Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)
Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

__Порівняльне дослідження методів збирання вимог для підвищення якості___
_____сформульованих вимог ІТ-проєкту_____
(тема)

Виконала:
студентка _2_ курсу, групи __УПГІТм-22-2__
_____Анна СТЕПЧЕНКО_____
(власне ім'я, прізвище)

Спеціальність____122 Комп'ютерні науки_____
(код і повна назва спеціальності)
Тип програми _____освітньо-наукова_____

(освітньо-професійна або освітньо-наукова)
Освітня програма Управління проектами в галузі інформаційних технологій_____
(повна назва освітньої програми)

Керівник проф. каф. ІУС, Ірина ПАНФЬОРОВА
(посада, власне ім'я, прізвище)

Допускається до захисту
Зав. кафедри _____ Костянтин_ПЕТРОВ
(підпис) (власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук_____
Кафедра_____Інформаційних управляючих систем_____
Рівень вищої освіти_____другий (магістерський)_____
Спеціальність_____122 Комп'ютерні науки_____
(код і повна назва)
Тип програми _____освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)
Освітня програма_Управління проектами в галузі інформаційних технологій_
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« 01 » квітня 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Степченко Анні Олексіївні_____
(прізвище, ім'я, по батькові)

1. Тема роботи Порівняльне дослідження методів збирання вимог для підвищення якості сформульованих вимог ІТ-проєкту
затверджена наказом університету від 01 квітня 2024 р. № 258Cm
2. Термін подання студентом роботи до екзаменаційної комісії 06.06.2024 р.
3. Вихідні дані до роботи: науково-технічні публікації; джерела інтернету; науково-технічна література, що стосуються теми кваліфікаційної роботи; результати дослідження ІТ-проєкту роботизації проведення банківської виписки.
4. Перелік питань, що потрібно опрацювати в роботі: аналіз існуючих методів збору вимог, постановка задачі вибору методів збору вимог, розробка методології вибору методів збору вимог, експериментальне дослідження використання методології вибору методів збору вимог.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз існуючих методів збору вимог та їх особливостей	01.04.2024	виконано
2	Обґрунтування мети кваліфікаційної роботи та постановка задачі	15.04.2024	виконано
3	Розробка класифікації методів збору вимог та розробка функції вигоди при виборі цих методів	20.04.2024	виконано
4	Розробка методології застосування функції вигоди, розробка матриці прийняття рішень для вирішення задачі вибору методів збору вимог	02.05.2024	виконано
5	Експериментальне дослідження ефективності розробленої методології на існуючому ІТ-проєкті роботизації проведення банківської виписки	05.05.2024	виконано
6	Оформлення пояснювальної записки та графічного матеріалу	24.05.2024	виконано
7	Підготовка презентації	28.05.2024	виконано
10	Попередній захист роботи	06.06.2024	виконано
13	Захист кваліфікаційної роботи	10.06.2024	виконано

Дата видачі завдання _01_ __квітня___2024 р.

Студент _____ _____
(підпис)

Керівник роботи _____ проф. каф. ІУС Ірина ПАНФЬОРОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 90 с., 4 рис., 23 табл., 35 джерел, 1 додаток.

ЗАДАЧА ВИБОРУ, ІНЖЕНЕРІЯ ВИМОГ, ІТ-ПРОЄКТ, КЛАСИФІКАЦІЯ, МАТРИЦЯ ПРИЙНЯТТЯ РІШЕНЬ, МЕТОДИ ЗБОРУ ВИМОГ, ФУНКЦІЯ ВИГОДИ.

Кваліфікаційна робота спрямована на дослідження існуючих методів збору вимог на ІТ-проєкті та розробки рішення по їх ефективному виборі.

Об’єктом дослідження кваліфікаційної роботи є найпоширеніших методів збору вимог.

Метою даної роботи є порівняльний аналіз методів збору вимог, аналіз їх спільних та відмінних рис, а також вирішення задачі вибору методів збору вимог, базуючись на об’єктивних критеріях. Практична значимість роботи полягає у розробці класифікації методів для забезпечення різноплановими вимогами проєкт, а також розробці функції вигоді при виборі методу із класу.

В результаті роботи була запропонована класифікація методів збору вимог, створені критерії, по яким можливо оцінити ці методи, розроблена функція вигоди, а також створена методологія використання цієї функції, в якій розроблена загальна матриця прийняття рішень. Результати роботи були експериментально перевірені на існуючому ІТ-проєкті.

