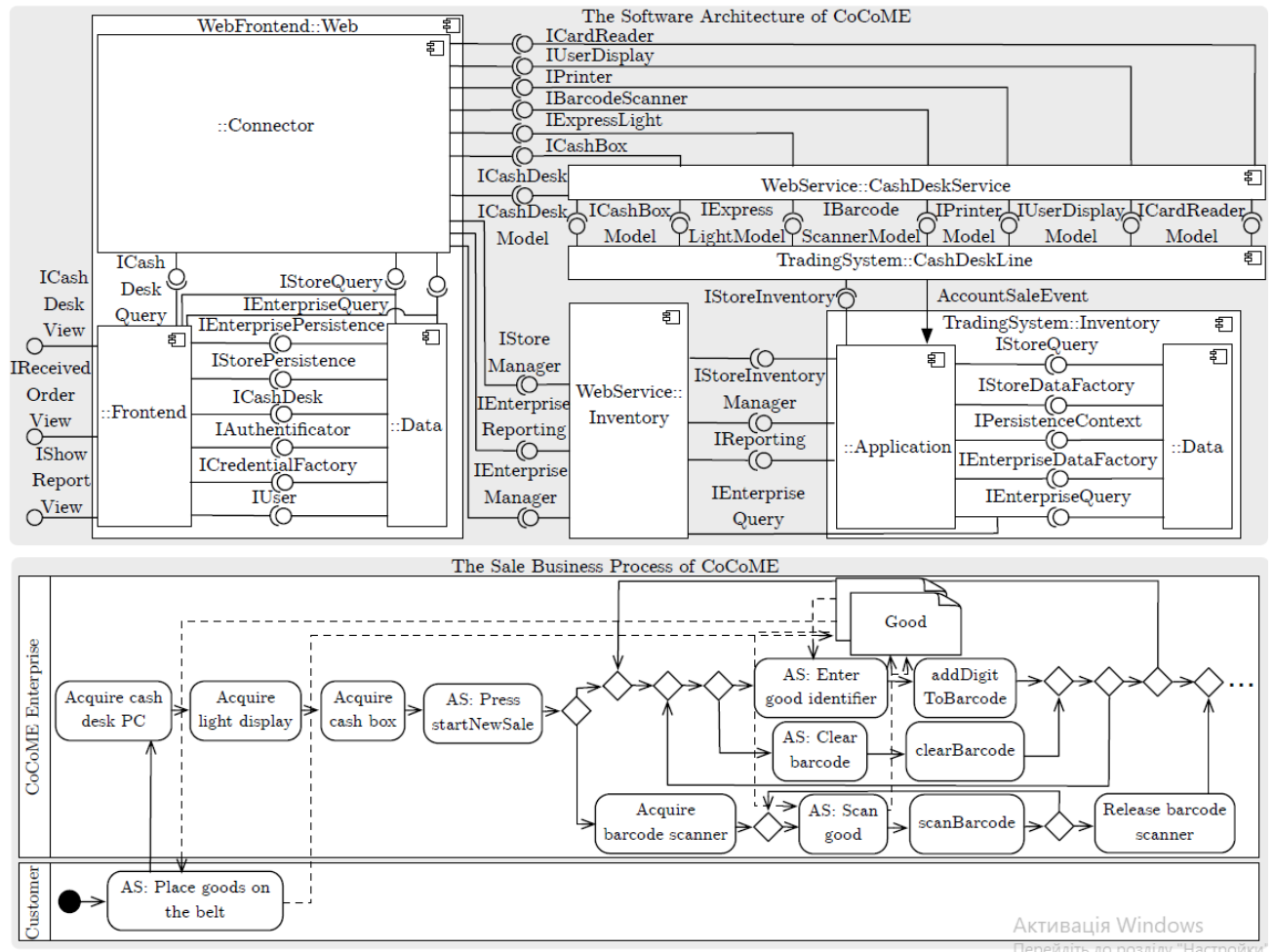


Рисунок 1.8 – Приклад застосування інструментального засобу КАМР для оцінювання змін на основі формальних візуальних моделей інформаційної системи та бізнес-процесу [19]

Але дослідження [15]-[19], які проводилися у період 2010-2018 р., базувалися на одному й тому ж припущенні про можливість повного формального опису RFC. Для дотримання цього припущення дослідники вимушені були виконувати значний обсяг робіт з перетворення неформальних публікацій RFC у вигляді текстів, аудіо- або відеофайлів у формальні описи цих RFC у вигляді структурних або об'єктно-орієнтованих візуальних або теоретико-

множинних моделей. Застосування цього припущення давало добрі результати у лабораторних дослідженнях, але викликало значні витрати на управління змінами у комерційних ІТ-проектах.

Досвід управління ІТ-проектами [2], [3] свідчить про те, що витрати на реалізацію та управління змінами можуть бути мінімізовані тільки у випадку, коли ці зміни вносяться до проекту на ранніх стадіях його життєвого циклу. Тому у 2018 р. було проведено дослідження впливу RFC через дефекти вимог на результати проектів розробки програмного забезпечення за методологією Waterfall [20]. У цій роботі розглядалися три типи дефектів вимог: неправильні вимоги, неповні вимоги та нові вимоги. Результати дослідження вимірювалися з точки зору загальних витрачених зусиль і дефектів програмного забезпечення, введених під час розробки програмного забезпечення.

Результати дослідження дозволили зробити такі висновки [20]:

- введення RFC у ІТ-проект шляхом публікації відповідних нових вимог приводить до збільшення кількості дефектів, а також до зростання обсягу зусиль (efforts, сучасний аналог застарілого терміну «трудовитрати» [2]);
- введення RFC у ІТ-проект шляхом коригування неправильних вимог збільшує кількість нових вимог, а також кількість введених дефектів;
- введення RFC у ІТ-проект шляхом уточнення неповних вимог не має вимірного впливу на результати проекту програмного забезпечення.

Під неповною вимогою тут слід розуміти вимогу до ІТ-продукту, модель якої містить актуальну та достовірну інформацію, але не охоплює усі необхідні сценарії виконання цієї вимоги. Це означає, що завжди може знайтися інший достовірний сценарій виконання вимоги, який з різних причин раніше не розглядався учасниками ІТ-проекту.

Приблизно з 2018 р. починається новий період дослідження, головну увагу в якому приділяють використанню для вирішення задачі оцінювання змін методів штучного інтелекту. Так, в [21] було запропоновано метод прогнозування типу технічного обслуговування, оцінювання наслідків змін і

часу технічного обслуговування експлуатованого програмного забезпечення. Джерелом інформації, необхідної для застосування цього методу, пропонувалося обрати існуючі репозиторії програмного забезпечення, що експлуатується. Для прогнозування завдань технічного обслуговування та їхнього впливу на зміни у [21] було запропоновано використати такі методи: лінійний опорний векторний класифікатор, Random Forest, логістичну регресію, мультиноміальний наївний метод Байєса, LSTM і Bi-LSTM. Для оцінки часу, необхідного для вирішення змін, у [21] було запропоновано використати штучну нейронну мережу (ANN). Результати експериментів показали, що Random Forest і LSTM мають точність 94 % і 95 % відповідно і показують кращі результати порівняно з іншими моделями. ANN надає середню квадратичну помилку 0,0028 [20].

У дослідженні [22] для оцінювання серйозності RFC шляхом вивчення його детального опису було запропоновано використовувати методології інтелектуального аналізу тексту та машинного навчання (ML). Під терміном «серйозність» тут розуміється терміновість (пріоритетність) реалізації RFC у IT-проекті. Під час дослідження було встановлено, що найкращим класифікатором RFC за рівнями їхньої серйозності є метод Random Forest. Було встановлено, що використання цього методу дозволяє передбачити можливу зміну серйозності RFC з точністю 94 % і визначити напрямок цієї зміни (збільшення або зменшення) з точністю 93 % [22].

Але застосування для вирішення задачі оцінювання змін у IT-проектах засобів штучного інтелекту не дозволяє відмовитись від припущення про необхідність опису RFC у вигляді формальних моделей. Невирішеними залишаються і усі проблеми, пов'язані зі створенням та застосуванням таких формальних описів RFC. Між тим, використання неструктурованих публікацій RFC, що є розповсюдженим у переважній кількості сучасних IT-проектів, призводить до суттєвого погіршення результатів застосування методів штучного інтелекту. Так, коли у [22] було поставлено завдання передбачити остаточний рівень серйозності RFC в ситуаціях із спотвореними даними, точність моделі

становила 68 %, що, за словами авторів публікації, відповідає наявній літературі. Ці результати показують, що використання методів штучного інтелекту і, зокрема, методів ML для вирішення задачі оцінювання змін недоцільне і треба шукати інший метод оцінки.

Як приклад такого іншого методу слід розглянути запропонований у [23] метод, який дозволяє отримати вигоду від відстеження вимог і прогнозування зусиль при змінах обсягу IT-проєкту. Результати застосування цього методу до IT-проєкту розробки програмного забезпечення синтезуються в індикатори. Ці індикатори демонструють, як зміни обсягу, реалізовані через різні ітерації розробки, впливають на інші вимоги до програмного забезпечення та як складність вимоги, що змінюється, може бути використано для визначення оцінки зусиль. Але цей метод не використовує складних розрахунків і значною мірою базується на використанні інтуїції та досвіду виконавців IT-проєкту.

Проведений аналіз сучасних досліджень методів оцінювання змін у IT-проєктах дозволяє зробити такі висновки:

- вирішенню задачі оцінювання змін та їхнього впливу на IT-проєкт останнім часом приділяється недостатньо уваги;
- застосування для оцінювання змін та їхнього впливу на IT-проєкт будь-яких формальних описів RFC може дати добрі результати в лабораторних умовах, але їх використання у комерційних IT-проєктах обмежено великими розподіленими ІС із хмарною архітектурою;
- застосування методів штучного інтелекту для оцінювання змін та їхнього впливу на IT-проєкт в умовах використання неструктурованих публікацій RFC не дає бажаного наслідку;
- перевага під час оцінювання змін та їхнього впливу на IT-проєкт віддається методам, які базуються на використанні інтуїції та досвіду команд виконавців цих проєктів.

1.4 Мета і задачі магістерської класифікаційної роботи

Проведений аналіз особливостей оцінювання змін у ІТ-проєктах дозволяє зробити такі висновки.

По-перше, актуальний фреймворк управління змінами розглядає оцінювання змін та їхнього впливу на ІТ-продукт як результат роботи «Окресліть масштаби змін». Ця робота може виконуватися у будь-який момент ІТ-проєкту незалежно від того, чи використовуються для управління цим проєктом традиційні або Agile-методології. Жодних рекомендацій з визначення особливостей роботи «Окресліть масштаби змін» на подальших стадіях життєвого циклу ІТ-проєкту актуальний фреймворк управління змінами не надає.

По-друге, єдиної сталої системи класифікації RFC, яка охоплювала б увесь життєвий цикл ІТ-проєкту, не існує. Практично для усіх ІТ-проєктів множина типів RFC обов'язково поділяється на дві множини, що не перетинаються між собою, – функціональні і технічні RFC. Але кожен різновид ІТ-проєкту може мати власне наповнення цих підмножин окремими типами RFC, кількість яких визначається як загальними характеристиками проєкту, так і особливостями самих RFC та осіб, які працюють з ними. При цьому переважна більшість класифікацій RFC побудовано на основі результатів обробки і аналізу вже існуючих RFC. Методи збирання потреб, на основі яких формуються ці запити, на думку дослідників, не впливають на результати вирішення задачі класифікації RFC.

По-третє, сучасними дослідниками вирішенню задачі оцінювання змін та їхнього впливу на ІТ-проєкт приділяється недостатньо уваги. Встановлено, що застосування будь-яких формальних описів RFC для оцінювання змін та їхнього впливу на ІТ-проєкт може дати добрі результати переважно в лабораторних умовах. Використання аналогічних формальних описів RFC у прикладних умовах обмежено ІТ-проєктами, які пов'язані зі створенням та розвитком великих розподілених ІС із хмарною архітектурою.

По-четверте, застосування методів штучного інтелекту для оцінювання змін та їхнього впливу на ІТ-проект в умовах використання неструктурованих публікацій RFC не дає бажаного наслідку. Перевага під час оцінювання змін та їхнього впливу на ІТ-проект віддається методам, які базуються на використанні інтуїції та досвіду команд виконавців цих проектів.

Виходячи з цих висновків, роботу «Окресліть масштаби змін» слід визнати як слабо досліджену роботу, результати виконання якої мають сильний вплив на рішення щодо доцільності впровадження оціненого RFC у ІТ-проект. Тому проведення досліджень, які дадуть змогу виробити рекомендації з виконання цієї роботи у конкретних ІТ-проектах та узагальнити ці рекомендації для окремих різновидів ІТ-проектів, слід вважати теоретично та практично актуальним.

Об'єктом дослідження в рамках магістерської кваліфікаційної роботи є робота «Окресліть масштаби змін» підпроцесу «Формулювання змін» процесу «Управління змінами» [2].

Предметом дослідження є методи інтуїтивного оцінювання змін функціональних вимог у ІТ-проекті.

Метою роботи є розробка ІТ оцінювання змін функціональних вимог до системи у ІТ-проекті створення ІС, яка дозволила б підвищити точність оцінювання обсягу змін функціональних вимог, що пропонуються під час виконання ІТ-проекту.

Для досягнення поставленої мети у роботі пропонується вирішити такі задачі:

- а) дослідити та проаналізувати особливості оцінювання змін у ІТ-проекті;
- б) дослідити особливості існуючих інтуїтивних методів оцінювання обсягу ІТ-проекту та вдосконалити метод, який найкраще відповідає особливостям оцінювання зусиль, необхідних для реалізації RFC функціональної вимоги до ІС;
- в) розробити ІТ оцінювання змін функціональних вимог до ІС на основі вдосконаленого методу;

г) провести практичну апробацію отриманих результатів на прикладі ІТ-проєкту розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри».

2 ВДОСКОНАЛЕННЯ ІНТУЇТИВНОГО МЕТОДУ ОЦІНЮВАННЯ ЗМІН ФУНКЦІОНАЛЬНИХ ВИМОГ У ІТ-ПРОЄКТІ

2.1 Порівняльний аналіз інтуїтивних методів оцінювання зусиль на реалізацію функціональних вимог до інформаційної системи

2.1.1 Дослідження особливостей інтуїтивного оцінювання функціональних вимог до інформаційної системи

Під час оцінювання вимог, які висуваються до ІС, менеджеру проекту треба надати керівництву відомості про передбачувану тривалість і вартість виконання робіт з реалізації цих вимог. Найбільші труднощі під час цього оцінювання викликає високий ступінь невизначеності, породжений малою кількістю інформації в публікації функціональної вимоги. Тому переважна більшість методологій управління ІТ-проектом покладають вирішення задач оцінювання зусиль (трудовитрат), потрібних на реалізацію відповідних функціональних вимог до ІС, на експертів ІТ-компанії. В ролі таких експертів найчастіше виступають системні архітектори та безпосередні виконавці ІТ-проекту зі створення, модернізації або розвитку ІС.

Як основні вимоги, які висуваються до потенціальних експертів з оцінювання зусиль на реалізацію функціональних вимог до ІС, [2] пропонує:

- компетентність в досліджуваній галузі;
- ерудованість в суміжних галузях;
- вчений ступінь;
- наукове звання;
- стаж наукової або практичної роботи в даній сфері;
- посада;
- принциповість;
- об'єктивність;
- спроможність творчо мислити;

– інтуїція.

Оскільки застосування інтуїтивних методів оцінювання орієнтовано на застосування Agile-методологій управління ІТ-проектами, вирішення задачі оцінювання з використанням цих методів повинно відповідати таким принципам [23]:

- а) принцип 1: «Велика швидкість оцінювання»;
- б) принцип 2: «Командна праця»;
- в) принцип 3: «Відносні одиниці виміру».

Найпопулярнішими в ІТ-спільноті методами інтуїтивного оцінювання є такі методи [24]:

- метод T-Shirt Sizes;
- метод Planning Poker;
- метод Bucket System;
- метод Dot-voting;
- метод Maximum Size or Less;
- метод Big/Small/Uncertain;
- метод Ordering Rule.

У наступних пунктах розглянемо особливості кожного з цих методів.

2.1.2 Метод T-Shirt Sizes

В цьому методі як одиниці вимірювання використовуються розміри футболок: XS, S, M, L, XL.

Команда приймає рішення про розмір тієї чи іншої користувальницької історії в ході спільної відкритої дискусії. У разі невизначеності можливе застосування голосування. При бажанні можна домовитися про співвідношення «розмірів», наприклад, S дорівнює 2xXS, M дорівнює 2xS тощо.

Зазвичай, попередньо оцінюються перші кілька завдань. Далі відбувається декомпозиція цих завдань. Коли команда в результаті декомпозиції встановлює найдрібніші завдання, вони приймаються за XS. Після цього інші завдання оцінюються з точки зору того, наскільки вони більше за XS. Залежно від результатів оцінювання, цим завданням присвоюється ступінь декомпозиції історій, якому відповідає розмір S, M, L або XL. У підсумку для кожного завдання команда знаходить відповідний розмір [25].

Можна також домовитися, що команда під час оцінювання буде використовувати великий розмір XXL. Присвоєння завданню цього розміру говорить, що насправді команда не може оцінити це завдання і воно потребує подальшої декомпозиції або уточнення [24], [25].

Даний метод є досить швидким і його можна використовувати для оцінки великої кількості завдань за одну сесію. За його допомогою цілком реально за годину оцінити 15-20 завдань [24], [25].

2.1.3 Метод Planning Poker

Цей метод є одним з найпопулярніших. Особливістю цього методу є використання учасниками спеціально пронумерованих карт (див. рис. 2.1), щоб голосувати з їхньою допомогою за оцінку завдання. Зазвичай для «покеру» використовуються карти з числами Фібоначчі (0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89), але можливі й інші варіанти [26].

Метод Planning Poker складається з таких етапів [24], [26]:

Етап 1. Кожен учасник отримує колоду карт з числовими значеннями для оцінки, зображенням «?» (запит уточнення завдання) і «чашки кави» (вимога перерви).



Рисунок 2.1 – Приклад карт, що застосовуються в методі Planning Poker [26]

Етап 2. Product owner робить короткий анонс чергового завдання і відповідає на питання команди щодо цього завдання.

Етап 3. Учасники «покеру» вибирають карту з відповідною, на їхню думку, оцінкою і кладуть їх сорочкою вгору (щоб не впливати на вибір один одного).

Етап 4. Після того, як всі члени команди вибрали свої оцінки, карти одночасно перевертаються.

Етап 5. Учасники з найнижчими і найвищими оцінками роблять короткі коментарі, пояснюючи свій вибір оцінки завдання.