Отримані результати можна використовувати для подальшої класифікації інших методів збору вимог, удосконалення функції вигоди і розробки додаткових критеріїв оцінки методів збору вимог для поліпшення фази виявлення вимог.

ABSTRACT

Explanatory note of the qualification work: 90 pp., 4 figures, 23 tables, 35 sources, 1 appendix.

CLASSIFICATION, DECISION MATRIX, IT PROJECT, PROFIT FUNCTION, REQUIREMENTS GATHERING METHODS, REQUIREMENTS ENGINEERING, SELECTION PROBLEM.

The qualification work is aimed at researching the existing methods of collecting requirements for an IT project and developing a solution for their effective selection.

The object of the qualification work research is 10 popular requirements gathering methods. The purpose of this work is a comparative analysis of requirements collection methods, analysis of their common and distinctive features, as well as solving the problem of choosing requirements collection methods based on objective criteria.

The practical significance of the work consists in the development of a classification of methods to meet the various requirements of the project, as well as the development of the benefit function when choosing a method from the class.

As a result of the work, a classification of requirements gathering methods was proposed, criteria were created by which it is possible to evaluate these methods, a benefit function was developed, and a methodology for using this function was created, in which a general decision-making matrix was developed. The results of the work were experimentally verified on the existing IT project.

The obtained results can be used for further classification of other requirements gathering methods, improvement of the benefit function and development of additional criteria for evaluating requirements gathering methods to improve the requirements discovery phase.

ЗМІСТ

С.

Скорочення та умовні познаки…… 8

Вступ……... 9

1 Аналіз предметної області та постановка задачі дослідження…… 10

1.1 Дослідження етапу збирання вимог в процесі інженерії вимог..…… 10

1.2 Дослідження існуючих методів збирання вимог в ІТ-проєктах…… 14

1.2.1 Інтерв'ю…… 16

1.2.2 Спостереження користувача …… 20

1.2.3 Прототипування……... 21

1.2.4 Аналіз документів…… 25

1.2.5 Семінари (воркшопи)……... 27

1.2.6 Анкетування…….. 28

1.2.7 Мозковий штурм ……... 30

1.2.8 Сценарний аналіз …… 32

1.2.9 Аналіз інтерфейсів ……... 34

1.2.10 Фокус-групи…… 36

1.3 Дослідження типів ІТ-проєктів …….. 37

1.4 Постановка задачі дослідження…….. 39

2 Постановка задачі вибору методів збору вимог…… 42

2.1 Задача класифікації методів збору вимог……... 42

2.2 Розробка критеріїв оцінки методів збору вимог……. 45

2.3 Розробка функції вигоди…….. 48

3 Розробка методології застосування функції вигоди при виборі методів збору вимог……. 52

3.1 Розробка матриці прийняття рішення вибору методів збору вимог…. 52
3.2 Схема заповнення матриці прийняття рішень…… 53
4 Експериментальне дослідження використання методології вибору методів збору вимог……... 55
4.1 Загальний опис IT-проєкта по роботизації проведення банківської виписки..…… 54
4.2 Аналіз вхідних даних на етапі збору вимог IT-проєкта проведення банківської виписки…….. 56
4.2.1 Аналіз результатів ефективності альтернативи А……. 57
4.2.2 Аналіз результатів ефективності альтернативи В……. 58
4.3 Аналіз результатів A/B тестування…… 61
Висновки…….. 64
Перелік джерел посилання…… 65
Додаток А Графічний матеріал……. 70

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІВ – Інженерія вимог

BA – Business Analyst

JAD – Joint application development

PM – Project Manager

QA – Quality Assurance

SME – Subject Matter Expert

ВСТУП

Процес розробки програмного забезпечення включає багато етапів життєвого циклу, серед яких збирання та аналіз вимог (виявлення вимог) є, ймовірно, найважливішим для успішності програмної системи. Виявлення вимог спрямоване на отримання інформації про вимоги та потреби користувачів. Зазвичай вибір методів виявлення вимог залежить від практик компанії або особистого досвіду розробників.

Добре відомо, що витрати на виправлення помилок у програмному забезпеченні експоненційно зростають на пізніших етапах розробки. Тому, якщо виявити відсутні або неправильні вимоги на якомога раніше, витрати на виправлення цих помилок значно зменшуються.

Опитування показали, що третина розпочатих проєктів так і не була завершена, а половина з них вдалася лише частково. Причиною такої невдачі є погане визначення вимог, точніше, відсутність участі користувача, неповнота вимог, двозначність вимог, нереалістичні очікування та нечіткі цілі [1].

Метою даної роботи є порівняльний аналіз методів збору вимог, аналіз їх спільних та відмінних рис, а також вирішення задачі вибору методів збору вимог, базуючись на об’єктивних критеріях.

Результати дослідження будуть корисні малим IT-компаніям, які ще не мають достатньо ресурсів для купівлі автоматизованих рішень для ефективного прогнозування використання того чи іншого метода збору вимог при реалізації проєкта. Це рішення має задіяти базові дані, які властиві будь-якому IT-проєкту та їх виявлення не потребує багато часу.

У світі конкуренції вкрай важливо мінімізувати помилки на ранніх етапах, і уніфіковане рішення щодо вибору методів збору вимог допоможе одразу усунути такі проблеми, як неповнота, неузгодженість вимог тощо.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Дослідження етапу збирання вимог в процесі інженерії вимог

Інженерія вимог (ІВ) – це систематичний і дисциплінований підхід до специфікації та управління вимогами з метою розуміння бажань і потреб зацікавлених сторін і мінімізації ризику створення системи, яка не відповідає бажанням і потребам [2]. Життєвий цикл ІВ представленийА на рисунку 1.1. Метою ІВ є забезпечення того, щоб команда проєкту розробила такий продукт, який задовольняє потреби користувачів.

Рисунок 1.1 – Життєвий цикл інженерії вимог

ІВ включає всі прийоми, методи і процедури, пов’язані з виявленням і управлінням потребами зацікавлених сторін проєкту: стейкхолдери, користувачі тощо. В свою чергу, виявлення вимог складається з їх збору та аналізу. Робота над збором вимог в процесі інженерії для ІТ-проєкту є дуже важливим етапом та може використовуватись в будь-яких методологіях реалізації ІТ-проєкту від класичних до гнучких.

ІВ у гнучкій розробці здійснюється протягом усього циклу розробки ітеративно на кожному спринті [3]. Нижче наведено деякі практики ІВ та їх цінність для реалізації ІТ-проєкту:

– безперервне планування: завчасне складання плану дій на кожному етапі ІВ важливе для виявлення змін у потребах клієнта та відповідної адаптації стратегій розробки продукту [4];

– прозора комунікація: для того, щоб використовувати гнучкий процес ефективно, зацікавлені сторони та члени команди повинні спілкуватися напряму [4];

– пріоритезація: для успішного виконання проєкту та уникнення зайвої роботи необхідно сформувати послідовність вимог замовника та пріоритезувати їх, що підкреслить активну участь команди при співпраці з клієнтом [5];

– прототипування: використання прототипу на базі отриманих вимог для оцінки відповідності прописаним вимогам та внесення змін у разі необхідності [5];

– управління змінами: несвоєчасне управління змінами може призвести до затримок у проєкті, невідповідності продукту очікуванням клієнта та збільшення витрат на виправлення помилок, тому вчасне додавання або видалення вимог та оновлення існуючих є критичним при реалізації ІТ-проєкту [5].

Вимога – гнучка та динамічна сутність, яка може змінюватись на кожному етапі життєвого циклу реалізації проєкту. У цьому циклі ІВ та всі її етапи

відіграють важливу роль, оскільки якість, успіх та задоволеність замовника проєкту залежить саме від правильності інтерпретації виявлених вимог [6]. Якість ІВ залежить не тільки від результатів першого етапу – збору та аналізу – тому в процесі дослідження будуть розглянуті всі складові ІВ: виявлення, документування, погодження та керування.

Збір вимог та їх аналіз дозволяє глибоко зрозуміти потреби та цілі бізнесу. Це мінімізує ризик непорозумінь і допомагає знайти рішення, яке вирішує фактичні проблеми, з якими стикається стейкхолдер. Чітко визначені вимоги допомагають зрозуміти конкретні функції та можливості, які повинна мати система. Це дає змогу ефективно планувати, оцінювати та контролювати обсяг проєкту під час реалізації, а також знижує ризик провалу проєкту та підвищує його конкурентоспроможність. Чіткі та вичерпні вимоги запобігають надмірним змінам у процесі розробки, які можуть спричинити затримки, збільшення витрат і зниження якості.

Прозоро задокументовані вимоги зменшують розрив між користувачами, розробниками та менеджерами, тобто між людьми з і без досвіду розробки програмного забезпечення. Добре прописана вимога допомагає зрозуміти, яку проблему потрібно розв’язувати за допомогою реалізації ІТ-проєкта.