Етап 6. В результаті процесу обговорення команда приходить до єдиного рішення і після цього переходить до наступного завдання (повертається до етапу 2). Якщо усі завдання розглянуто, використання методу припиняється.

Метод Planning Poker є одним з найточніших методів інтуїтивного оцінювання, але підходить для порівняно невеликої кількості завдань. Протягом годинної сесії таким способом можна оцінити від 4 до 10 завдань [24], [26].

2.1.4 Метод Bucket System

У цьому методі завдання оцінюються і поміщаються у «відерця» відповідного розміру. Для зазначення розміру також можна використовувати числа Фібоначчі. Однак, на відміну від методу Planning Poker, в методі Bucket System після початкового масштабування задач завдання розділяються між учасниками для оцінки [27].

Метод Bucket System складається з послідовного виконання таких етапів

Етап 1. Всі завдання, які потрібно оцінити, виписуються на картки.

Етап 2. На столі або дошці вибудовується послідовність з «відерць» для завдань різного розміру.

Етап 3. Команда вибирає по черзі 3-5 довільних карток з завданнями і оцінює їх в ході відкритого обговорення, порівнюючи і вибудовуючи їх одну відносно іншої.

Етап 4. Завдання розміщуються у відповідні «відра», задаючи загальний масштаб і орієнтири для подальших оцінок.

Етап 5. Решта завдань порівну розділяються між усіма учасниками і оцінюються ними самостійно, з урахуванням отриманої шкали вимірювань. Якщо хтось із членів команди не може оцінити будь-яку історію, то він передає її іншому.

Етап 6. Якщо залишилися нерозподілені завдання, слід повернутися до виконання етапу 5. В іншому випадку слід завершити використання методу.

Даний метод, на відміну від методу Planning Poker, може використовуватися для швидкої оцінки дуже великого числа завдань (від 50) і з великою кількістю учасників [24], [27].

2.1.5 Метод Dot-voting

Цей метод передбачає використання спеціальних «точок», які показують голоси учасників або бали оцінки, призначені тому чи іншому завданню. Як такі «точки» можуть використовуватися наклейки, стікери, магніти, точки або штрихи, проставлені маркерами.

Метод Dot-voting складається з таких етапів [24], [27].

Етап 1. Всі оцінювані завдання виписуються на окремі картки і розміщуються на столі або дошці.

Етап 2. Для виконання оцінки кожен з учасників отримує однакову кількість «точок». Кожен член команди розподіляє свої «точки» між завданнями як він вважає за потрібне, враховуючи, що чим більше «точок», тим складніше завдання і тим більше необхідно часу на виконання цього завдання.

Етап 3. Після того, як кожен учасник зробив свою оцінку і розподілив всі свої «точки», підраховується загальна кількість точок, виставлених для кожного завдання.

Етап 4. Проводиться ранжування всіх оцінюваних завдань за кількістю «точок», виділених кожному завданню.

Даний метод є дуже простим і швидким, але він буде ефективно працювати під час оцінювання лише невеликої кількості завдань (до 8 або до 10) [24], [27].

2.1.6 Метод Maximum Size or Less

Цей метод спрямовано на допомогу учасникам процесу оцінювання спочатку визначити максимально можливий розмір завдання у беклогу завдань. Найчастіше як максимальне значення обирається 1 людино-день. В цьому

випадку найбільші завдання повинні вимагати для їх виконання 1 людиною не більше 1 дня.

Метод Maximum Size or Less можна представити як послідовність таких етапів [24], [27].

Етап 1. Кожне завдання обговорюється всіма учасниками, щоб відповісти на питання: «Завдання, яке оцінюється, більше максимального значення чи менше?».

Етап 2. Якщо оцінюване завдання більше максимального розміру, то група учасників декомпозує це завдання на підзавдання і повертається до етапу 1.

Етап 3. Етапи 1 та 2 виконуються доти, доки всі оцінювані завдання не опиняться в дозволеному діапазоні розмірів (дорівнюватимуть або будуть менше обраного за максимальне значення). Після цього використання методу завершується.

Цей метод оцінки також дуже простий у використанні, і з його допомогою команда здатна впоратися з 15-30 завданнями (в залежності від складності та досвіду декомпозиції) [24], [27].

2.1.7 Метод Big/Small/Uncertain

Метод Big/Small/Uncertain (великий/малий/невизначений) в цілому схожий на метод Bucket System. Основна відмінність полягає в тому, що в методі Uncertain використовується тільки 3 категорії оцінювання: «Великий розмір завдання», «Малий розмір завдання», та «Невизначений розмір завдання».

Метод Big/Small/Uncertain можна представити як послідовність таких етапів [24], [27].

Етап 1. Всі оцінювані завдання обговорюються учасниками і поміщаються в одну з трьох категорій «Big», «Small» або «Uncertain».

Етап 2. Група учасників проводить сумісне обговорення кількох перших завдань (від 3 до 5), визначаючи масштаб і орієнтири для кожної категорії.

Етап 3. Решта завдань розподіляються між учасниками і оцінюються самостійно. Після оцінювання усіх завдань – завершення використання методу. Метод Big/Small/Uncertain є одним з найшвидших методів інтуїтивного оцінювання. Він дозволяє оцінити за одну сесію велику кількість завдань (від 50) і дозволяє залучати до процесу одночасно багато учасників [24], [27].

2.1.8 Метод Ordering Rule

Даний метод є покроковою грою, мета якої – впорядкувати всі завдання одне відносно одного за єдиною шкалою розміру.

Метод Ordering Rule можна представити як послідовність таких етапів [24], [27].

Етап 1. Всі оцінювані завдання виписуються на картки.

Етап 2. Картки з завданнями випадковим чином розміщуються на столі або дошці зі шкалою, на межах якої вказані позначки «Малий розмір» і «Великий розмір».

Етап 3. Всі учасники по черзі роблять свій хід оцінювання. Такий хід включає одну з таких можливих дій:

- перемістити будь-яке завдання за шкалою на одну поділку (тобто поміняти оцінку на нижчу або вищу);
- обговорити завдання з іншими учасниками;
- пропустити свій хід.

В результаті ходів учасників завдання можуть переміщатися дошкою в межах встановленої шкали, їхня оцінка одне відносно одного уточнюється.

Етап 4. Завершення використання методу за умови, якщо всі учасники пропускають свій хід.

В результаті застосування методу Ordering Rule всі завдання розподілені за шкалою між значеннями «малий розмір» і «великий розмір». Цей метод досить ефективний для оцінки невеликої кількості завдань (від 5 до 15). Учасники залучені до загального гейміфікованого процесу оцінювання і, змінюючи положення завдань одне відносно одного, домагаються високої точності оцінки [24], [27].

2.1.9 Результати порівняльного аналізу розглянутих методів інтуїтивного оцінювання за обраними критеріями

Для визначення можливості застосування методів інтуїтивного оцінювання для виконання роботи «Окресліть масштаби змін» виявимо особливості виконання цієї роботи, які можуть вплинути на прийняття рішення про доцільність застосування конкретного методу у цій роботі.

Перш за все, робота «Окресліть масштаби змін» характеризується достатньо великою кількістю змін, що можуть виникати в будь-який момент життєвого циклу ІТ-проєкту. Це підтверджується відомостями, наданими у [7] та [11]. Так, у [7] апробація проводилася на 40 RFC.

Другою важливою особливістю роботи «Окресліть масштаби змін» є необхідність якомога швидкого її виконання. Ця особливість витікає із необхідності застосування принципу 1 (див. п. 2.1.1) Agile-методологій управління ІТ-проєктами.

Третьою важливою особливістю роботи «Окресліть масштаби змін» є участь у процесі оцінювання великої кількості фахівців. Як стверджується у [14], оцінювання змін відбувається найкраще тоді, коли в цьому процесі задіяні усі

системні архітектори та провідні розробники усіх видів забезпечень створюваного ІТ-продукту.

Четвертою важливою особливістю роботи «Окресліть масштаби змін» є необхідність отримати достатньо точні оцінки зусиль та часу, необхідних для реалізації встановлених RFC. Особливо це стосується такого різновиду RFC, як RFC функціональних вимог до ІС, тому що саме ці RFC дуже сильно впливають на зміст ІТ-проекту та загальний час його виконання.

Виходячи з цих особливостей, було запропоновано охарактеризувати роботу «Окресліть масштаби змін» за такими критеріями [28]:

- а) критерій «Можливість оцінювання великої кількості завдань» K_1 ;
- б) критерій «Велика швидкість оцінювання» K_2 ;
- в) критерій «Можливість участі великої кількості зацікавлених осіб» K_3 ;
- г) критерій «Велика точність оцінювання» K_4 .

Кожен з цих критеріїв може приймати два значення: «Так» (або «1») або «Ні» (або «0»).

Значення цих критеріїв для роботи «Окресліть масштаби змін» наведено у табл. 2.1 [28].

Таблиця 2.1 – Значення критеріїв K_1 , K_2 , K_3 та K_4 для роботи «Окресліть масштаби змін»

Назва роботи	K_1	K_2	K_3	K_4
«Окресліть масштаби змін»	1	1	1	1

Значення цих критеріїв для кожного з методів інтуїтивного оцінювання, розглянутих у пп. 2.1.2 – 2.1.8, наведено у табл. 2.2 [28].

Таблиця 2.2 – Значення критеріїв K_1 , K_2 , K_3 та K_4 для розглянутих методів інтуїтивного оцінювання

№ методу	Назва методу	K_1	K_2	K_3	K_4
1	Метод T-Shirt Sizes	1	1	0	0
2	Метод Planning Poker	0	0	0	1
3	Метод Bucket System	1	1	1	0
4	Метод Dot-voting	0	1	0	1
5	Метод Maximum Size or Less	1	1	0	0
6	Метод Big/Small/Uncertain	1	1	1	0
7	Метод Ordering Rule	0	0	0	1

Для ранжування розглянутих методів інтуїтивного оцінювання за встановленими критеріями запропоновано використовувати відстань Хеммінга. Ця відстань між векторами оцінок роботи «Окресліть масштаби змін» (табл. 2.1) та векторами оцінок кожного окремого методу інтуїтивного оцінювання за критеріями K_1 , K_2 , K_3 та K_4 (табл. 2.2) розраховується за формулою [28]

$$d_H(A, M_i) = \sum_{t=1}^4 |K_{At} - K_{it}|, \quad (2.1)$$

де A – робота «Окресліть масштаби змін»;

M_i – i -й метод інтуїтивного оцінювання, $i=1, \dots, 7$;

t – номер критерію оцінювання, $t=1, \dots, 4$;

K_{At} – значення t -го критерію, отримане під час оцінювання роботи A (див. табл. 2.1);

K_{it} – значення t -го критерію, отримане під час оцінювання i -го методу (див. табл. 2.2).

Слід зазначити, якщо оцінки за критеріями K_1 , K_2 , K_3 та K_4 методу інтуїтивного оцінювання будуть збігатися з оцінками роботи «Окресліть масштаби змін», відстань Хеммінга буде дорівнювати 0.

Результати ранжування методів інтуїтивного оцінювання за збільшенням значення відстані Хеммінга наведено у табл. 2.3 [28].

Таблиця 2.3 – Результати ранжування методів інтуїтивного оцінювання за значенням відстані Хеммінга

№ методу	Назва методу	K_1	K_2	K_3	K_4	$d_H(A, M_i)$
3	Метод Bucket System	1	1	1	0	1
6	Метод Big/Small/Uncertain	1	1	1	0	1
1	Метод T-Shirt Sizes	1	1	0	0	2
4	Метод Dot-voting	0	1	0	1	2
5	Метод Maximum Size or Less	1	1	0	0	2
2	Метод Planning Poker	0	0	0	1	3
7	Метод Ordering Rule	0	0	0	1	3

За наведеними у табл. 2.3 результатами слід зробити такі висновки [28]:

а) не існує методу інтуїтивного оцінювання, який був би одночасно достатньо розповсюдженим та повністю відповідав би особливостям виконання роботи «Окресліть масштаби змін»;

б) з існуючих найрозповсюдженіших методів інтуїтивного оцінювання найбільш відповідають особливостям роботи «Окресліть масштаби змін» методи Bucket System та Big/Small/Uncertain;

в) головною проблемою застосування методів Bucket System та Big/Small/Uncertain для оцінювання RFC функціональних вимог до ІС слід вважати порівняно низьку точність оцінок, які розраховуються за цими методами.

Для вдосконалення методу Bucket System або Big/Small/Uncertain з метою підвищення точності оцінювання RFC функціональних вимог до ІС запропоновано за результатами порівняльного аналізу, наведеними у табл. 2.3,

використовувати один з методів, який забезпечую високу точність оцінювання, а саме [28]:

- метод Dot-voting;
- метод Planning Poker;
- метод Ordering Rule.

2.2 Вдосконалення методу Big/Small/Uncertain для оцінювання змін функціональних вимог до інформаційної системи

За результатами порівняльного аналізу методів інтуїтивного оцінювання у підрозд. 2.1 було рекомендовано обрати для вдосконалення методи Bucket System та Big/Small/Uncertain як такі, що найбільш відповідають особливостям роботи «Окресліть масштаби змін». З цих двох методів метод Big/Small/Uncertain є простішим у використанні, тому що він не вимагає створення множини багатьох категорій («відерець») для оцінювання.

Для вдосконалення методу Big/Small/Uncertain пропонується використати концепцію ансамблювання [29]. За цією концепцією подолання недоліків одного способу вирішення проблеми здійснюється за рахунок використання іншого способу вирішення цієї ж проблеми, в якому цей недолік або відсутній, або не має суттєвого значення. Таким недоліком для методу Big/Small/Uncertain за результатами порівняльного аналізу (див. табл. 2.2) є порівняно низька точність оцінювання.

Для подолання встановленого недоліку методу Big/Small/Uncertain у п. 2.1.9 було запропоновано використати один з трьох методів інтуїтивного оцінювання: метод Dot-voting, метод Planning Poker або метод Ordering Rule. Кожен з цих методів дає найточніші з можливих інтуїтивних оцінок змін функціональних вимог до ІС. Кожен з цих методів також відповідає одному з

існуючих напрямів ансамблювання, а саме так званій «Суміші експертів». Цей напрям ансамблювання застосовує стратегію «розділяй та володарюй» – спробу спільного навчання суміші параметричних моделей і використання правил об'єднання, щоб отримати на основі цих моделей загальні рішення [29]. В даній роботі ролі параметричних моделей відіграють експерти, кожен з яких має власну думку відносно оцінки зусиль, потрібних для реалізації зміни функціональної вимоги до ІС. Але з цих трьох методів метод Ordering Rule є найпростішим у застосуванні, оскільки він не вимагає використання ніяких додаткових засобів (карти для Planning Poker; наклейки, стікери або магніти для позначення «точок» тощо).

Тому метод Big/Small/Uncertain пропонується вдосконалити саме із застосуванням методу Ordering Rule для підвищення підсумкової точності оцінювання змін функціональних вимог до ІС. Результатом цього вдосконалення є ансамблевий метод Size Ordering Rule, який складається з таких етапів [28].

Етап 1. Визначаються три категорії оцінювання, кожна з яких позначається словами «Big», «Small» та «Uncertain» відповідно.

Етап 2. Учасники сумісно визначають показник, який визначає межу між категоріями «Big» та «Small», та його значення.

Етап 3. Кожен оцінюваний RFC публікується у вигляді ідентифікованої картки з описом цього RFC і надається для ознайомлення усім учасникам методу.

Етап 4. Кожна з ідентифікованих карток випадковим чином розташовується в одній з категорій оцінювання «Big», «Small» та «Uncertain». Ця процедура відбувається доти, доки не залишається ідентифікованих карток, що не розподілені між цими категоріями.

Етап 5. Всі учасники по черзі роблять свій хід як одну з таких дій:

- перемістити будь-яку ідентифіковану картку в іншу категорію оцінювання;
- обговорити зміст ідентифікованої картки з іншими учасниками;
- пропустити свій хід.

Виконання етапу 5 завершується, коли усі учасники методу пропустять свій хід.

Етап 6. Якщо у категорії «Uncertain» залишилася непуста множина ідентифікованих карток, то картки, які складають цю множину, розподіляються випадковим чином між учасниками методу і розподіляються цими учасниками випадковим чином між категоріями «Big» та «Small». Ця процедура відбувається, доки у категорії «Uncertain» не залишиться жодної ідентифікованої картки.

Етап 7. Обирається одна з нерозглянутих категорій: «Big» або «Small». В межах цієї категорії встановлюється шкала, на кордонах якої вказані позначки «Малий розмір» і «Великий розмір». Для уточнення шкали використовується значення показника, обране на етапі 2.

Етап 8. Ідентифіковані картки відповідної категорії випадковим чином розміщуються в межах шкали, встановленої на етапі 7.

Етап 9. Всі учасники по черзі роблять свій хід як одну з таких дій:

- перемістити будь-яку ідентифіковану картку за шкалою на одну поділку (тобто поміняти оцінку на нижчу або вищу);
- обговорити обрану ідентифіковану картку з іншими учасниками;
- пропустити свій хід.

Виконання етапу 9 завершується, коли усі учасники методу пропустять свій хід.

Етап 10. Обирається інша з нерозглянутих з категорій і відбувається повернення до етапу 7. У випадку, якщо обидві категорії розглянуто, використання методу завершується.

Джерела вибору та розробки етапів ансамблевого методу Size Ordering Rule наведено у табл. 2.4.

Таблиця 2.4 – Джерела вибору та розробки етапів ансамблевого методу Size

Номер етапу	Джерело вибору/розробки етапу
Етап 1	Похідний метод Big/Small/Uncertain
Етап 2	Запропоновано автором
Етап 3	Похідний метод Ordering Rule
Етап 4	Похідний метод Big/Small/Uncertain
Етап 5	Похідний метод Ordering Rule
Етап 6	Запропоновано автором
Етап 7	Похідний метод Ordering Rule
Етап 8	Похідний метод Ordering Rule
Етап 9	Похідний метод Ordering Rule
Етап 10	Похідний метод Ordering Rule

Розроблений ансамблевий метод Size Ordering Rule дозволяє отримати точніші оцінки зусиль на реалізацію RFC функціональних вимог до ІС внаслідок застосування елементів похідного методу Ordering Rule для підвищення точності оцінювання кожного з RFC, який визначено групою учасників виконання методу в межах кожної з окремих категорій «Big» та «Small». Крім того, в розробленому ансамблевому методі Size Ordering Rule запропоновано відмовитися від виділення ідентифікованих карток з публікаціями RFC функціональних вимог до ІС у категорію «Uncertain», що зменшує рівень невизначеності під час проведення оцінювання.

2.3 Висновки до другого розділу

В цьому розділі було визначено основні особливості процесу інтуїтивного оцінювання RFC функціональних вимог до ІС. Наведено описи найрозповсюдженіших методів інтуїтивного оцінювання та встановлено основні особливості використання кожного з цих методів.