Документування вимог є кроком, що забезпечує збереження вимог у певній формі, оскільки без цього можливе втрачання або неправильне тлумачення вимог. Важливість документації полягає в тому, що вона створює міцну основу для підтвердження вимог і виявлення конфліктів, що потребують узгодження. Для валідації вимог необхідно чітко задокументовані вимоги, які становлять основу для обговорення. Документація також слугує джерелом для визначення вимог, дозволяючи використовувати вимоги з попередніх версій і систем-попередників або повторно використовувати їх з інших проєктів. Крім того, документація підтримує керування вимогами, що зосереджується на таких аспектах, як планування, встановлення пріоритетів, моніторинг і керування

змінами, і без неї багато аспектів управління вимогами не можуть функціонувати належним чином [7].

Під час виявлення і документації вимог визначаються очікувані класи користувачів продукту та аналізується інформація, отримана від замовника, а також переноситься в той формат документації, який прийнятий на проєкті, будь-то PDD (process definition document), SRS (Software Requirements Specification) або просто User Story тощо.

Етап погодження вимог в процесі ІВ включає в себе узгодження інформації про вимоги до продукту між різними зацікавленими сторонами, такими як клієнти, менеджери проєкту та команда розробки. Під час цього процесу вирішуються питання про те, які вимоги є найкритичнішими, як вони будуть реалізовані та як вони впливають на інші аспекти проєкту. На цьому етапі створюється та пріоритезується беклог – список функцій, вимог або завдань, які потрібно реалізувати в процесі розробки проєкту. Цей етап має велику цінність, оскільки він дозволяє уникнути непорозумінь і конфліктів між зацікавленими сторонами, забезпечуючи узгодженість та зрозумілість щодо того, що буде реалізовано в кінцевому продукті.

Перевірка вимог для їх погодження в основному відбувається за двома принципами: верифікація та валідація. Верифікація спрямована на те, щоб перевірити, чи відображені вимоги вірно та повністю. Верифікація включає аналіз вимог на відповідність критеріям якості, а також перевірку їх на реалістичність та можливість реалізації. Наприклад, це може включати перевірку вимог на відсутність суперечностей або недоліків. Валідація вимог спрямована на перевірку того, чи відповідають вимоги реальним потребам та очікуванням зацікавлених сторін. Валідація включає перевірку того, чи задовольняють вимоги потребам користувачів та бізнесу, чи сприятимуть вони досягненню поставлених цілей та чи є вони прийнятними для зацікавлених сторін. Наприклад, це може

включати проведення сесій з участю користувачів для оцінки, наскільки вимоги відповідають їхнім потребам і очікуванням.

Останньою складовою інженерії вимог є управління змінами. Ця складова включає додаткові активності в процесі еволюції вимоги з часом на проєкті. Перелік активностей та їх опис наведений нижче:

– ідентифікація будь-яких змін у вимогах або умовах, які можуть виникнути під час розробки продукту;

– оцінка впливу змін на весь проєкт, включаючи його бюджет, графік, ресурси та функціональність продукту;

– визначення та прийняття рішення щодо того, як реагувати на зміни, тобто який сценарій використовувати (прийняття, відхилення, прийняття з обмеженнями та мінімізація ризиків);

– актуалізація будь-яких відповідних документів, таких як специфікації вимог, плани проєкту, графіки, щоб відображати зміни;

– інформування зацікавлених сторін про будь-які зміни в вимогах, їхні впливи та прийняті рішення;

– систематичне відстеження та контроль за змінами вимог протягом усього процесу розробки продукту [8].

1.2 Дослідження існуючих методів збирання вимог в IT-проєктах

Зараз багато продуктів є апаратно-програмними гібридами, що означає, що розробка стає дедалі складнішою. Наприклад, в одному із продуктів IoT (Internet of things), такий як «розумний дім», апаратна складова включає в себе різноманітні датчики (датчики руху, датчики температури, датчики вологості),

камери спостереження, електронні замки та інші пристрої, які забезпечують збір даних про стан оточуючого середовища та дії користувачів. Ці дані передаються на обчислювальний центр, де вони обробляються та аналізуються.

Програмна складова включає в себе програмне забезпечення, яке контролює та керує різними аспектами «розумного дому», такими як освітлення, опалення, кондиціювання повітря, безпека та інші системи. Користувачі можуть взаємодіяти з системою через мобільні додатки або голосові асистенти для встановлення режимів, моніторингу стану системи та здійснення управління віддалено [9].

Отже, лише зміна однієї вимоги – навмисно чи ні – може спровокувати ненавмисну ланцюгову реакцію, яка зрештою призведе до руйнування продукту. Розглядаючи зміну вимог або коду, людський мозок не в змозі врахувати всі можливі варіанти і наслідки, оскільки вони становлять складний набір взаємодій. Інструменти керування вимогами при ІВ дають змогу керувати всіма вимогами, змінами та створювати відстежуваність, яка підтримує ефективнішу розробку продукту. Загалом, чим швидше організація, яка створює складні продукти, зможе реалізувати переваги спрощеного процесу інженерії вимог, тим успішніше вона буде. Нажаль, у більшості випадків зацікавлені сторони не знають про всі альтернативи, які існують, коли йдеться про реалізацію конкретного проєкту. Крім того, немає універсального рішення. Жодна єдина техніка збору вимог не допоможе отримати повний набір вимог, який задовольнить потребам стейкхолдерів та витримає перевірку. Тому необхідно дослідити багатокритеріальний підхід до збору вимог. Передовий досвід розробки вимог рекомендує використовувати різні методи, спрямовані на різні фази процесу.

Існує велика кількість різних методів збору вимог, але в контексті даної кваліфікаційної роботи будуть розглянуті найбільш розповсюджені методи [10]:

– інтерв’ю;

– спостереження користувача;

– прототипування;
– аналіз документів;
– семінари (воркшопи);
– анкетування;
– мозковий штурм;
– сценарний аналіз;
– аналіз інтерфейсу;
– фокус-групи.

Наступні пункти присвячені детальному аналізу кожного з цих методів.

1.2.1 Інтерв’ю

Інтерв’ю один на один є найпоширенішим методом збору інформації про потреби, а також одним з основних джерел інформації про них. Щоб отримати максимальну користь від інтерв’ю, його слід добре продумати і підготувати перед зустріччю з респондентом. Інтерв’юер повинен визначити зацікавлені сторони, з якими потрібно провести інтерв’ю. Це можуть бути користувачі, які взаємодіють з поточною або новою системою, керівництво, фінансисти проєкту або будь-хто інший, хто буде залучений до системи.

Під час інтерв’ю інтерв’юер повинен отримати дозвіл від респондента на використання запису, щоб у разі упущення деталей під час зустрічі можна було легко їх знайти. Наприкінці інтерв’ю результати повинні бути надані респонденту для підтвердження його відповідей.

Інтерв’ю поділяють на групові та індивідуальні. Групові інтерв’ю схожі на індивідуальні, за винятком того, що в них беруть участь більше однієї людини.

Групові інтерв'ю добре працюють, коли співбесідники перебувають на одному рівні або на одній посаді. Групове інтерв'ю також має перевагу, коли є обмеження в часі. Можна згенерувати більше думок і дискусій, оскільки хтось у групі може висловити або запропонувати ідею, яку, можливо, не помітили інші, що, в свою чергу, може призвести до дискусії або надати більше інформації з певного питання. Інтерв'юер може оцінити, які питання є більш загальноприйнятими, а які відрізняються від інших. Великим недоліком може бути планування інтерв'ю. Коли в інтерв'ю беруть участь кілька осіб, може бути важко або довго визначити дату і час, які б влаштовували всі сторони, або ж це може зайняти багато часу [11].

Дослідження показало, що інтерв'ю розділене не тільки за кількістю учасників та форматом проведення, а також за типами.

Неструктуровані інтерв'ю, як правило, неформальні та мають розмовний тон. Питання не вирішуються заздалегідь, скоріше інтерв'юер досліджуватиме зацікавлену сторону, щоб отримати багату, глибоку інформацію. Інтерв'ю такого типу ефективні під час налагодження стосунків із зацікавленою стороною або під час обговорення делікатних тем, однак, для збору всієї необхідної інформації можуть знадобитися численні раунди інтерв'ю.

На відміну від неструктурованих, структуровані інтерв'ю проводяться за жорстким набором питань без можливості маневрувати. Наявність списку вичерпних запитань дає можливість інтерв'юеру дізнатися точну інформацію, яка йому потрібна, що може бути корисним для збору вимог, оскільки важливо мати чітке розуміння того, що вимагає зацікавлена сторона. Проте, коли певний набір запитань задається без відхилень, немає можливостей для подальшого дослідження зацікавленої сторони або вивчення інших тем, які можуть виникнути.