Для проведення порівняльного аналізу виділених найрозповсюдженіших методів інтуїтивного оцінювання було встановлено основні особливості роботи «Окресліть масштаби змін», які можуть вплинути на прийняття рішення про доцільність застосування конкретного методу у цій роботі. За цими особливостями встановлено чотири критерії порівняльного аналізу виділених методів та встановлено множини припустимих значень кожного з цих критеріїв. За обраними критеріями було оцінено роботу «Окресліть масштаби змін» в умовах інтуїтивного оцінювання цих масштабів (див. табл. 2.1) та кожен з виділених методів інтуїтивного оцінювання (див. табл. 2.2). Результати порівняльного аналізу відповідності виділених методів особливостям роботи «Окресліть масштаби змін» наведено у табл. 2.3. За результатами порівняння було рекомендовано обрати для подальшого вдосконалення методи Bucket System та Big/Small/Uncertain за умови, що вдасться поліпшити точність їхнього оцінювання. Для цього поліпшення було запропоновано використати методи Dot-voting, Planning Poker або Ordering Rule.

Було обґрунтовано рішення використати для вдосконалення похідні методи Big/Small/Uncertain та Ordering Rule. Як основну концепцію вдосконалення було запропоновано використати напрям «Суміш експертів» концепції ансамблювання. В результаті вдосконалення було розроблено ансамблевий метод Size Ordering Rule та встановлено джерела вдосконалення кожного з етапів цього методу.

Основні результати розділу опубліковано у роботі [28].

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОЦІНЮВАННЯ ЗМІН ФУНКЦІОНАЛЬНИХ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розробка моделі потоків даних інформаційної технології

3.1.1 Визначення вимог до інформаційної технології в цілому

ІТ оцінювання змін функціональних вимог до ІС є однією з ІТ, які складають ІС управління ІТ-проєктами, що застосовуються для автоматизованого управління ІТ-проєктами створення, модернізації або розвитку ІС управління підприємствами та організаціями. Тому запропоновано розглядати цю ІТ як таку, що вдосконалює та розвиває існуючі ІС або ІТ управління ІТ-проєктами на різних стадіях їхнього життєвого циклу.

Оскільки основними об'єктами змін в даній роботі є функціональні вимоги до ІС, було прийнято рішення розглядати ІТ оцінювання змін функціональних вимог до ІС як технологію, що розвиває ІТ в галузі інженерії вимог до системи. Як приклад такої технології в роботі запропоновано використовувати інтелектуальну ІТ прискореної розробки ІС, описану у [30]. Основними функціями (ІТ-послугами) інтелектуальної ІТ прискореної розробки ІС є [30]:

- а) виявлення вимог до ІС;
- б) формування ієрархій термінів предметної області (ПрО) створюваної ІС;
- в) порівняння сформованих ієрархій термінів ПрО ІС з розробленими раніше ієрархіями термінів ПрО;
- г) виділення і уточнення патернів вимог до ІС;
- д) синтез архітектури ІС на основі виділених патернів вимог до ІС;
- е) здійснення взаємно-однозначного відображення патернів ІС в сутності бази даних і класи програмного забезпечення розроблюваної ІС.

Оцінювання змін функціональних вимог до ІС, яке виконується в межах роботи «Окресліть масштаби змін», має особливість, що відрізняє цю роботу від типових робіт з оцінювання зусиль на реалізацію сформульованих

функціональних вимог до ІС. Ця особливість полягає у такому: під час оцінювання зусиль на реалізацію сформульованої функціональної вимоги до ІС враховуються особливості лише цієї функціональної вимоги, в той час як під час оцінювання зусиль на реалізацію зміни функціональної вимоги до ІС треба оцінити:

- а) зусилля на реалізацію RFC функціональної вимоги до ІС;
- б) зусилля на реалізацію змін у всіх інших сформульованих функціональних вимогах до ІС, які можуть бути зачепленими в процесі реалізації RFC функціональної вимоги до ІС.

Ця особливість разом із розглянутими у [30] особливостями інтелектуальної ІТ прискореної розробки ІС визначають сукупність вимог, які висуваються до ІТ оцінювання змін функціональних вимог до ІС. Цю сукупність вимог запропоновано розділити на три основні множини вимог:

- множина вимог до ІТ в цілому;
- множина функціональних вимог до ІТ;
- множина вимог до видів забезпечень ІТ.

Множина вимог до ІТ в цілому складається з таких вимог.

ІТ оцінювання змін функціональних вимог до ІС потрібно розглядати як окрему ІС, що складається з трьох функціональних модулів:

- функціональний модуль «Оцінювання RFC функціональних вимог до ІС, які визнано доцільними для подальшої реалізації» (далі – функціональний модуль 1);
- функціональний модуль «Оцінювання впливу на існуючий проєкт ІС тих RFC функціональних вимог до ІС, які визнано доцільними для подальшої реалізації» (далі – функціональний модуль 2);
- функціональний модуль «Формування та публікація звіту з результатами оцінювання RFC функціональних вимог до ІС» (далі – функціональний модуль 3).

Взаємозв'язок ІТ оцінювання змін функціональних вимог до ІС з інтелектуальною ІТ прискореної розробки ІС потрібно здійснювати на рівні БД вдосконалюваної ІТ шляхом виконання запитів до таблиць БД.

Контекстну діаграму моделі потоків даних, яка описує основні взаємодії ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, з користувачами та інтелектуальною ІТ прискореної розробки ІС, наведено на рис. 3.1.

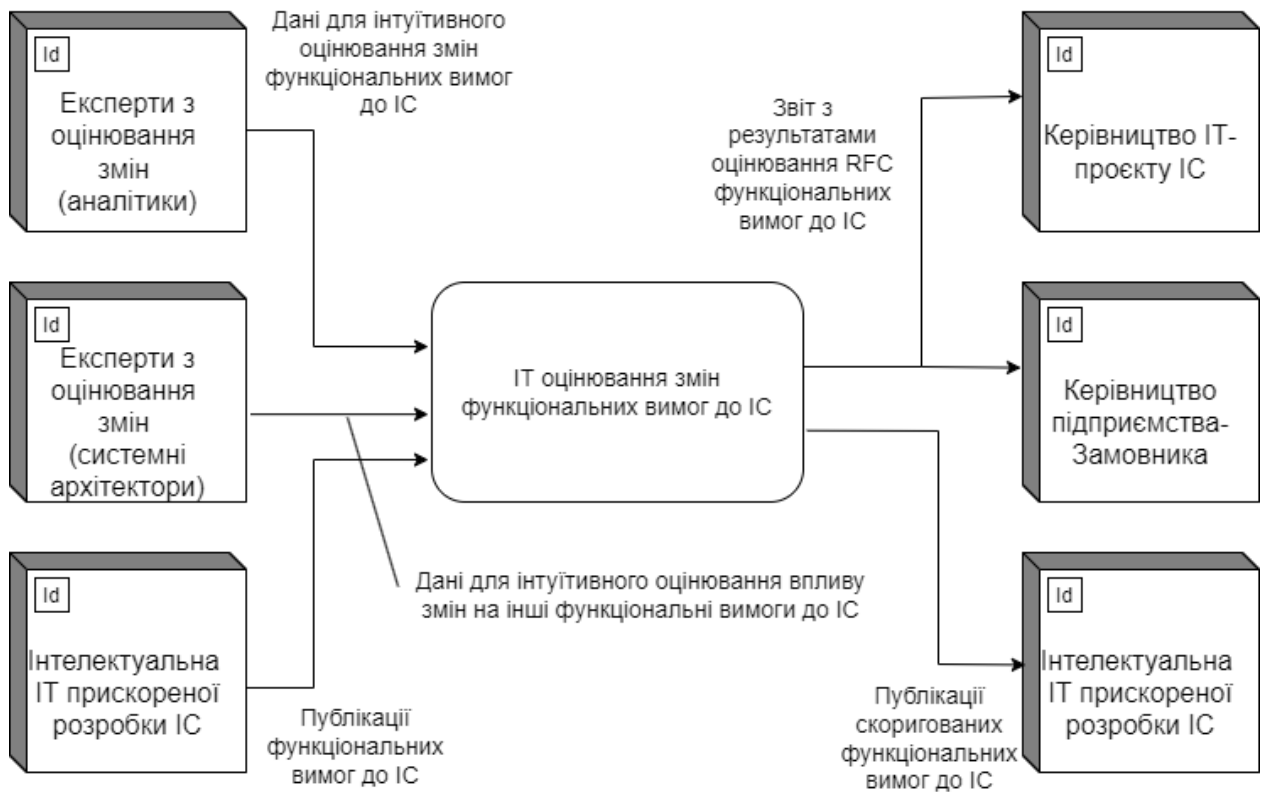


Рисунок 3.1 – Контекстна діаграма моделі потоків даних інформаційної технології оцінювання змін функціональних вимог до інформаційної системи

Діаграму декомпозиції першого рівня моделі потоків даних, яка описує розділення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, на окремі функціональні модулі та взаємодії цих модулів в межах ІС, наведено на рис. 3.2. На цій діаграмі показано тільки основні зовнішні та внутрішні потоки даних, щоб зменшити кількість елементів діаграми, поліпшити легкість прочитання цієї діаграми та спростити опис взаємодії функціональних модулів в межах ІТ.

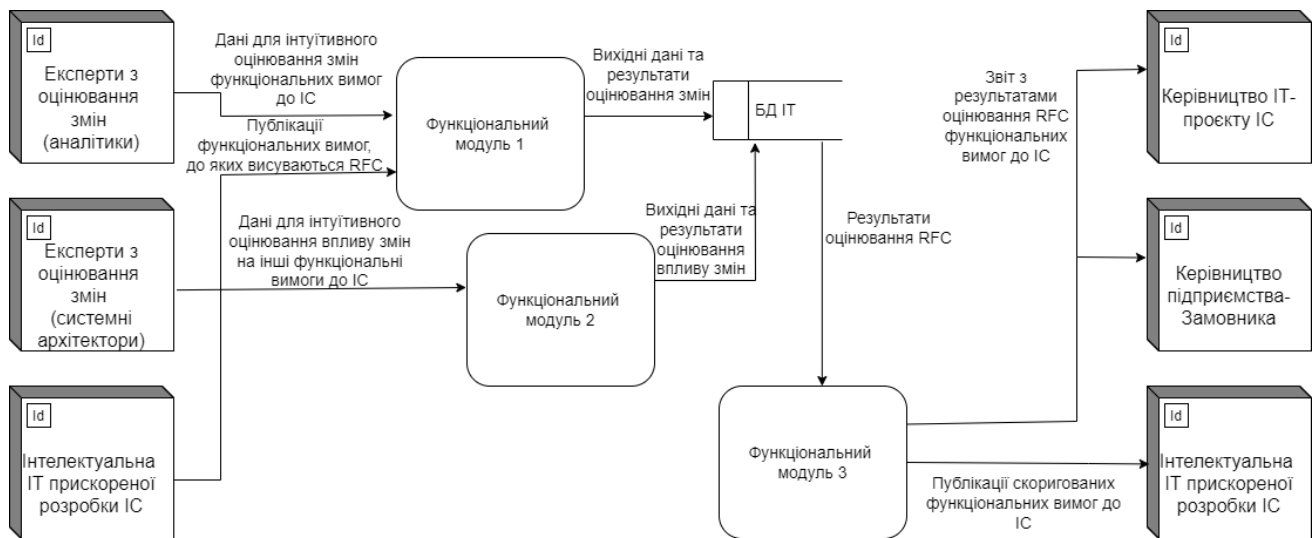


Рисунок 3.2 – Діаграма декомпозиції першого рівня моделі потоків даних інформаційної технології оцінювання змін функціональних вимог до інформаційної системи

ІС, яка реалізує ІТ оцінювання змін функціональних вимог до ІС, повинна функціонувати в діалоговому режимі.

ІС, яка реалізує ІТ оцінювання змін функціональних вимог до ІС, повинна взаємодіяти із кількістю майбутніх користувачів, яка може змінюватися від 1 до 100. Ці майбутні користувачі повинні володіти навичками візуального моделювання функціональних вимог до ІС та RFC функціональних вимог до ІС.

Для майбутніх користувачів ІС, яка реалізує ІТ оцінювання змін функціональних вимог до ІС, повинна бути забезпечена можливість працювати з цією ІС 24 години на добу, тому що вони можуть знаходитися в різних часових поясах.

ІС, яка реалізує ІТ оцінювання змін функціональних вимог до ІС, повинна забезпечувати зберігання інформації в умовах непередбачуваних відключень електроенергії.

3.1.2 Визначення функціональних вимог до інформаційної технології

В межах функціонального модуля 1 повинні виконуватися такі функції:

- формування категорій інтуїтивного оцінювання RFC функціональних вимог до ІС (функція 1.1);
- формування та публікація множини RFC функціональних вимог до ІС (функція 1.2);
- облік та контроль розподілення публікацій RFC функціональних вимог до ІС за категоріями оцінювання (функція 1.3);
- облік та контроль оцінювання публікацій RFC функціональних вимог в межах категорії оцінювання (функція 1.4).

В межах функціонального модуля 2 повинні виконуватися такі функції:

- формування категорій інтуїтивного оцінювання впливу RFC функціональних вимог до ІС на актуальні публікації функціональних вимог до ІС (функція 2.1);
- формування та публікація множини функціональних вимог до ІС, на які має вплив оцінюваний RFC (функція 2.2);
- облік та контроль розподілення публікацій множини функціональних вимог до ІС, на які має вплив оцінюваний RFC, за категоріями оцінювання (функція 2.3);
- облік та контроль оцінювання публікацій множини функціональних вимог до ІС, на які має вплив оцінюваний RFC, в межах категорії оцінювання (функція 2.4).

В межах функціонального модуля 3 повинні виконуватися такі функції:

- формування підсумкових результатів інтуїтивного оцінювання окремого RFC функціональної вимоги до ІС (функція 3.1);
- формування та публікація підсумкового документу «Звіт з результатами оцінювання RFC функціональних вимог до ІС» (функція 3.2);

– коригування публікацій неповних функціональних вимог до ІС, до яких були висунуті та оцінені RFC (функція 3.3).

Опис основних результатів виконання кожної функції наведено у табл. 3.1.

Таблиця 3.1 – Опис основних результатів виконання функцій інформаційної системи, що реалізує інформаційну технологію оцінювання змін функціональних вимог до інформаційної системи

№ функції	Опис результату виконання функції
1	2
Функція 1.1	Відеограма, яка відображує категорії інтуїтивного оцінювання («Big», «Small» та «Uncertain»), показник, який розділяє категорії «Big» та «Small», а також значення цього показника
Функція 1.2	Відеограма, яка відображує масив ідентифікованих карток з публікаціями неповних функціональних вимог до ІС та RFC, які висуваються до відповідних функціональних вимог
Функція 1.3	Відеограма, яка відображує результат розподілення ідентифікованих карток за категоріями інтуїтивного оцінювання «Big» та «Small». Масив даних з результатами розподілення ідентифікованих карток за категоріями інтуїтивного оцінювання «Big» та «Small».
Функція 1.4	Відеограма, яка відображує результат розподілення ідентифікованих карток за встановленою шкалою в межах обраної категорії інтуїтивного оцінювання («Big» або «Small») Масив даних з результатами розподілення ідентифікованих карток за встановленою шкалою в межах обраної категорії інтуїтивного оцінювання («Big» або «Small»).

Продовження таблиці 3.1

1	2
Функція 2.1	Відеограма, яка відображує категорії інтуїтивного оцінювання («Big», «Small» та «Uncertain»), показник, який розділяє категорії «Big» та «Small», а також значення цього показника.
Функція 2.2	Відеограма, яка відображує масив ідентифікованих карток з публікаціями множини функціональних вимог до ІС, на які має вплив оцінюваний RFC.
Функція 2.3	Відеограма, яка відображує результат розподілення ідентифікованих карток за категоріями інтуїтивного оцінювання «Big» та «Small». Масив даних з результатами розподілення ідентифікованих карток за категоріями інтуїтивного оцінювання «Big» та «Small».
Функція 2.4	Відеограма, яка відображує результат розподілення ідентифікованих карток за встановленою шкалою в межах обраної категорії інтуїтивного оцінювання («Big» або «Small») Масив даних з результатами розподілення ідентифікованих карток за встановленою шкалою в межах обраної категорії інтуїтивного оцінювання («Big» або «Small»).
Функція 3.1	Відеограма з переліком оцінюваних RFC функціональних вимог до ІС. Відеограма, яка відображує підсумкові результати інтуїтивного оцінювання за обраним RFC функціональної вимоги до ІС.

Кінець таблиці 3.1

1	2
Функція 3.2	Відеограма, яка відображує підсумковий документ «Звіт з результатами оцінювання RFC функціональних вимог до ІС». Файл з електронною версією підсумкового документу «Звіт з результатами оцінювання RFC функціональних вимог до ІС».
Функція 3.3	Масив даних з результатами коригування публікацій неповних функціональних вимог до ІС шляхом додавання до них публікацій RFC відповідних функціональних вимог.

Діаграми декомпозиції другого рівня моделі потоків даних, що описує розділення окремих функціональних модулів ІС, яка реалізує ІТ оцінювання змін функціональних вимог до ІС, та взаємодії цих функцій в межах модуля, наведено на рис. А.1, А.2 та А.3 відповідно у додатку А.

У функції 1.3 контролю підлягають:

- факт пропуску ходу конкретним учасником етапу 5 ансамблевого методу Size Ordering;
- факт пропуску ходу усіма учасниками етапу 5 ансамблевого методу Size Ordering Rule;
- факт відсутності ідентифікованих карток в категорії «Uncertain» як умова успішного завершення виконання цієї функції;
- факт відсутності ідентифікованих карток, які знаходяться на руках учасників етапу 6 ансамблевого методу Size Ordering Rule і не розподілені до однієї з категорій інтуїтивного оцінювання («Big» або «Small»).

У функції 1.4 контролю підлягають:

- факт пропуску ходу конкретним учасником Етапу 9 ансамблевого методу Size Ordering Rule;
- факт пропуску ходу усіма учасниками етапу 9 ансамблевого методу Size Ordering Rule.

У функції 2.3 контролю підлягають:

- факт пропуску ходу конкретним учасником етапу 5 ансамблевого методу Size Ordering Rule;

- факт пропуску ходу усіма учасниками етапу 5 ансамблевого методу Size Ordering Rule;

- факт відсутності ідентифікованих карток з публікаціями множини функціональних вимог до ІС, на які має вплив оцінюваний RFC, в категорії «Uncertain» як умова успішного завершення виконання цієї функції;

- факт відсутності ідентифікованих карток з публікаціями множини функціональних вимог до ІС, на які має вплив оцінюваний RFC, які знаходяться на руках учасників етапу 6 ансамблевого методу Size Ordering Rule і не розподілені до однієї з категорій інтуїтивного оцінювання («Big» або «Small»).

У функції 2.4 контролю підлягають:

- факт пропуску ходу конкретним учасником етапу 9 ансамблевого методу Size Ordering Rule;

- факт пропуску ходу усіма учасниками етапу 9 ансамблевого методу Size Ordering Rule.

3.1.3 Визначення вимог до видів забезпечень інформаційної технології

Основною вимогою до математичного забезпечення ІТ оцінювання змін функціональних вимог до ІС є вимога застосувати як основний метод оцінювання розроблений у підрозд. 2.2 ансамблевий метод Size Ordering Rule. Розподіл етапів цього методу між функціями ІТ оцінювання змін функціональних вимог до ІС наведено у табл. А.1 додатку А.

Основними вимогами до інформаційного забезпечення ІТ оцінювання змін функціональних вимог до ІС є такі:

- склад, структура та способи організації даних в ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинні забезпечувати ефективне та якісне виконання функцій, перелік яких та опис основних вихідних результатів наведено у п. 3.1.2;

- інформаційний обмін між функціональними модулями ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинен здійснюватися через централізовану БД;

- інформаційне забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинне мати інтерфейси або інші можливості здійснення інформаційного обміну між цією ІТ та інтелектуальною ІТ прискореної розробки ІС;

- інформаційне забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинне базуватися на використанні методик та алгоритмів класифікації та кодування інформації, які функціонують в межах інтелектуальної ІТ прискореної розробки ІС;

- для проєктування та розробки БД як ключового елемента інформаційного забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, рекомендується обрати СУБД Oracle як таку, що співпадає із СУБД інтелектуальної ІТ прискореної розробки ІС (остаточний вибір пропонується здійснити за результатами ескізного і технічного (або техноробочого) проєктування БД);

- під час проєктування та розробки інформаційного забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, необхідно дотримуватися прийнятих в Україні форматів представлення даних.

Основними вимогами до лінгвістичного забезпечення є такі:

- головною мовою, яка використовується в ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, є українська;

- допоміжною мовою, яка використовується в ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, є англійська;

- під час опису публікацій функціональних вимог до ІС та RFC функціональних вимог до ІС в ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинні використовуватися мови візуального моделювання, якими створено відповідні публікації;

- спеціальні вимоги до способів організації діалогу, до розробки та використання словників і тезаурусів, до опису синтаксису формалізованих мов співпадають із аналогічними вимогами до інтелектуальної ІТ прискореної розробки ІС.

Основними вимогами до програмного забезпечення є такі:

- як основна архітектура прикладного програмного забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинна розглядатися веббазована архітектура або їй подібні;

- програмне забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинно складатися із СУБД, прикладних програмних модулів, які забезпечують реалізацію заявлених у п. 3.1.2 функцій, та додаткових ІТ-продуктів, які можуть стати необхідними для ефективною та якісною реалізації окремих функцій;

- усі інструментальні засоби, які повинні використовуватися для створення прикладного програмного забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, повинні мати відповідні чинні ліцензії або підтвердження знаходження їх у вільному доступі.

Основні вимоги до технічного забезпечення ІС, що реалізує ІТ оцінювання змін функціональних вимог до ІС, визначаються, виходячи з аналогічних вимог до інтелектуальної ІТ прискореної розробки ІС.

3.2 Розробка елементів інформаційного забезпечення інформаційної технології

Головним компонентом інформаційного забезпечення ІТ оцінювання змін функціональних вимог до ІС є централізована БД. Тому в роботі основну увагу приділено розробці елементів цієї БД.

На першому етапі розробки БД ІТ оцінювання змін функціональних вимог до ІС було створено концептуальну схему даних цієї БД. Відомості про типи сутностей ІТ оцінювання змін функціональних вимог до ІС наведено в табл. 3.2.

Відомості про типи зв'язків між сутностями ІТ оцінювання змін функціональних вимог до ІС наведено в табл. 3.3.

Відомості про атрибути сутностей ІТ оцінювання змін функціональних вимог до ІС наведено в табл. 3.4.

Таблиця 3.2 – Відомості про типи сутностей

Ім'я типу сутності	Опис	Особливості використання
1	2	3
ІТ_проект	Стислий опис ІТ-проекту, в межах якого вирішується задача оцінювання змін функціональних вимог до ІС	Використовується для встановлення зв'язку задачі оцінювання змін з іншими операціями ІТ-проекту
Задача_оцінювання	Стислий опис задачі оцінювання змін функціональних вимог та значень її загальних показників	Використовується для ідентифікації задачі оцінювання змін функціональних вимог як операції ІТ-проекту

Продовження таблиці 3.2

1	2	3
Стадія_оцінювання	Довідник можливих стадій оцінювання (за замовчуванням – дві стадії: оцінювання RFC (1) та оцінювання впливу RFC на множину актуальних функціональних вимог до IC (2))	Використовується для визначення і подальшого журналювання факту перебування вирішення задачі оцінювання змін функціональних вимог на конкретній стадії
Категорії_оцінювання	Довідник можливих категорій оцінювання («Big», «Small» та «Uncertain»)	Використовується для визначення категорій оцінювання, до яких належать ідентифіковані картки
Ідентифіковані_картки	Публікації актуальних функціональних вимог до IC та RFC, які до них висуваються	Використовується для візуалізації публікацій функціональних вимог до IC та RFC, які до них висуваються
Клас_експертів	Довідник можливих класів експертів (аналітики або системні архітектори)	Використовується для визначення ролі експерта у вирішенні задачі оцінювання
Довідник_посад	Довідник посад експертів	Використовується для визначення ролі експерта у вирішенні задачі оцінювання

Кінець таблиці 3.2

1	2	3
Експерти	Опис фахівців, які виступають в ролі експертів у вирішенні задачі оцінювання	Використовується для визначення авторів рішень з оцінювання змін функціональних вимог до ІС
Журнал_оцінювання	Опис фактів, які виникають під час вирішення задачі оцінювання	Використовується для зберігання особливостей ходу вирішення конкретної задачі оцінювання
Підсумки_оцінювання	Опис результатів вирішення задачі оцінювання змін функціональних вимог до ІС на кожній стадії	Використовується для зберігання даних про особливості підсумкового вирішення конкретної задачі оцінювання

Таблиця 3.3 – Відомості про типи зв'язків

Тип сутності	Тип зв'язку	Тип сутності	Ступінь зв'язку
1	2	3	4
ІТ_проект	Публікує	Ідентифіковані_картки	1:М
Експерти	Належать	Клас_експертів	М:1
Довідник_посад	Визначає посаду	Експерти	1:М
Задача_оцінювання	Вирішується	Експерти	М:М
Задача_оцінювання	Фіксує результати вирішення	Журнал_оцінювання	1:М

Кінець таблиці 3.3

1	2	3	4
Задача_оцінювання	Отримує підсумкові рішення	Підсумки_оцінювання	1:M
Стадія_оцінювання	Визначає особливості рішення	Журнал_оцінювання	1:M
Стадія_оцінювання	Визначає особливості підсумкових рішень	Підсумки_оцінювання	1:M
Категорії_оцінювання	Визначає особливості рішення	Журнал_оцінювання	1:M
Категорії_оцінювання	Визначає особливості підсумкових рішень	Підсумки_оцінювання	1:M
Ідентифіковані_картки	Отримують оцінки	Журнал_оцінювання	1:M
Ідентифіковані_картки	Отримують підсумкові оцінки	Підсумки_оцінювання	1:M

Таблиця 3.4 – Відомості про атрибути

Тип сутності	Атрибут	Опис	Тип даних, довжина	Обмеження	Припустимість Null
1	2	3	4	5	6
ІТ_проект	ID_проекта	Ідентифікатор проекту	Числовий, до 5 розрядів	Первинний ключ	Ні
	Назва_проекта	Назва ІТ-проекта	Символьний, до 255 символів		Ні
	Дата_початку_проекта	Дата початку ІТ-проекта	Дата		Ні
	Дата_завершення_проекта	Дата завершення ІТ-проекта	Дата		Так
Задача_оцінювання	ID_задачі	Ідентифікатор задачі оцінювання	Числовий, до 6 розрядів	Первинний ключ	Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Задача _оціню вання	Дата_та_час_поча тку_вирішення	Дата та час початку вирішення задачі оцінювання	Дата/час		Ні
	Значення_показн ика_стадія_1	Значення показника, який розділяє категорії «Big» та «Small» на стадії 1	Числовий , до 4 розрядів, два розряди після коми		Ні
	Значення_показн ика_стадія_2	Значення показника, який розділяє категорії «Big» та «Small» на стадії 2	Числовий , до 4 розрядів, два розряди після коми		Ні
Стадія _оціню вання	ID_стадії	Ідентифікат ор стадії оцінювання	Числовий , до 2 розрядів	Первинний ключ	Ні
	Назва_стадії	Назва стадії оцінювання	Символь ний, до 15 символів		Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Категорія оцінювання	ID_категорії	Ідентифікатор категорії оцінювання	Числовий, до 2 розрядів	Первинний ключ	Ні
	Назва_категорії	Назва категорії оцінювання	Символьний, до 15 символів		Ні
Ідентифіковані картки	ID_картки	Ідентифікатор картки	Числовий, до 6 розрядів	Первинний ключ	Ні
Ідентифіковані картки	ID_проекта	Ідентифікатор проекту	Числовий, до 5 розрядів	Зовнішній ключ з таблиці ІТ_проект	Ні
	Дата_час_створення_картки	Дата та час створення ідентифікованої картки	Дата/час		Ні
	Зміст_картки	Публікації функціональних вимог та RFC до них	Файл		Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Клас_експертів	ID_класу	Ідентифікатор класу експертів	Числовий, до 6 розрядів	Первинний ключ	Ні
	Найменування_класу	Найменування класу експертів	Символьний, до 25 символів		Ні
Довідник_посад	ID_посади	Ідентифікатор посади експерта	Числовий, до 6 розрядів	Первинний ключ	Ні
Ідентифіковані_картки	ID_проекта	Ідентифікатор проекту	Числовий, до 5 розрядів	Зовнішній ключ з таблиці ІТ_проект	Ні
	Дата_час_створення_картки	Дата та час створення ідентифікованої картки	Дата/час		Ні
	Зміст_картки	Публікації функціональних вимог та RFC до них	Файл		Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Довідник_посад	Назва_посади	Назва посади експерта	Символьний, до 25 символів		Ні
	Дата_початку_існування	Дата створення посади	Дата		Ні
	Дата_завершення_існування	Дата ліквідації посади	Дата		Так
Експерти	ID_експерта	Ідентифікатор експерта	Числовий, до 6 розрядів	Первинний ключ	Ні
	ID_класу	Ідентифікатор класу експерта	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Клас_експертів	Ні
	Прізвище	Прізвище експерта	Символьний, до 25 символів		Ні
	Ім'я	Ім'я експерта	Символьний, до 25 символів		Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Експерти	По_батькові	По батькові експерта	Символьний, до 25 символів		Так
	ID_посади	Ідентифікатор посади експерта	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Довідник_посад	Ні
Журнал оцінювання	ID_проекта	Ідентифікатор ІТ-проекта	Числовий, до 5 розрядів	Зовнішній ключ з таблиці ІТ_проект	Ні
	ID_задачі	Ідентифікатор задачі оцінювання	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Задача_оцінювання	Ні
	ID_стадії	Ідентифікатор стадії оцінювання	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Стадії_оцінювання	Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Журнал оцінювання	ID_картки	Ідентифікатор картки	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Ідентифіковані_картки	Так
	ID_експерта	Ідентифікатор експерта	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Експерти	Ні
	ID_класу	Ідентифікатор класу експерта	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Клас_експертів	Ні
	ID_посади	Ідентифікатор посади експерта	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Довідник_посад	Ні
	Участь_у_вирішенні_задачі	Факт участі експерта у вирішенні задачі оцінювання	Числовий, до 1 розряду		Так

Продовження таблиці 3.4

1	2	3	4	5	6
Журнал_оцінювання	ID_запропонованої_категорії	Ідентифікатор категорії оцінювання, запропонованої експертом	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Категорії_оцінювання	Так
	Запропоноване_значення_шкали	Значення шкали оцінювання, запропоноване експертом для картки	Числовий, до 4 розрядів, 2 розряди після коми		Так
	Пропозиція_обговорення	Факт пропонування експертом обговорення картки	Числовий, до 1 розряду		Так
Підсумки_оцінювання	ID_проекта	Ідентифікатор ІТ-проекта	Числовий, до 5 розрядів	Зовнішній ключ з таблиці ІТ_проект	Ні

Продовження таблиці 3.4

1	2	3	4	5	6
Підсумки_оцінювання	ID_задачі	Ідентифікатор задачі оцінювання	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Задача_оцінювання	Ні
	ID_стадії	Ідентифікатор стадії оцінювання	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Стадії_оцінювання	Ні
	ID_картки	Ідентифікатор оціненої картки	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Ідентифіковані_картки	Ні
	ID_категорії	Ідентифікатор підсумкової категорії оцінювання	Числовий, до 6 розрядів	Зовнішній ключ з таблиці Категорії_оцінювання	Ні

Кінець таблиці 3.4

1	2	3	4	5	6
Підсумки оцінювання	Запропоноване значення шкали	Підсумкове значення шкали оцінювання для картки	Числовий, до 4 розрядів, 2 розряди після коми		Ні

Концептуальну схему даних ІТ оцінювання змін функціональних вимог до ІС наведено на рис. 3.3.

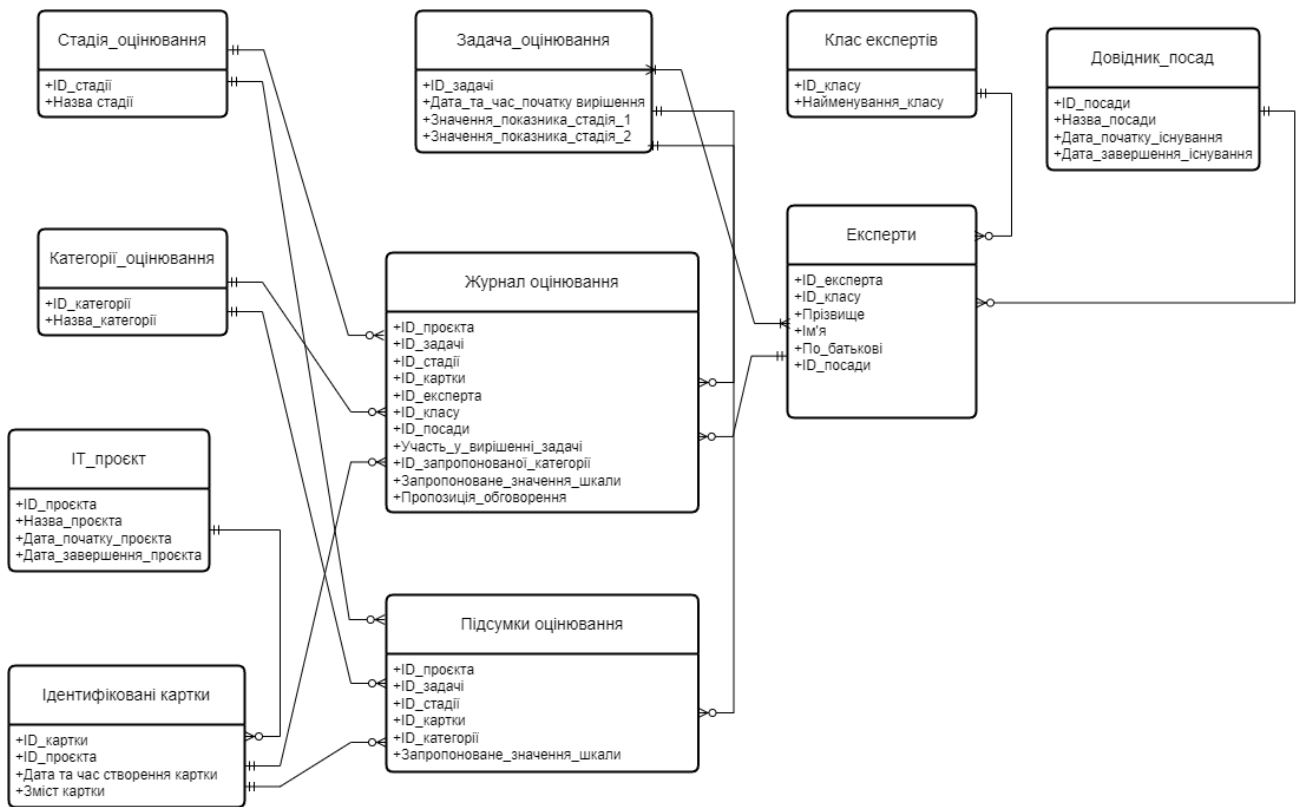


Рисунок 3.3 – Концептуальна схема даних інформаційної технології оцінювання змін функціональних вимог до інформаційної системи

В процесі побудови логічної схеми даних для забезпечення відповідності очікуваної схеми третій нормальній формі було прийнято рішення усунути зв'язок «багато до багатьох» між сутностями «Задача_оцінювання» та «Експерти». Для вирішення цієї проблеми було створено додаткову сутність «Учасники_задачі_оцінювання», опис атрибутів якої наведено у табл. 3.5.

Таблиця 3.5 – Відомості про атрибути додаткової сутності «Учасники_задачі_оцінювання»

Тип сутності	Атрибут	Опис	Тип даних, довжина	Обмеження	Припустимість Null
1	2	3	4	5	6
Учасники_задачі_оцінювання	ID_задачі	Ідентифікатор задачі оцінювання	Числовий, до 6 розрядів	Складовий первинний ключ	Ні
	ID_експерта	Ідентифікатор експерта, який бере участь у вирішенні задачі оцінювання	Числовий, до 6 розрядів	Складовий первинний ключ	Ні

Кінець таблиці 3.5

1	2	3	4	5	6
Учасники_задачі_оцінювання	Дата_призначення	Дата призначення експерта для вирішення задачі оцінювання	Дата		Ні
	Джерело_призначення	Номер рішення про призначення експерта для вирішення задачі оцінювання	Числовий, до 6 розрядів		Ні

Логічну схему даних ІТ оцінювання змін функціональних вимог до ІС наведено на рис. 3.3.

Фрагменти логічної схеми даних інтелектуальної ІТ прискореної розробки ІС, які описують фрагменти БД, призначені для зберігання вимог до ІС та публікацій цих вимог у вигляді структурних або об'єктно-орієнтованих візуальних моделей, наведено у додатку Б.