Взявши найкраще з обох методів, напівструктуровані інтерв'ю дозволяють інтерв'юеру мати орієнтовний набір запитань, водночас дозволяючи додаткове

зондування відповідей. Напівструктуровані інтерв'ю надають інтерв'юеру велику гнучкість. Наявність орієнтовних запитань гарантує, що інтерв'ю буде проходити за планом і що вся необхідна інформація буде отримана під час одного інтерв'ю. Малоймовірно, що будуть потрібні подальші раунди інтерв'ю, оскільки інтерв'юер може продовжувати запитувати та досліджувати, поки не буде впевнений, що має всю необхідну інформацію.

В кожному типі інтерв'ю є можливість задавати різні запитання в залежності від конкретної цілі.

Закриті запитання використовуються, коли потрібна коротка, зосереджена відповідь. Відповіді зазвичай будуть правильними чи неправильними, і вони ефективні в сценаріях встановлення фактів. Приклади закритих питань:

– питання «так/ні»;

– вибір зі списку варіантів;

– ідентифікація конкретної інформації.

Хоча закриті запитання не надають стільки деталей, як відкриті, вони можуть бути корисними для охоплення більшої кількості тем за менший проміжок часу.

Навпаки, відкриті запитання дозволяють давати довші відповіді, які можуть надати набагато більше інформації. Відкриті запитання, як правило, допомагають отримати цінну інформацію про різних людей і про те, як вони взаємодіють з системою або бачать її. Ці типи запитань вимагають від респондента пояснення або опису своїх думок, і на них не можна просто відповісти "так" чи "ні". Прикладами відкритих запитань можуть бути запитання про те, що респонденту подобається в існуючій системі або як він її використовує.

Ці типи питань можуть допомогти інтерв'юеру з'ясувати більше деталей за допомогою подальших запитань, щоб отримати більш детальну інформацію. Закриті запитання, в свою чергу, можуть бути корисними, коли інтерв'юер шукає конкретну відповідь.

Завантажені питання, запитання, що наводять респондента на якусь думку, мають бути дуже делікатними і направленими саме на конкретну людину, щоб спрямовати її на відповідь у певному напрямку. Це дуже непомітно змушує респондента задуматись, як вони бачать свій процес, і підводить до того, що він налаштований не так добре, як міг би.

Часто жартівливі, риторичні запитання використовуються для сприяння роздумам і не вимагають відповіді. Постановка риторичних запитань підтримує взаємодію респондента з інтерв’юером.

Послідовні запитання починаються широко і стають більш обмежувальними, закінчуючись закритим питанням [12].

Використання інтерв'ю як метода збору вимог має свої недоліки, які можуть вплинути на ефективність дослідження. Перш за все, інтерв'ю можуть бути часо- та ресурсоємкими. Проведення інтерв'ю з кожним зацікавленим стейкхолдером може забрати значну кількість часу, особливо якщо кількість зацікавлених стейкхолдерів велика. Це може затримати розробку проєкту та збільшити його вартість.

Другий недолік полягає в можливих спотвореннях відповідей. Учасники інтерв'ю можуть бути неуважними або недостатньо відвертими у своїх відповідях, що може призвести до неповного розуміння їхніх потреб та вимог. Крім того, є ризик впливу на учасників інтерв'ю з боку інтерв'юера, що може призвести до бажання учасників відповідати "правильно" або згідно з очікуваннями інтерв'юера, замість вираження їхньої справжньої думки та потреб. Що, в свою чергу, призведе до невалідних вимог, які не відповідають дійсності.

1.2.2 Спостереження користувача

Спостереження користувача є дуже корисним при реалізації ІТ-проєкту, щоб зрозуміти поведінку користувача, виявити проблеми із зручністю використання, отримати розуміння контексту: спостереження за користувачами в їхньому природному середовищі дає змогу зрозуміти їхні робочі процеси, обмеження навколишнього середовища та ситуаційні фактори, які впливають на їхню взаємодію; виявити невиразні потреби, можливості для інновацій та больові точки, а також оцінити можливість довгострокового використання.

Існує чотири основних типи спостереження за користувачами для визначення вимог: спостереження з участю, спостереження без участі, приховане спостереження та відкрите спостереження.