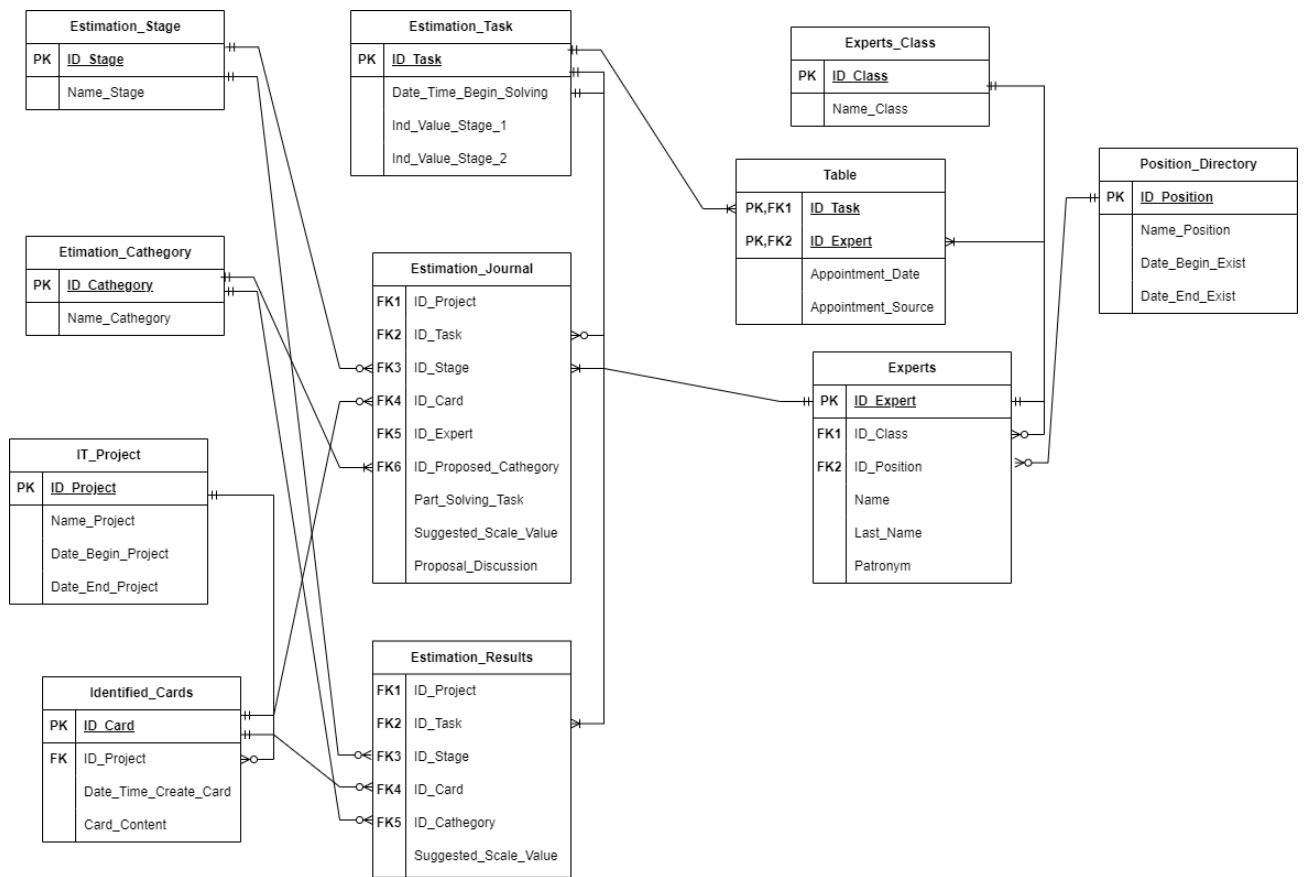


Рисунок 3.4 – Логічна схема даних інформаційної технології оцінювання змін функціональних вимог до інформаційної системи

Для реалізації розробленої логічної схеми даних пропонується використати ту ж СУБД, що і для інтелектуальної ІТ прискореної розробки ІС, а саме СУБД Oracle XE 11g [30].

3.3 Розробка елементів алгоритмічного забезпечення інформаційної технології

Головною особливістю автоматизованого виконання роботи «Окресліть масштаби змін» є необхідність оцінювання таких різновидів зусиль:

а) зусилля, спрямовані на реалізацію опублікованих RFC конкретної функціональної вимоги до ІС;

б) зусилля, спрямовані на реалізацію змін у всіх інших функціональних вимогах до ІС, на які впливає опублікований RFC до функціональної вимоги.

Тому узагальнений алгоритм виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС є послідовністю таких кроків.

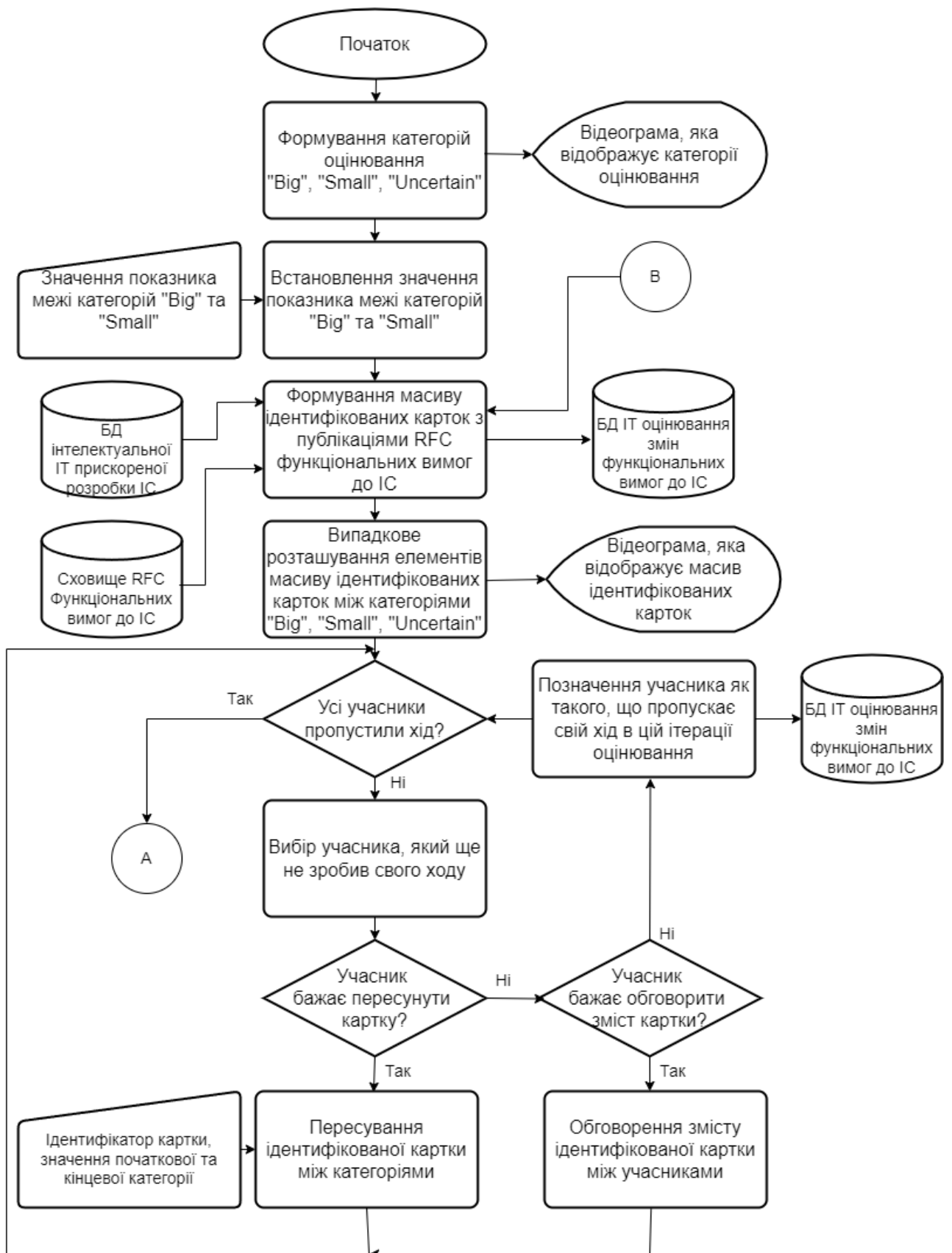
Крок 1. Оцінювання зусиль на реалізацію кожного окремого RFC функціональної вимоги до ІС з множини опублікованих RFC.

Крок 2. Оцінювання зусиль на реалізацію змін множини функціональних вимог, на які впливає кожний окремий RFC функціональної вимоги до ІС з множини опублікованих RFC.

Крок 3. Формування звіту з результатами оцінювання. Якщо приймається рішення про необхідність повторного оцінювання окремих RFC функціональних вимог до ІС, то визначення цих RFC та повернення до кроку 1. В іншому випадку – публікація звіту з результатами оцінювання і завершення роботи алгоритму.

Схему виконання кроку 1 узагальненого алгоритму виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС наведено на рис. 3.5.

Схему виконання кроку 2 узагальненого алгоритму виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС наведено на рис. 3.6.



3.5 – Схема виконання кроку 1 узагальненого алгоритму

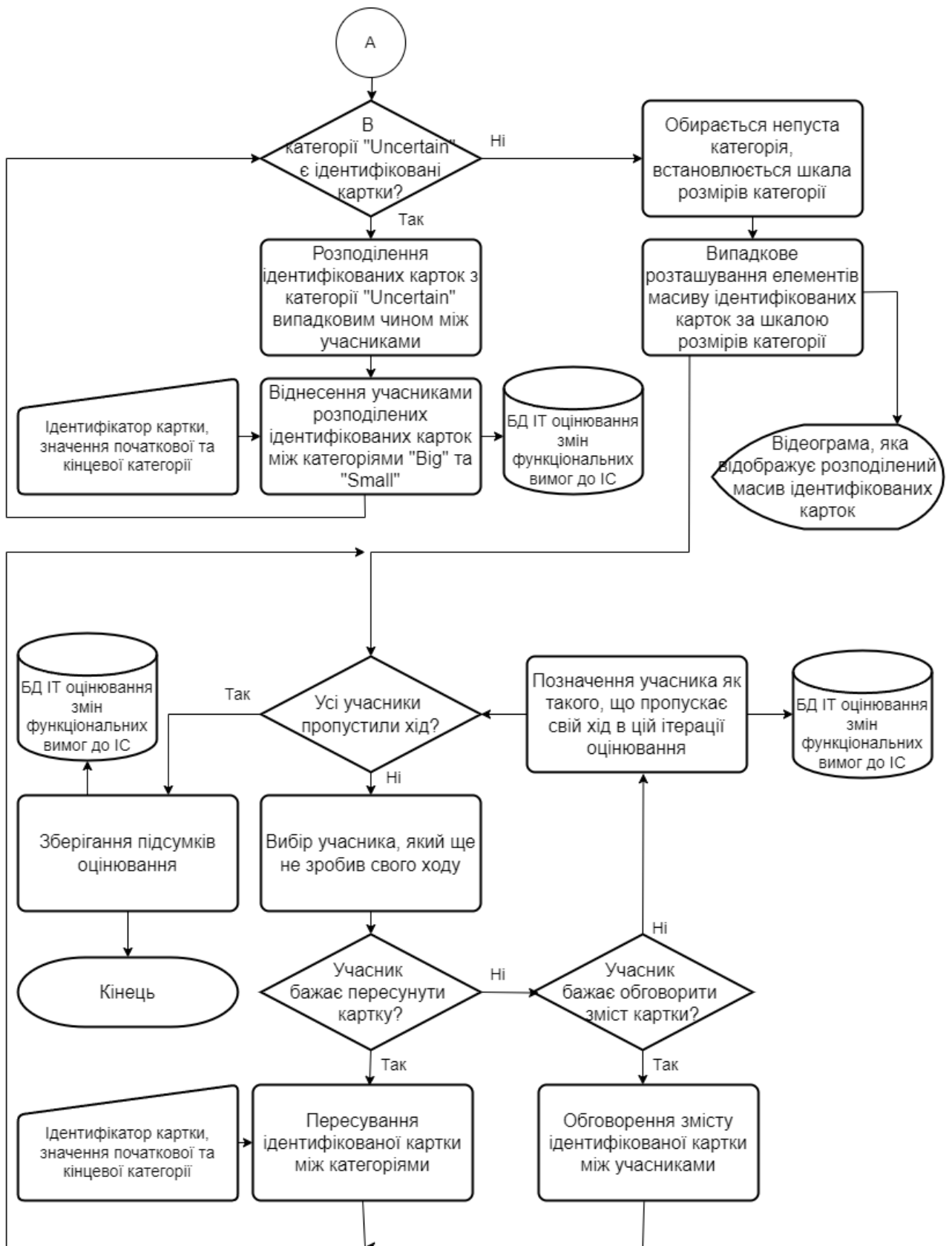


Рисунок 3.5, аркуш 2

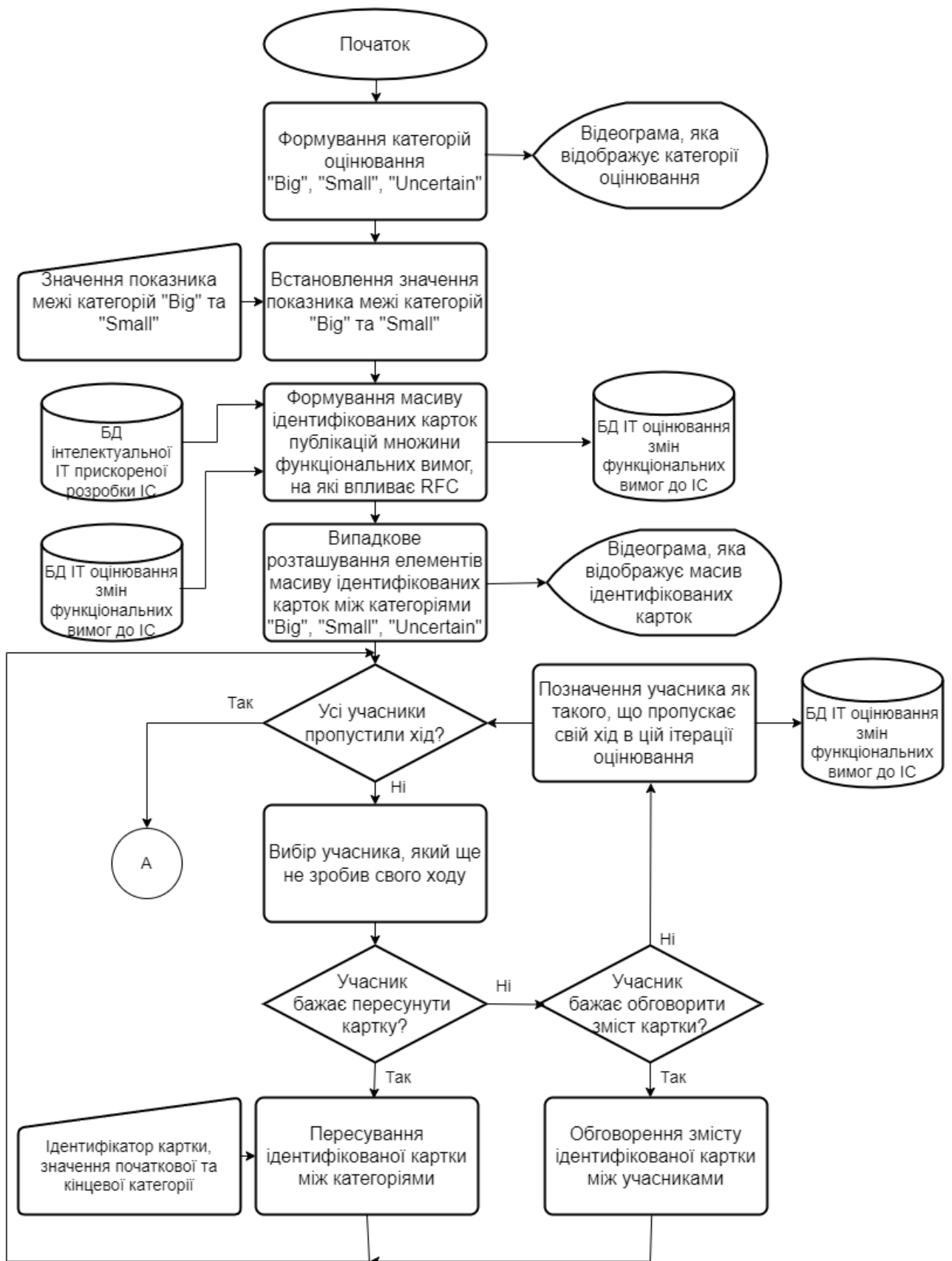


Рисунок 3.6 – Схема виконання кроку 2 узагальненого алгоритму

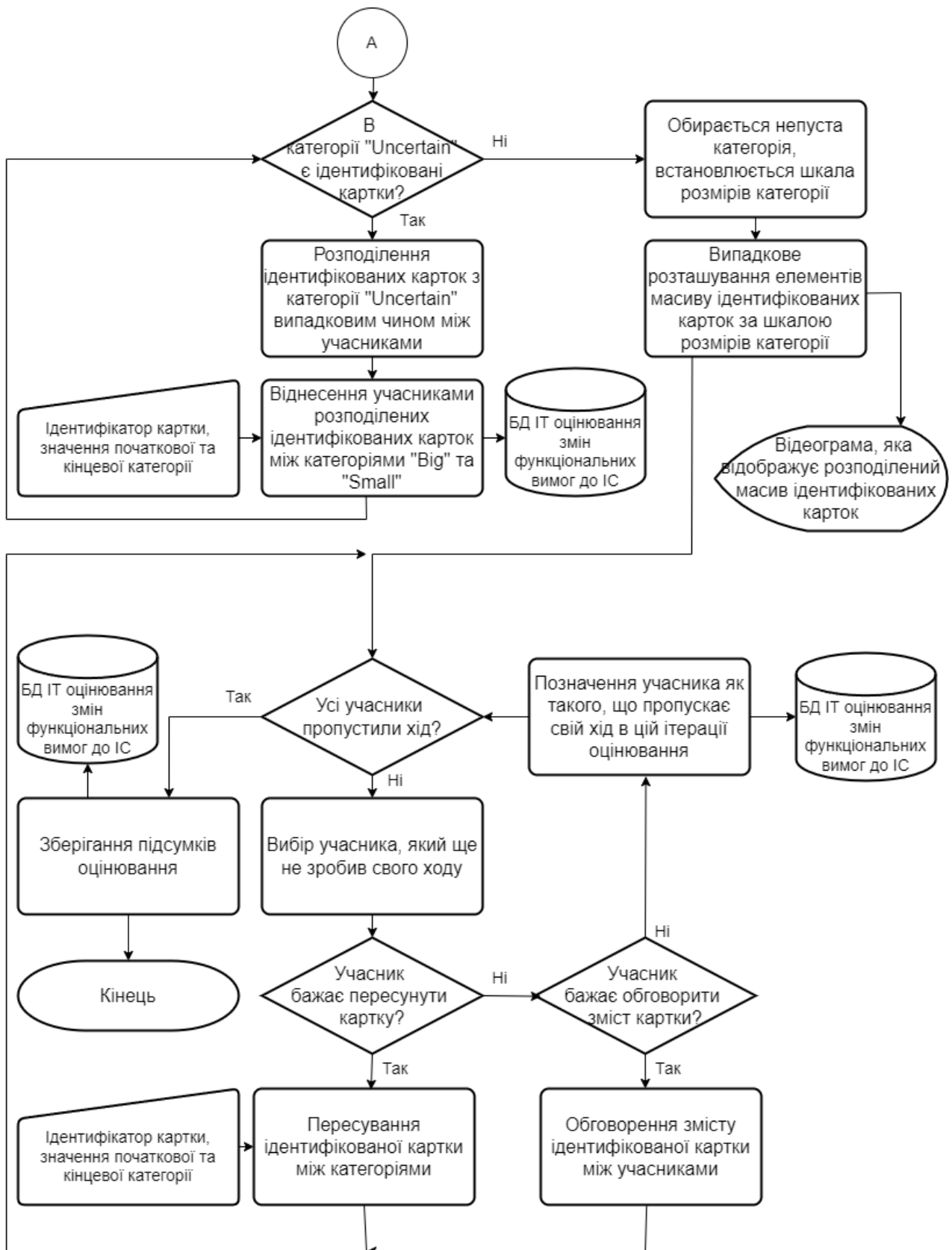


Рисунок 3.6, аркуш 2

Схему виконання кроку 3 узагальненого алгоритму виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС наведено на рис. 3.7.

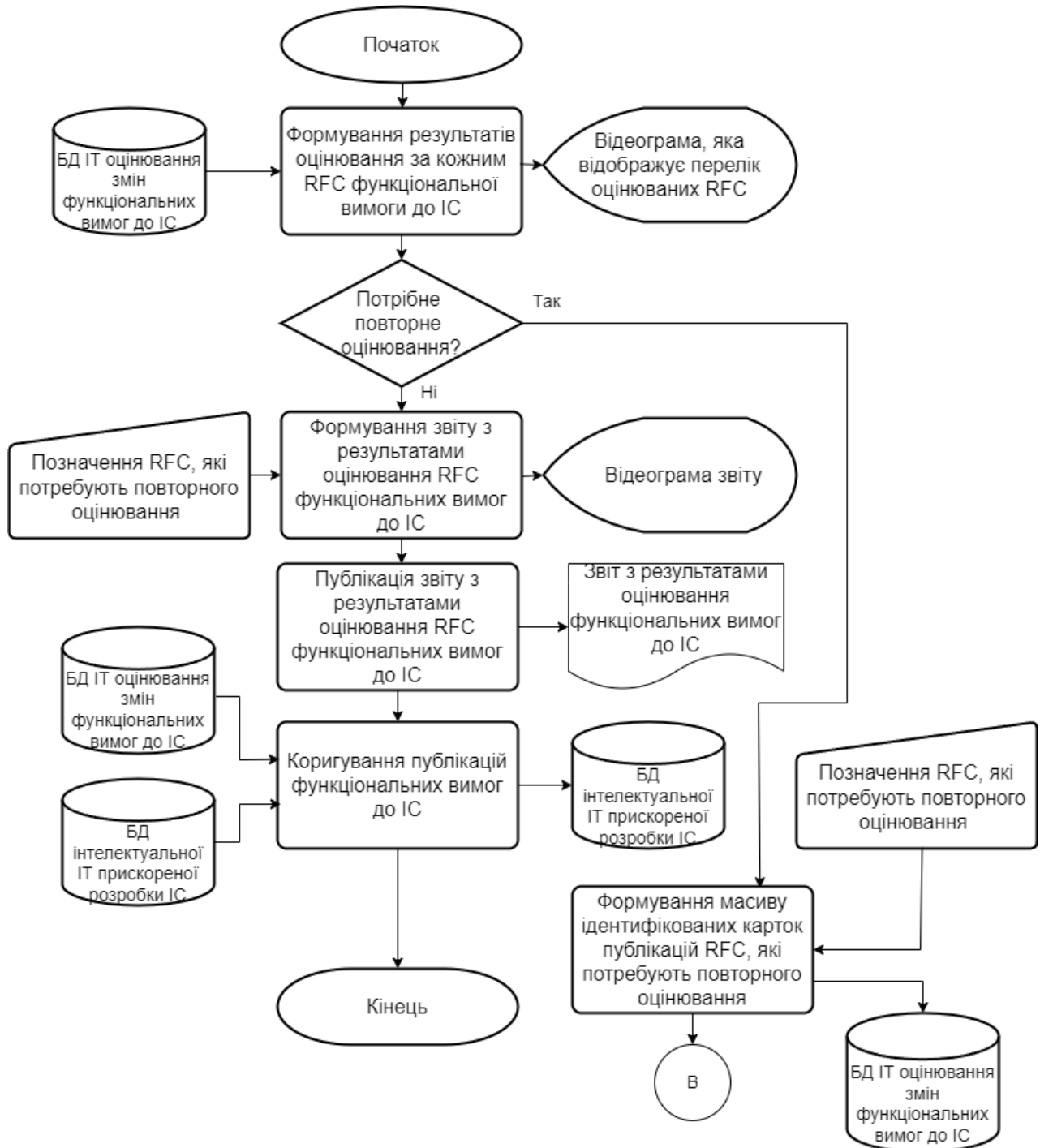


Рисунок 3.7 – Схема виконання кроку 3 узагальненого алгоритму

Візуальний аналіз схем виконання кроку 1 та кроку 2 (рис. 3.5 та рис. 3.6 відповідно) узагальненого алгоритму виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС дозволяє стверджувати про майже повну тотожність тих частин схем, які відповідають за реалізацію етапів 4-9 ансамблевого методу Size Ordering Rule. Цю тотожність треба враховувати під час розробки елементів програмного забезпечення ІТ оцінювання змін функціональних вимог до ІС для ліквідації дублювання програмних модулів, що реалізують операції таких пар функцій:

- функція 1.2 та функція 2.2;
- функція 1.3 та функція 2.3;
- функція 1.4 та функція 2.4.

Розроблений алгоритм виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС є платформонезалежним і може уточнюватися після вибору технологічного стека реалізації зазначеної ІТ. Під технологічним стеком тут розуміється набір технологій, які використовуються разом для розробки та підтримки програмного забезпечення [31].

3.4 Розробка елементів програмного забезпечення інформаційної технології

Для визначення функціональних вимог, які висуваються до прикладного програмного забезпечення ІТ оцінювання змін функціональних вимог до ІС, розроблено діаграми прецедентів. Ці діаграми у вигляді окремих прецедентів описують сукупність операцій прикладного програмного забезпечення, які необхідно виконати для реалізації сценарію виконання окремої функціональної вимоги, що висунута до ІТ оцінювання змін функціональних вимог до ІС. Окрім

цього, діаграми прецедентів надають візуальне представлення різних способів взаємодії користувача з прикладним програмним забезпеченням ІТ, а також демонструють перелік прецедентів, які прикладне програмне забезпечення ІТ надає окремому користувачеві.

Під час визначення функціональних вимог до прикладного програмного забезпечення ІТ оцінювання змін функціональних вимог до ІС враховувалися такі особливості виконання цієї роботи:

- межі кожної діаграми прецедентів визначаються межами окремих функцій ІТ оцінювання змін функціональних вимог до ІС, встановленими на діаграмах моделі потоків даних цієї ІТ (див. рис. А.1, рис. А.2 та рис. А.3);

- перелік ролей користувачів кожної діаграми прецедентів визначається в загальному випадку переліком зовнішніх сутностей, які позначають користувачів на діаграмах моделі потоків даних цієї ІТ (див. рис. А.1, рис. А.2 та рис. А.3);

- перелік прецедентів кожної діаграми прецедентів визначається в загальному випадку переліком окремих елементів схем виконання кроків узагальненого алгоритму виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС (див. рис. 3.5, рис. 3.6 та рис. 3.7).

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 1.1 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.1 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 1.2 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.2 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 1.3 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.3 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 1.4 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В. додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 2.1 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.5 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 2.2 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.6 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 2.3 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.7 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 2.4 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.8 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 3.1 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.9 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 3.2 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.10 додатку В.

Діаграму прецедентів, яка описує функціональну вимогу до прикладного програмного забезпечення, що повинно реалізувати функцію 3.3 ІТ оцінювання змін функціональних вимог до ІС, наведено на рис. В.11 додатку В.

Аналіз діаграм прецедентів, які описують функціональні вимоги до прикладного програмного забезпечення ІТ оцінювання змін функціональних вимог до ІС, показує наявність таких прецедентів, які повторюються в різних публікаціях вимог. Це свідчить про повне або часткове дублювання функціональних вимог до програмного забезпечення цієї ІТ. Для ліквідації

такого дублювання було запропоновано скористатися методом аналізу сценаріїв використання, розробленим Алістером Коберном. Цей метод у [32] представлено як послідовність з трьох етапів. Перший етап аналізу сценаріїв використання полягає у виділенні базового сценарію, що описує ті дії системи, які будуть повторюватися найчастіше незалежно від типу запиту діючої особи і умов виконання цих запитів. Другий етап аналізу сценаріїв використання полягає у описі зв'язків між вимогами до програмного забезпечення одна з іншою. Для цього краще за все використовувати діаграми прецедентів. Третій етап аналізу сценаріїв використання полягає у виявленні взаємовиключних вимог і здійсненні дроблення сценаріїв, базуючись на описах окремих вимог і описі взаємозв'язків вимог в рамках створюваної системи.

В процесі виконання першого етапу методу аналізу сценаріїв використання треба виявити прецеденти діаграм прецедентів, які повторюються у публікаціях різних функціональних вимог до програмного забезпечення ІТ. Результати цього виявлення наведено у табл. 3.6.

Таблиця 3.6 – Перелік прецедентів, що повторюються у публікаціях функціональних вимог до програмного забезпечення інформаційної технології оцінювання змін у функціональних вимогах до інформаційної системи

Назва прецеденту	Позначення функції ІТ
1	2
Формування категорій оцінювання «Big», «Small» та «Uncertain»	Функція 1.1
	Функція 2.1
Встановлення значення показника меж категорій «Big» та «Small»	Функція 1.1
	Функція 2.1
Вибір учасника, який ще не зробив свій хід	Функція 1.3
	Функція 1.4
	Функція 2.3
	Функція 2.4

Кінець таблиці 3.6

1	2
Пересування ідентифікованої картки між категоріями	Функція 1.3
	Функція 1.4
	Функція 2.3
	Функція 2.4
Обговорення змісту ідентифікованої картки між учасниками	Функція 1.3
	Функція 1.4
	Функція 2.3
	Функція 2.4
Позначення учасника як такого, що пропускає свій хід	Функція 1.3
	Функція 1.4
	Функція 2.3
	Функція 2.4
Розподілення ідентифікованих карток з категорії «Uncertain» випадковим чином між учасниками	Функція 1.3
	Функція 2.3
Віднесення учасниками розподілених ідентифікованих карток між категоріями «Big» та «Small»	Функція 1.3
	Функція 2.3
Для непустиї категорії встановлюється шкала розмірів	Функція 1.4
	Функція 2.4
Випадкове розташування елементів масиву ідентифікованих карток за шкалою розмірів категорії	Функція 1.4
	Функція 2.4

В табл. 3.6 розглянуто тільки ті прецеденти, які повністю дублюються в публікаціях різних функціональних вимог до прикладного програмного забезпечення ІТ оцінювання змін функціональних вимог до ІС. Але є ще ситуація часткового дублювання, яка виявляється в результаті порівняльного аналізу таких прецедентів:

а) прецедент «Формування масиву ідентифікованих карток з публікаціями RFC функціональних вимог до ІС», який використовується в публікації вимоги до функції 1.2;

б) прецедент «Формування масиву ідентифікованих карток публікацій множини функціональних вимог, на які впливає RFC», який використовується в публікації вимоги до функції 2.2;

в) прецедент «Формування масиву ідентифікованих карток публікацій RFC, які потребують повторного оцінювання», який використовується в публікації вимоги до функції 3.1.

Для ліквідації цієї ситуації запропоновано представити ці прецеденти так, як показано на рис. 3.8.

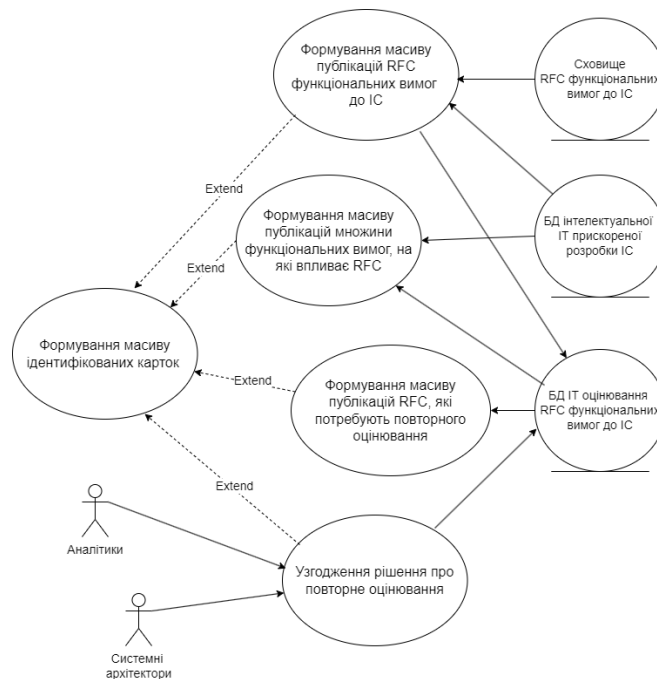


Рисунок 3.8 – Use Case діаграма, яка описує результат ліквідації ситуації часткового дублювання прецедентів у публікаціях функціональних вимог до прикладного програмного забезпечення розроблюваної інформаційної технології

Загальний результат виконання методу аналізу сценаріїв використання наведено на рис. 3.9.

На цьому рисунку прямими лініями виділено групи прецедентів, які рекомендується об'єднати у єдині програмні модулі. Таке об'єднання дає змогу представити прикладне програмне забезпечення ІТ оцінювання змін функціональних вимог до ІС як сукупність програмних модулів, які частково пов'язані між собою. Схему розподілення цих модулів між виконавцями робіт з їх розробки та між спринтами (проміжками часу, який відводиться на розробку одного модуля) наведено на рис. 3.10.

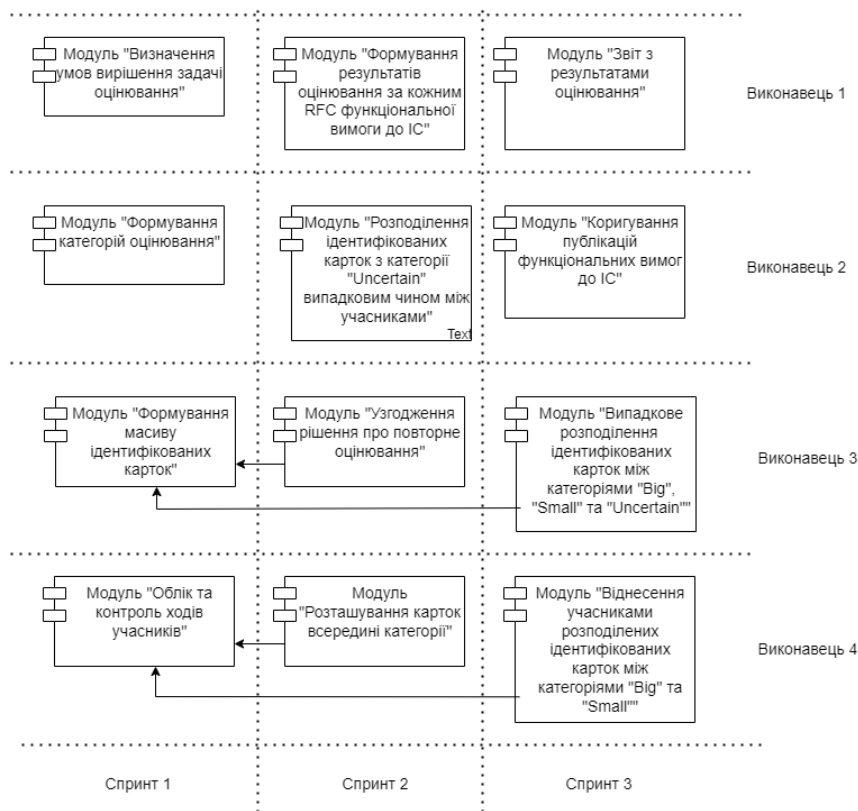


Рисунок 3.10 – Схема розподілення програмних модулів прикладного програмного забезпечення розроблюваної інформаційної технології між виконавцями та між спринтами виконання ІТ-проєкту

Перелік прецедентів, які повинні бути реалізовані в рамках встановлених модулів, наведено у табл. 3.7.

Таблиця 3.7 – Перелік прецедентів, які повинні бути реалізовані в рамках встановлених модулів прикладного програмного забезпечення розроблюваної інформаційної технології

Назва модуля	Назва прецеденту
1	2
«Визначення умов вирішення задачі оцінювання»	Визначення умов вирішення задачі оцінювання
«Формування результатів оцінювання за кожним RFC функціональної вимоги до ІС»	Формування результатів оцінювання за кожним RFC функціональної вимоги до ІС
«Звіт з результатами оцінювання»	Формування звіту з результатами оцінювання RFC функціональних вимог до ІС
	Публікація звіту з результатами оцінювання RFC функціональних вимог до ІС
	Розсилання звіту з результатами оцінювання RFC функціональних вимог до ІС зацікавленим особам ІТ-проєкту
«Формування категорій оцінювання»	Формування категорій оцінювання «Big», «Small» та «Uncertain»
	Встановлення значення показника меж категорій «Big» та «Small»
«Розподілення ідентифікованих карток з категорії «Uncertain» випадковим чином між учасниками»	Розподілення ідентифікованих карток з категорії «Uncertain» випадковим чином між учасниками

Продовження таблиці 3.7

1	2
«Коригування публікацій функціональних вимог до ІС»	Коригування публікацій функціональних вимог до ІС
«Формування масиву ідентифікованих карток»	Формування масиву ідентифікованих карток
	Формування масиву публікацій RFC функціональних вимог до ІС
	Формування масиву публікацій множини функціональних вимог, на які впливає RFC
	Формування масиву публікацій RFC, які потребують повторного оцінювання
«Узгодження рішення про повторне оцінювання»	Узгодження рішення про повторне оцінювання
«Випадкове розподілення ідентифікованих карток між категоріями «Big», «Small» та «Uncertain»»	Випадкове розподілення ідентифікованих карток між категоріями «Big», «Small» та «Uncertain»
«Облік та контроль ходів учасників»	Вибір учасника, який ще не зробив свій хід
	Позначення учасника як такого, що пропускає свій хід
	Обговорення змісту ідентифікованої картки між учасниками

Кінець таблиці 3.7

1	2
«Облік та контроль ходів учасників»	Пересування ідентифікованої картки між категоріями
«Розташування карток всередині категорії»	Випадкове розташування елементів масиву ідентифікованих карток за шкалою розмірів категорії Для непустиї категорії встановлюється шкала розмірів
«Віднесення учасниками розподілених ідентифікованих карток між категоріями «Big» та «Small»»	Віднесення учасниками розподілених ідентифікованих карток між категоріями «Big» та «Small»

3.5 Висновки до третього розділу

У третьому розділі розглянуто основні рішення зі створення ІТ оцінювання змін функціональних вимог до ІС. У підрозділі 3.1 було визначено основні вимоги до ІТ в цілому, до функцій ІТ та до видів забезпечень цієї ІТ. Опис взаємозв'язків між функціями ІТ було зроблено у вигляді моделі потоків даних, діаграми якої було наведено у підрозділі 3.1 та у додатку А.