Спостереження з безпосередньою участю спеціаліста з команди розробки, будь-то бізнес-аналітик (BA), системний аналітик або менеджер проєктів (PM), за користувачами дозволяє дослідникам перевірити визначення термінів, які вони використовували в інтерв'ю, спостерігати за подіями, якими користувачі можуть не мати змоги або не бажають поділитися, та ситуаціями, які користувачі описували в інтерв'ю [13].

Спостереження зазвичай вимагають докладного вивчення обмежених чи специфічних груп, що може призвести до недостатньої репрезентативності, надійності і узагальненості результатів. Існує ризик того, що, з одного боку, дослідник може перейняти поведінку групи, яку він вивчає, проводячи відкрите дослідження за участі учасників. З іншого боку, відкрите спостереження, незалежно від того, чи є дослідник активним учасником, може підійти під сумнів щодо достовірності, коли саме учасники можуть змінити свою поведінку, оскільки вони знають, що їх вивчають [14].

1.2.3 Прототипування

Прототип служить візуальним проявом запропонованого результату проєкту, охоплюючи або змодельований програмний інтерфейс, або функціональну модель, що відображає очікуваний кінцевий продукт. У міру наближення кульмінації етапу збору вимог до проєкту використання методології створення прототипів стає важливим для отримання безцінних відгуків. Використання прототипування окремо від інших методів неможливо. Завдяки презентації прототипу зацікавлені сторони отримують відчутне уявлення про передбачуваний результат, що дає їм можливість окреслити межі проєкту, водночас дозволяючи команді проєкту оцінити здійсненність та ефективність початкових концепцій.

Основна мета прототипу полягає в тому, щоб надати зацікавленим сторонам можливість випробувати очікуваний кінцевий продукт з перших вуст, дозволяючи їм надавати практичний зворотний зв’язок щодо різних аспектів, включаючи відсутні функції, зайві функції, області, які потребують уточнення, додаткові входи або результати та невідповідності. Ці відгуки ретельно документуються, щоб інформувати про наступні ітерації прототипу, забезпечуючи відповідність очікуванням зацікавлених сторін і цілям проєкту [15].

Прототипи поділяються на дві основні типології [16]. Їх різновид представлений на рисунку 1.2.

Рисунок 1.2 – Типи прототипів

Горизонтальний прототип програмного забезпечення являє собою модель передбачуваного інтерфейсу користувача, створену для дослідження поведінки системи та уточнення вимог. Він забезпечує видимість функціональності, не включаючи її повного втілення, показуючи зовнішній вигляд екранів та часткову навігацію між ними. Прототип демонструє зовнішній вигляд інтерфейсу користувача, структуру доступу до інформації та функціональні можливості, але не надає повноцінної функціональності. Використання реальних даних у прототипі підвищує його достовірність та допомагає користувачам краще зрозуміти концепцію системи. Горизонтальні прототипи дозволяють користувачам оцінити конкретні варіанти використання та висловити зауваження щодо реалізації функцій, кроків взаємодії та умов виключення. Під час взаємодії з горизонтальним прототипом, користувачі зосереджуються на загальних вимогах та послідовності дій, не заглиблюючись у деталі зовнішнього вигляду елементів інтерфейсу. Точне оформлення інтерфейсу стає актуальним на пізніших етапах розробки після визначення загальної структури інтерфейсу та уточнення вимог до системи.

Вертикальний прототип, також відомий як структурний прототип або перевірка концепції, являє собою модель функціональності програми, що охоплює всі рівні його реалізації від інтерфейсу користувача до сервісних

функцій. Цей тип прототипу імітує роботу реальної системи, що дозволяє оцінити здійсненність передбачуваного підходу до архітектури, оптимізувати алгоритми, проаналізувати пропоновану схему бази даних та перевірити критично важливі терміни. Створення вертикальних прототипів з використанням аналогічних середовищ розробки забезпечує реалістичне моделювання та допомагає досліджувати критично важливі вимоги до інтерфейсу та часу виконання, а також знижувати ризики на етапі проєктування системи.

Перш ніж створювати прототип, необхідно ухвалити рішення про його подальшу долю: чи стане він частиною продукту, що випускається, або буде утилізований після оцінки. Одноразовий прототип, також відомий як дослідницький прототип, використовується для того, щоб позбутись невизначеностей, та покращення вимог до програмного забезпечення. Він створюється швидко і дешево, з фокусом на швидкості реалізації та модифікації, замість стійкості та довготривалої зручності супроводу. Мета одноразового прототипу полягає у вирішенні незрозумілості та двозначностей у вимогах, що знижує ризик продовження розробки та допомагає візуалізувати реалізацію вимог для користувачів та розробників. Одноразовий прототип є найбільш доречним рішенням при зіткненні з невизначеністю або неповнотою вимог до програмного забезпечення. Його використання допомагає виявити прогалини в документації та визначити, чи відповідають вимоги фактичному створенню продукту. Важливо стежити за тим, щоб прототип не ставав надмірно складним, що передбачає додавання до нього більше функцій, ніж необхідно для досягнення поставленої мети.