На основі результатів визначення функціональних вимог до створюваної ІТ, було розроблено концептуальну і логічну схеми БД цієї ІТ (див. рис. 3.3 та рис. 3.4). Оскільки створювану ІТ планується використовувати у взаємодії з інтелектуальною ІТ прискореної розробки ІС, як СУБД було обрано СУБД Oracle XE 11g.

Виходячи з результатів розробки моделі потоків даних та визначення сценаріїв виконання окремих функцій, було розроблено узагальнений алгоритм

виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС та схеми алгоритмів (див. рис. 3.5, рис. 3.6 та рис. 3.7) виконання окремих кроків цього узагальненого алгоритму.

На основі функціональних вимог до окремих функцій ІТ оцінювання змін функціональних вимог до ІС, розробленої моделі потоків даних та розроблених схем алгоритмів виконання кроків узагальненого алгоритму було визначено основні функціональні вимоги до прикладного програмного забезпечення створюваної ІТ. Публікації цих вимог у вигляді Use Case діаграм мови UML наведено у додатку В на рис. В.1-В. 11.

Для ліквідації дублюючих вимог було застосовано метод аналізу сценаріїв використання, розроблений А. Коберном. Результатом застосування цього методу є Use Case діаграма (див. рис. 3.9), яка описує варіант архітектури прикладного програмного забезпечення, що мінімізує дублювання окремих прецедентів створюваної ІТ. Було запропоновано розділити елементи цієї діаграми на окремі програмні модулі. Розподіл прецедентів Діаграми прецедентів, яку наведено на рис. 3.9, між програмними модулями прикладного програмного забезпечення ІТ оцінювання змін функціональних вимог до ІС, наведено у табл. 3.7. Схему розподілу цих модулів між виконавцями ІТ-проєкту та спринтами, що відведено на створення цих модулів, наведено на рис. 3.10.

Таким чином, в цьому розділі наведено усі основні платформонезалежні рішення з опису функціональної структури, інформаційного, алгоритмічного та програмного забезпечень ІТ оцінювання змін функціональних вимог до ІС, які визначають основні особливості будь-якої платформозалежної реалізації цієї технології.

4 ПРАКТИЧНЕ ВИКОРИСТАННЯ АНСАМБЛЕВОГО МЕТОДУ SIZE ORDERING RULE

4.1 Опис основних особливостей змінюваної функціональної задачі

Для практичної апробації розробленого ансамблевого методу Size Ordering Rule запропоновано використати проміжні результати ІТ-проєкту розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри». Цей проєкт було розроблено здобувачкою вищої освіти гр. УПГІТм-20-1 Кузьмою Є.А., а його результати опубліковано у [33], [34].

Метою даного ІТ-проєкту було створення функціональної задачі, яка дозволяла користувачу автоматизувати роботи з формування і наступних коригувань документу «Індивідуальний план науково-педагогічного працівника кафедри». При цьому кількість помилок та, відповідно, повторних ітерацій коригування вихідного документу повинно було зводитися до мінімуму. Функціональна задача «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» розглядалась як розвиток інформаційно-аналітичної системи «Університет», що експлуатується у Харківському національному університеті радіоелектроніки. Раніше в системі «Університет» було реалізовано функціональну задачу «Розподіл навчального навантаження між викладачами кафедри», основний вихідний документ якої є одним з розділів документу «Індивідуальний план науково-педагогічного працівника кафедри» [34].

Під час збору та аналізу функціональних вимог в даному ІТ-проєкті було встановлено відсутність дублювання функціональних вимог та визнано доцільним описати кожну з висунутих функціональних вимог як окрему роботу діаграми потоків даних.

Під час планування даного ІТ-проєкту було розроблено [33], [34]:

а) операційний опис архітектури функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»;

б) структурний опис архітектури функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри».

Операційний опис архітектури функціональної задачі представлено у вигляді функціональної структури, модель потоків даних якої наведена на рис. Г.1 додатку Г. Побудову цієї моделі було проведено з використанням діаграм потоків даних (Data Flow Diagram, DFD) засобами інтегрованого програмного пакету AllFusion Process Modeler [33].

Структурний опис архітектури функціональної задачі представлено у вигляді схеми логічної структури БД. Цю схему наведено на рис. Г.2 додатку Г. Побудову цієї схеми було проведено з використанням методології діаграм «сутність-зв'язок» ERD (Entity-Relationship Diagrams) засобами інтегрованого програмного пакету AllFusion ERwin Data Modeler [33].

Табличне представлення операційного опису архітектури функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» наведено у табл. Г.1 [34]. В табл. Г.1 як числові номери вказано номери (ідентифікатори) робіт і вхідних та вихідних потоків, які було сформовано CASE-засобом AllFusion Process Modeler в процесі створення діаграми потоків даних, показаної на рис. Г.1.

Табличне представлення структурного опису архітектури функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» наведено у табл. Г.2 [34]. В табл. Г.2 як числові номери вказано номери (ідентифікатори) сутностей, які були сформовані CASE-засобом AllFusion ERwin Data Modeler в процесі створення діаграми потоків даних, показаної на рис. Г.2.

Результати об'єднання операційного та структурного описів архітектури функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» наведено у табл. Г.3 [34]. Ці результати

визначають перелік ідентифікаторів сутностей з табл. Г.2, які описують елементи діаграми потоків даних, ідентифікатори яких взято з табл. Г.1. У табл. Г.3 прийнято такі позначення:

а) «№ КЕ» – номер конфігураційного елемента (функції) функціональної задачі;

б) «Ідентифікатор підмножини» - номер, який позначає тип елемента діаграми потоків даних (1 – робота, 2 – вхідний потік, 3 – вихідний потік);

в) «№ елемента» – ідентифікатори елемента діаграми потоків даних, які взято з табл. Г.1;

г) «Множина ідентифікаторів сутностей, які описують елемент» – множина номерів, які позначають сутності схеми даних і наведені у табл. Г.2.

4.2 Опис запитів на зміни функцій задачі

В процесі розробки видів забезпечень функціональної задачі було запропоновано два RFC функціональних вимог, наведених на рис. Г.1 як окремі роботи діаграми потоків даних. На початку визначення ці RFC було сформульовано у вигляді текстових публікацій таких потреб:

а) RFC 1: «Видалити з функціональної задачі усі функції, які пов'язані із КРІ»;

б) RFC 2: «Додати у функціональну задачу функції, які дозволять автоматизувати облік і контроль виконання умов індивідуального контракту викладача кафедри».

Але такі текстові публікації RFC функціональних вимог є надто узагальненими і не відповідають умовам існування добрих функціональних вимог до функціональної задачі. Тому під час виконання роботи «Визначте/уточніть потребу в змінах» текстовий опис RFC було уточнено. В

процесі уточнення RFC 1 було проведено пошук робіт діаграми потоків даних, в назвах яких зустрічається аббревіатура «КРІ». В процесі уточнення RFC 1 учасниками ІТ-проекту було прийнято рішення розділити цю публікацію на дві окремі публікації дочірніх RFC, які відповідають умовам існування добрих функціональних вимог. Результати уточнення наведено у табл. 4.1.

Таблиця 4.1 – Уточнені текстові публікації запитів на зміну функціональних вимог задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»

№ батьківського RFC	№ дочірнього RFC	Ідентифікатор RFC	Текстова публікація RFC
1	1	1	«Видалити з функціональної задачі функцію 7 «Формування та ведення НДІ по ключовим КРІ»
1	2	2	«Видалити з функціональної задачі функцію 8 «Формування КРІ викладача та частки від КРІ кафедри»
2	1	3	«Додати у функціональну задачу функцію 11 «Облік умов індивідуального контракту викладача кафедри»
2	2	4	«Додати у функціональну задачу функцію 12 «Контроль виконання умов індивідуального контракту викладача кафедри»

Наведені у табл. 4.1 текстові публікації RFC функціональних вимог до функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» було визнано учасниками ІТ-проєкту доцільними для подальшої реалізації та передано для вирішення задачі оцінювання обсягу робіт з їх реалізації. Як експертів, призначених для вирішення задачі оцінювання визначених RFC, було визначено учасників ІТ-проєкту, перелік посад яких наведено у табл. 4.2.

Таблиця 4.2 – Перелік посад учасників вирішення задачі оцінювання

№ з/п	Належність	Посада
1	Розробник	Аналітик
2	Розробник	Системний архітектор
3	Розробник	Програміст
4	Замовник	Викладач (професор)
5	Замовник	Викладач (асистент)

4.3 Хід та результати оцінювання запропонованих запитів на зміни функціональних вимог до задачі

В цьому підрозділі наведено опис особливостей та основних результатів вирішення задачі оцінювання RFC функціональних вимог до задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри», публікації яких наведено у табл. 4.1, з використанням розроблених ансамблевого методу Size Ordering Rule та ІТ оцінювання змін функціональних вимог до ІС.

Під час виконання стадії 1 оцінювання в процесі виконання етапу 1 розробленого методу було виконано крок «Формування категорій оцінювання

«Big», «Small», «Uncertain»». Результатом виконання етапу 1 є відеограма у вигляді табл. 4.3 (див. табл. 4.3).

Таблиця 4.3 – Сформовані категорії оцінювання

Категорія «Big»	Категорія «Small»	Категорія «Uncertain»
-	-	-

В процесі виконання етапу 2 розробленого методу експертами, визначеними у табл. 4.2, було під час обговорення визначено, що для розділення категорій «Big» та «Small» буде використовуватися показник «Кількість спринтів, які виділяються на реалізацію RFC», значення якого в даному випадку буде дорівнювати одному спринту (тривалість спринта встановлено у два календарні тижні). RFC, які для реалізації потребують менше одного спринта, слід відносити до категорії «Small», а RFC, які для реалізації потребують більше одного спринта, слід відносити до категорії «Big».

В процесі виконання етапу 3 розробленого методу було сформовано та ідентифіковано картки, що мають вигляд, приклад якого наведений на рис. 4.1.

Id: 00001	Проект «Розробка функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»»
RFC	«Видалити з функціональної задачі функцію «Формування та ведення НДІ по ключовим КРІ»
Функціональна вимога	Функція «Формування і ведення НДІ по ключовим КРІ»
Id елемента візуальної моделі	7

Рисунок 4.1 – Приклад ідентифікованої картки з описом запиту на зміну та змінюваної вимоги

В процесі виконання етапу 4 розробленого методу ідентифіковані картки чотирьох RFC, визначених у табл. 4.1, було розподілено випадковим чином за категоріями «Big», «Small» та «Uncertain». Результатом виконання етапу 4 став такий розподіл:

- а) до категорії «Small» було віднесено RFC 2;
- б) до категорії «Big» було віднесено RFC 1 та RFC 4;
- в) до категорії «Uncertain» було віднесено RFC 3.

Приклад фрагменту протоколу виконання етапу 5 стадії 1 вирішення задачі оцінювання визнаних RFC функціональних вимог до задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» наведено у табл. 4.4.

У табл. 4.4 прийнято такі позначення: категорія «Small» позначена ідентифікатором 1; категорія «Big» позначена ідентифікатором 2; категорія «Uncertain» позначена ідентифікатором 3; атрибут «Part_Solving_Task» приймає значення 1 у тому випадку, якщо експерт робив свій хід, а не відмовлявся від нього; атрибут «Proposal_Discussion» приймає значення 1 в тому випадку, коли експерт замість ходу пропонував обговорити ситуацію з ідентифікованою картою.

З табл. 4.4 виходить, що перший хід зробив аналітик, який переніс ідентифіковану картку з RFC 1 з категорії «Big» до категорії «Small». На другому ході системний архітектор переніс ідентифіковану картку з RFC 3 з категорії «Uncertain» до категорії «Big». На третьому ході програміст пропустив свій хід. На четвертому ході викладач (професор) переніс ідентифіковану картку з RFC 3 з категорії «Small» до категорії «Big». На п'ятому ході викладач (асистент) переніс ідентифіковану картку з RFC 4 з категорії «Small» до категорії «Big». На шостому ході аналітик зробив зворотнє перенесення ідентифікованої картки з RFC 4 з категорії «Big» до категорії «Small». На сьомому ході системний архітектор запропонував обговорити оцінку ідентифікованої картки з RFC 4. За результатом обговорення було погоджено рішення залишити цю ідентифіковану картку у категорії «Small», після чого усі експерти по черзі пропустили свої ходи для завершення виконання етапу 5 розробленого методу.

Таблиця 4.4 – Протокол виконання функції 1.3 інформаційної технології оцінювання змін функціональних вимог до задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»

Id_Project	Id_Task	Id_Stage	Id_Card	Id_Expert	Id_Proposed_Catagory	Part_Solving_Task	Suggested_Scale_Value	Proposal_Discussion
1	1	1	1	1	1	1	-	-
1	1	1	3	2	2	1	-	-
1	1	1	3	3	2	-	-	-
1	1	1	3	4	1	1	-	-
1	1	1	4	5	1	1	-	-
1	1	1	4	1	2	1	-	-
1	1	1	4	2	2	-	-	1
1	1	1	4	3	2	-	-	-
1	1	1	4	4	2	-	-	-
1	1	1	4	5	2	-	-	-
1	1	1	4	1	2	-	-	-
1	1	1	4	2	2	-	-	-

Етап 6 розробленого методу не виконувався, оскільки категорія «Uncertain» за результатами виконання етапу 5 залишилася пустою.

Для виконання етапу 7 спочатку було обрано категорію «Small», в якій знаходилися три ідентифіковані картки з RFC 1, RFC 2 та RFC 3. Для цієї категорії було сформовано шкалу, найменше значення якої дорівнює 1 календарному дню, а найбільше значення дорівнює 14 календарним дням (одному спринту), як виходить зі значення показника.

В процесі виконання етапу 8 розробленого методу за сформованою шкалою відбувався випадковий розподіл ідентифікованих карток таким чином:

- а) ідентифіковану картку з RFC 1 розподілено на відмітку «2 календарних дні»;
- б) ідентифіковану картку з RFC 2 розподілено на відмітку «7 календарних днів»;
- в) ідентифіковану картку з RFC 3 розподілено на відмітку «12 календарних днів».

Протокол виконання етапу 9 розробленого методу на стадії 1 наведено у табл. 4.5. На першому ході аналітик переносить ідентифіковану картку з RFC 1 з відмітки «2 календарних дні» на відмітку «1 календарний день». На другому ході системний архітектор пропускає свій хід. На третьому ході програміст переносить ідентифіковану картку з RFC 2 з відмітки «7 календарних днів» на відмітку «1 календарний день». На четвертому ході викладач (професор) переносить ідентифіковану картку з RFC 3 з відмітки «12 календарних днів» на відмітку «1 календарний день». На п'ятому ході викладач (асистент) пропускає свій хід. Далі інші чотири експерта теж пропускають свої ходи, на чому виконання етапу 9 закінчується.

В процесі виконання етапу 10 розробленого методу було побудовано шкалу, мінімальним значенням якої було 14 календарних днів, а максимальним – 56 календарних днів (4 спринти). В результаті ходів було визнано, що за оцінкою експертів для реалізації ідентифікованої картки з RFC 4 потрібно 21 календарний день (1,5 спринта).

Таблиця 4.5 – Протокол виконання функції 1.4 інформаційної технології оцінювання змін функціональних вимог до задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»

Id_Project	Id_Task	Id_Stage	Id_Card	Id_Expert	Id_Proposed_Cathegory	Part_Solving_Task	Suggested_Scale_Value	Proposal_Discussion
1	1	1	1	1	1	1	1	-
1	1	1	1	2	1	-	-	-
1	1	1	2	3	2	1	1	-
1	1	1	3	4	1	1	1	-
1	1	1	3	5	1	-	-	-
1	1	1	3	1	1	-	-	-
1	1	1	3	2	1	-	-	-
1	1	1	3	3	1	-	-	-
1	1	1	3	4	1	-	-	-
1	1	1	3	5	1	-	-	-

На стадії 2 вирішення задачі оцінювання відбувається оцінювання впливу, який зазначені RFC можуть здійснювати на інші актуальні функціональні вимоги, які було висунуто до окремих функцій задачі. Під час виконання стадії 1 оцінювання в процесі виконання етапу 1 розробленого методу було виконано крок «Формування категорій оцінювання «Big», «Small», «Uncertain»». В процесі виконання етапу 2 розробленого методу на стадії 2 експертами, визначеними у табл. 4.2, було прийнято рішення використовувати той самий показник і з тим самим значенням, що й під час виконання аналогічного етапу на стадії 1. В процесі виконання етапу 3 розробленого методу було сформовано та ідентифіковано картки, що мають вигляд, приклад якого наведений на рис. 4.2. Під час формування та ідентифікації карток функціональні вимоги обиралися за співпадінням ідентифікаторів сутностей, які беруть участь у описах елементів діаграми потоків даних, які відповідають публікаціям цих вимог (роботи, вхідні та вихідні потоки). Для прикладу, наведеного на рис. 4.2, таким ідентифікатором є 12 – ідентифікатор сутності «КРІ» (див. табл. Г.2). При цьому з розгляду під час пошуку виключалися функціональні вимоги, до яких безпосередньо висувалися визначені RFC. Для прикладу, наведеного на рис. 4.2, такою вимогою є вимога до функції «Формування і ведення НДІ по ключовим КРІ», яка на діаграмі потоків даних позначена ідентифікатором 7.

У табл. 4.6 наведено перелік RFC, перелік ідентифікаторів та найменування робіт діаграми потоків даних, що описують функції, на які впливає відповідний RFC.

В процесі виконання етапу 4 розробленого методу ідентифіковані картки чотирьох RFC та вимог, на які вони впливають, було розподілено випадковим чином за категоріями «Big», «Small» та «Uncertain»». Результатом виконання етапу 4 став такий розподіл:

- а) до категорії «Small» було віднесено RFC 1;
- б) до категорії «Big» було віднесено RFC 3;
- в) до категорії «Uncertain» було віднесено RFC 2 та RFC 4.

Id: 00001	Проект «Розробка функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри»»
RFC	«Видалити з функціональної задачі функцію «Формування та ведення НДІ по ключовим КРІ»
Функціональна вимога	Функція «Формування КРІ викладача та частки від КРІ кафедри»
Id елемента візуальної моделі	8

Рисунок 4.2 – Приклад ідентифікованої картки з описом запиту на зміну та функціональних вимог, на які впливає цей запит

Таблиця 4.6 – Відповідність RFC та функціональних вимог, на які вони впливають

RFC	Id елемента	Найменування роботи даних
RFC 1	8	«Формування КРІ викладача та частки від КРІ кафедри»
RFC 2	2	«Формування розділу «Наукова робота»»
	9	«Формування зведеної таблиці на навчальний рік»
	10	«Формування вихідного документу «ІП»»
RFC 3	2	«Формування розділу «Наукова робота»»
	3	«Формування розділу «Методична робота»»
	4	«Формування розділу «Організаційна робота»»
RFC 4	2	«Формування розділу «Наукова робота»»
	3	«Формування розділу «Методична робота»»
	4	«Формування розділу «Організаційна робота»»

В процесі виконання Етапу 5 стадії 2 вирішення задачі оцінювання визнаних RFC функціональних вимог до задачі «Формування та ведення

індивідуального плану науково-педагогічного працівника кафедри» експертами були прийняті такі рішення:

- а) RFC 1 залишено у категорії «Small»;
- б) RFC 2 перенесено з категорії «Uncertain» до категорії «Small»;
- в) RFC 3 залишено у категорії «Big»;
- г) RFC 4 перенесено з категорії «Uncertain» до категорії «Big».

Протоколювання ходів експертів під час виконання етапу 5 стадії 2 здійснювалося таким самим способом, що й під час виконання етапу 5 стадії 1 (аналогічно табл. 4.4).

Етап 6 розробленого методу на стадії 2 також не виконується, оскільки категорія «Uncertain» за результатами виконання етапу 5 стадії 2 залишилася пустою.

Для виконання етапу 7 стадії 2 спочатку також було обрано категорію «Small», в якій знаходяться дві ідентифіковані картки з RFC 1 та RFC 2. Для цієї категорії було сформовано шкалу, найменше значення якої дорівнює 1 календарному дню, а найбільше значення дорівнює 14 календарним дням (одному спринту), як виходить зі значення показника.

В процесі виконання етапу 8 розробленого методу за цією шкалою відбувається випадковий розподіл ідентифікованих карток таким чином:

- а) ідентифіковану картку з RFC 1 розподілено на відмітку «4 календарних дні»;
- б) ідентифіковану картку з RFC 2 розподілено на відмітку «3 календарних дні».

В процесі виконання етапу 9 розробленого методу на стадії 2 вирішення задачі оцінювання визнаних RFC функціональних вимог до задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри» експертами були прийняті такі рішення:

- а) ідентифіковану картку з RFC 1 перенесено з відмітки «4 календарних дні» до відмітки «1 календарний день»;

б) ідентифіковану картку з RFC 2 перенесено з відмітки «3 календарних дні» до відмітки «1 календарний день».

Протоколювання ходів експертів під час виконання етапу 9 стадії 2 здійснювалося таким самим способом, що й під час виконання етапу 9 стадії 1 (аналогічно табл. 4.5).

В процесі виконання етапу 10 розробленого методу на с2 було побудовано шкалу, мінімальним значенням якої було 14 календарних днів, а максимальним – 56 календарних днів (4 спринти). За цією шкалою відбувся випадковий розподіл ідентифікованих карток таким чином:

а) ідентифіковану картку з RFC 3 розподілено на відмітку «24 календарних дні»;

б) ідентифіковану картку з RFC 4 розподілено на відмітку «43 календарних дні».

В результаті здійснення ходів експертами було прийнято такі рішення:

а) ідентифіковану картку з RFC 3 перенесено з відмітки «24 календарних дні» до відмітки «16 календарних днів»;

б) ідентифіковану картку з RFC 4 перенесено з відмітки «43 календарних дні» до відмітки «16 календарних днів».

Під час формування звіту з результатами оцінювання було отримано такий вихідний документ (див. рис. 4.3).

За результатами інтуїтивного оцінювання чотирьох змін, які було висунуто до функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри», було встановлено, що на реалізацію висунутих RFC вимагається:

а) RFC 1 – 2 календарних дні;

б) RFC 2 – 2 календарних дні;

в) RFC 3 – 17 календарних днів;

г) RFC 4 – 37 календарних днів.

RFC	Вимога, до якої висунуто RFC	Витрати часу	Вимоги, на які впливає RFC	Витрати часу	Підсумок
1	2	3	4	5	6
RFC 1	«Формування та ведення НДІ по ключовим КРІ»	1	«Формування КРІ викладача та частки від КРІ кафедри»	1	2
RFC 2	«Формування КРІ викладача та частки від КРІ кафедри»	1	«Формування розділу «Наукова робота»»	1	2
			«Формування зведеної таблиці на навчальний рік»		
			«Формування вихідного документу «ІІ»»		
RFC 3	«Облік умов індивідуального контракту викладача кафедри»	1	«Формування розділу «Наукова робота»»	16	17

Рисунок 4.3 – Вихідний документ «Звіт з результатами оцінювання»

1	2	3	4	5	6
RFC 3	«Облік умов індивідуального контракту викладача кафедри»		«Формування розділу «Методична робота»»		
			«Формування розділу «Організаційна робота»»		
RFC 4	«Контроль виконання умов індивідуального контракту викладача кафедри»	21	«Формування розділу «Наукова робота»»	16	37
			«Формування розділу «Методична робота»»		
			«Формування розділу «Організаційна робота»»		

Рисунок 4.3, аркуш 2

4.4 Висновки до четвертого розділу

Під час виконання четвертого розділу кваліфікаційної роботи було визначено ІТ-проект, за даними якого необхідно провести апробацію результатів магістерської роботи. Як такий ІТ-проект було обрано ІТ-проект розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри». Було визначено головні особливості цього ІТ-проекту і, зокрема, описи архітектури функціональної задачі, побудовані на основі функціональних вимог, які висувалися до окремих функцій цієї задачі.

Було сформульовано RFC функціональних вимог, які висувалися до окремих функцій функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри». Встановлено, що прості текстові публікації цих RFC не вдовольняють вимогам, які висуваються до добре сформульованих вимог. Тому ці RFC було уточнено, в результаті чого було отримано чотири RFC, які відповідають вимогам, що висуваються до добре сформульованих вимог. Публікації цих RFC наведено у табл. 4.1

Для цих чотирьох RFC було вирішено задачу інтуїтивного оцінювання з використанням розроблених ансамблевого методу Size Ordering Rule та ІТ оцінювання змін функціональних вимог до ІС. В результаті вирішення цієї задачі було отримано такі результати (див. рис. 4.3):

- а) тривалість виконання RFC 1 – 2 календарних дні;
- б) тривалість виконання RFC 2 – 2 календарних дні;
- в) тривалість виконання RFC 3 – 17 календарних днів;
- г) тривалість виконання RFC 4 – 37 календарних днів.

ВИСНОВКИ

В магістерській кваліфікаційній роботі вирішено задачу дослідження інтуїтивних методів оцінювання змін функціональних вимог у ІТ-проєкті. Під час вирішення цієї задачі було отримано такі результати.

Проведено аналіз актуального фреймворку управління змінами, встановлено його основні переваги та недоліки. Визнано, що цей фреймворк не дає жодних рекомендацій з визначення особливостей роботи «Окресліть масштаби змін» на подальших стадіях життєвого циклу проєкту. В результаті аналізу систем класифікації RFC встановлено, що єдиної системи класифікації таких запитів не існує, але практично усі існуючі системи класифікації RFC виділяють як окремий клас RFC функціональних вимог. На основі аналізу сучасних досліджень в галузі оцінювання RFC функціональних вимог було встановлено, що зараз вирішенню цього питання не приділяється достатньо уваги, використання засобів штучного інтелекту не дає бажаних наслідків, а застосування формальних описів RFC функціональних вимог стає комерційно вигідним тільки у випадку управління змінами у великих ІС. Виходячи з цих результатів аналізу, було сформульовано мету і основні задачі магістерської кваліфікаційної роботи.

Було визначено основні особливості процесу інтуїтивного оцінювання та розглянуто основні методи інтуїтивного оцінювання, які рекомендується застосовувати у ІТ-проєктах. Проведено порівняльний аналіз цих методів за чотирма обраними критеріями, за результатами якого для вирішення задачі інтуїтивного оцінювання RFC функціональних вимог до ІС рекомендовано обрати для подальшого вдосконалення методи Bucket System та Big/Small/Uncertain за умови, що вдасться поліпшити точність їхнього оцінювання. Для цього поліпшення було запропоновано використати методи Dot-voting, Planning Poker або Ordering Rule. Як основну концепцію

вдосконалення було запропоновано використати напрям «Суміш експертів» концепції ансамблювання. В результаті вдосконалення було розроблено ансамблевий метод Size Ordering Rule та встановлено джерела вдосконалення кожного з етапів цього методу.

Базуючись на результатах розробки ансамблевого методу Size Ordering Rule, визначено основні вимоги до ІТ оцінювання змін функціональних вимог до ІС, до функцій та до видів забезпечень цієї ІТ. Розроблено концептуальну і логічну схеми БД цієї ІТ. Розроблено узагальнений алгоритм виконання роботи «Окресліть масштаби змін» з використанням ІТ оцінювання змін функціональних вимог до ІС та схеми алгоритмів виконання окремих кроків цього узагальненого алгоритму. Визначено основні функціональні вимоги до прикладного програмного забезпечення створюваної ІТ та вирішено задачу усунення дублюючих прецедентів цих вимог з використанням методу аналізу сценаріїв використання. Запропоновано схему розподілу прецедентів програмного забезпечення ІТ між окремими модулями та схему розподілу цих модулів між виконавцями ІТ-проєкту та спринтами, що відведено на створення цих модулів.

Для практичної апробації отриманих результатів було обрано ІТ-проєкт розробки функціональної задачі «Формування та ведення індивідуального плану науково-педагогічного працівника кафедри». Було визначено описи архітектури цієї задачі та публікації чотирьох RFC, висунутих до окремих функціональних вимог. Описано хід вирішення задачі інтуїтивного оцінювання з використанням розроблених методу та ІТ та наведено отримані результати оцінювання.

Отримані результати можуть бути використані для створення спеціалізованих ІС та ІТ управління ІТ-проєктами розробки та супроводження ІС управління підприємствами та організаціями.

Результати магістерської кваліфікаційної роботи опубліковано у [27].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення магістерської кваліфікаційної роботи за спеціальністю 122 Комп'ютерні науки (освітня програма «Управління проєктами 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, 2021. 28 с.

2. Настанова до Зводу знань з управління проєктами. Настанова РМВОК. Сьоме видання. Стандарт з управління проєктами. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 2021. 370 с.

3. Managing change in organizations: a practice guide. Project Management Institute, Inc., 14 Campus Boulevard Newtown Square, Pennsylvania 19073-3299 USA, 2013. 122 p.

4. ISO/IEC/IEEE 12207:2017. Systems and software engineering – Software life cycle processes. Чинний від 2017-11-01. 150 p.

5. Євланов М.В., Юр'єв І.О., Крук Б.Є. Розробка методу вирішення задачі формування і класифікації запитів на зміни ІТ-продукту. *АСУ та прилади автоматики*. 2024. Вип. 182. С. 5-25. DOI: 10.30837/0135-1710.2024.182.005

6. Віжан К. Процес управління змінами в ІТ-компанії, або Як побороти страх роботи з чендж реквестами. *DOU*. URL: <https://dou.ua/forums/topic/42501/> (дата звернення: 27.03.2024 р.).

7. Riesener M, Dölle C, Mendl-Heinisch M, Schuh G, Keuper A. Derivation of Description Features for Engineering Change Request by Aid of Latent Dirichlet Allocation. *Proceedings of the Design Society: DESIGN Conference*. 2020. 1. P. 697-706. DOI: 10.1017/dsd.2020.98

8. Jarratt T.A.W. et al. Engineering change. An overview and perspective on the literature. *Research in Engineering Design*. 2011. Vol. 22, No. 2. P. 103-124. DOI: 10.1007/s00163-010-0097-y
9. Feldhusen J., Grote K.-H. Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung, 8., vollständig überarbeitete Auflage, Springer Vieweg, Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-642-29569-0
10. Sakhrawi Z., Sellami A., Bouassida N. Requirements Change Requests Classification: An Ontology-Based Approach. *Intelligent Systems Design and Applications. ISDA 2019. Advances in Intelligent Systems and Computing*. Springer, Cham. 2021. Vol. 1181. DOI: 10.1007/978-3-030-49342-4_47
11. Zhou Y., Su Y., Chen T., Huang Z., Gall H.C., Panichella S. User Review-Based Change File Localization for Mobile Applications. *IEEE Transactions on Software Engineering*. 2019. Vol.47. P. 2755-2770.
12. Panichella S., Zaugg N. An Empirical Investigation of Relevant Changes and Automation Needs in Modern Code Review. *Empiric Software Engineering*. 2020. Vol. 25. P. 4833–4872. DOI: 10.1007/s10664-020-09870-3
13. Beller M, Bacchelli A, Zaidman A, Jurgens E. Modern code reviews in open-source projects: which problems do they fix? *11Th working conference on mining software repositories, MSR 2014, proceedings*. Hyderabad, India. 2014. P. 202–211.
14. Meneely, A., Corcoran, M., Williams, L. Improving developer activity metrics with issue tracking annotations. *WETSoM '10: Proceedings of the 2010 ICSE Workshop on Emerging Trends in Software Metrics*. P. 75–80. DOI: 10.1145/1809223.1809234
15. Gethers, M., Dit, B., Kagdi, H., Poshyvanyk, D. Integrated impact analysis for managing software changes. *34th International Conference on Software Engineering (ICSE)*. Zurich, Switzerland. 2012. P. 430-440. DOI: 10.1109/ICSE.2012.6227172
16. Manapian, A., Prompoon, N. Software time estimation model for requirements change based on software prototype profiles using an analogy estimation

method. *2014 International Computer Science and Engineering Conference (ICSEC)*. Khon Kaen. Thailand. 2014. P. 366-371. DOI: 10.1109/ICSEC.2014.6978224

17. Miguel, M.A., Araújo, M.A.P., David, J.M.N., Braga, R. A framework to support effort estimation on software maintenance and evolution activities. *12th Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era, Proceedings*. Florianopolis - Santa Catarina. 2016. P. 232-239.

18. Rostami, K., Stammel, J., Heinrich, R., Reussner, R. Change Impact Analysis by Architecture-based Assessment and Planning. *Lecture Notes in Informatics, Proceedings - Series of the Gesellschaft fur Informatik*. Hannover. 2017. Vol. P-267, P. 69-70.

19. Rostami, K., Heinrich, R., Busch, A., Reussner, R. Architecture-Based Change Impact Analysis in Information Systems and Business Processes. *2017 IEEE International Conference on Software Architecture (ICSA)*. Gothenburg, Sweden. 2017. P. 179-188. DOI: 10.1109/ICSA.2017.17

20. Chari, K., Agrawal, M. Impact of incorrect and new requirements on waterfall software project outcomes. *Empirical Software Engineering*. 2018. Vol. 23, Iss. 1. P. 165-185. DOI: 10.1007/s10664-017-9506-4

21. Yimer, S. T., Molla, Y. S., Alemneh, E. Predicting Software Maintenance Type, Change Impact, and Maintenance Time Using Machine Learning Algorithms. *2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*. Bahir Dar, Ethiopia. 2022. P. 37-41. DOI: 10.1109/ICT4DA56482.2022.9971350

22. Dugar, M. How Machine Learning Can Help Developers. *24th International Arab Conference on Information Technology, ACIT 2023*. Ajman. 2023. Code 198200. DOI: 10.1109/ACIT58888.2023.10453880

23. Amaral, A., de Franco Rosa, F. Method for Assessing the Potential Impact of Changes in Software Requirements of Agile Methodologies Based Projects. *Human Interface and the Management of Information. HCII 2023. Lecture Notes in Computer*

Science. Springer, Cham. 2023. Vol. 14016. P. 3-21. DOI: 10.1007/978-3-031-35129-7_1

24. Менеджмент в IT. *Doitsmartly*. 2021. URL: <https://doitsmartly.ru/all-articles/management/99-agile/115-agile-estimation-techniques.html> (дата звернення: 25.10.2021).

25. Князька Н. Оцінка тестових завдань. Техніки і як їх використовувати. *DOU*. URL: <https://dou.ua/lenta/columns/estimation-of-testing-tasks/> (дата звернення: 25.10.2024).

26. Planning (Scrum) Poker. *QATestLab*. URL: <https://training.qatestlab.com/blog/technical-articles/planning-poker/> (дата звернення: 25.10.2024).

27. Опара В., Шквира В. Estimates or Guesstimates? Обираємо метод оцінювання задач. *DOU*. URL: <https://dou.ua/lenta/articles/estimates-or-guesstimates/> (дата звернення: 25.10.2024).

28. Євланов М.В., Міщенко О.А. Вдосконалення методу інтуїтивного оцінювання запитів на зміни функціональних вимог до інформаційної системи. *LII International scientific and practical conference «Scientific Research in the Age of Virtual Reality: Exploring New Frontiers» (December 18-20, 2024)*. Montreal, Canada: International Scientific Unity. 2024. P. 111-116.

29. Zhou Zhi-hua. Ensemble Methods. Foundations and Algorithms. Boca Raton, London, New York: CRC Press, Taylor & Francis Group. 2012. 220 p.

30. Левикін В.М., Євланов М.В., Керносов М.А. Патерни проєктування вимог до інформаційних систем: моделювання та використання: монографія. Харків: ООО «Компанія СМІТ», 2014. 320 с.

31. Як вибрати правильний технологічний стек для вашого проєкту *REDSTONE*. URL: <https://redstone.agency/blog/yak-vybraty-pravylyniy-tekhnolohichnyi-stek-dlia-vashoho-proektu/> (дата звернення: 29.12.2024).

32. Cockburn A. Writing Effective Use Cases. Addison-Wesley Professional. 2000. 458 p.

33. Кузьма Є. А. Дослідження моделей і методів аналізу конфігурації ІТ-продукту : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 122 Комп'ютерні науки / Є. А. Кузьма ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків, 2021. 107 с.

34. Ievlanov M., Vasilcova N., Neumyvakina O., Panforova I. Development of a method for solving the problem of IT product configuration analysis. *Eastern-European Journal of Enterprise Technologies*. 2022. Vol. 6. № 2. P. 6-19. DOI: 10.15587/1729-4061.2022.269133