Еволюційний прототип відрізняється від одноразового тим, що являє собою стійкий архітектурний фундамент для послідовного розвитку кінцевого продукту відповідно до вимог, що змінюються. На відміну від швидкого та тимчасового створення одноразового прототипу, еволюційний прототип вимагає від початку використання якісного коду та приділення особливої уваги

архітектурі та принципам проєктування. Створення еволюційного прототипу займає більше часу, але дозволяє забезпечити легке зростання та гнучке розширення продукту, що особливо важливо для проєктів, що потребують поступового розвитку, наприклад, проєктів з інтеграції інформаційних систем. Еволюційне прототипування включає кілька циклів розробки, починаючи з пробного випуску, який реалізує найбільш зрозумілі і стабільні частини вимог. Відгуки користувачів на прототип та його початкове використання використовуються для покращення та модифікації у наступних циклах. Кінцевий продукт формується як результат серії еволюційних прототипів, що дозволяє користувачам оперативно отримувати зразки, що працюють, і поступово коригувати і уточнювати вимоги [16].

Кожен описаний тип можна комбінувати між собою згідно типологіям, щоб отримати ефективний результат в залежності від типу IT-проєкту.

Прототипи існують декількох видів (таблиця 1.1) і кожен із них може використовуватись при реалізації певного IT-проєкту.

Таблиця 1.1 – Види прототипів

Рівень інтерактивності / Рівень деталізації	Низький	Середній	Високий
Низький	Sketch		
Середній	Wireframe	Wireframe based prototype	
Високий	Mockup		Mockup based prototype

Прототипування пропонує ряд переваг і недоліків у процесі розробки. З позитивного боку це сприяє скороченню часу та витрат за рахунок підвищення якості специфікацій і вимог, що надаються клієнтам. Завдяки створенню

прототипів клієнти отримують уявлення про потенційні проблеми проєкту, витрати та очікувані кінцеві результати, сприяючи прийняттю обґрунтованих рішень і зменшуючи ризик майбутніх катастроф. Крім того, прототипування сприяє покращенню та збільшенню залучення користувачів, узгоджуючи результати проєкту з очікуваннями та вподобаннями клієнтів. Дозволяючи клієнтам взаємодіяти з робочими моделями та надавати негайний зворотний зв'язок, прототипування допомагає усунути непорозуміння, підвищуючи загальну ясність проєкту та задоволення.

Однак прототипування також має деякі недоліки. Недостатній аналіз може статися, коли розробники надмірно зосереджуються на обмежених прототипах, відволікаючи увагу від комплексного аналізу проєкту. Це може призвести до ігнорування рішень, неповних специфікацій або перетворення обмежених прототипів у погано розроблені кінцеві проєкти, що ускладнює роботу з обслуговування. Крім того, у користувачів може виникнути плутанина, якщо клієнти помилково приймають приблизні прототипи за готові проєкти або неправильно припускають, що прототипи точно відображають кінцеву продуктивність системи. Крім того, клієнти можуть розробляти вкладення до функцій прототипу, які в кінцевому підсумку виключаються з кінцевого продукту, що потенційно може призвести до незадоволення та непорозуміння.

1.2.4 Аналіз документів

Вивчення вже існуючих документів може бути ефективним підходом до збору вимог, незалежно або як доповнення до інших методів. Оцінка поточних процесів і документації допомагає аналітику зрозуміти організацію або систему та її поточний стан. Ці документи надають цінну інформацію, наприклад,

ідентифікацію зацікавлених сторін, пов’язаних із системою, а також можуть допомогти з формулюванням питань на інтерв’ю чи опитувальників для виявлення додаткових вимог. Якщо виникають невизначеності щодо існуючих процедур, цей підхід сприяє цілеспрямованому опитуванню під час співбесід. Під час аналізу вимог аналітики можуть виявити проблеми або прогалини в документації, зокрема відсутню інформацію або зайві кроки. Перегляд старих документів вимог дозволяє повторно використовувати вимоги, відкидаючи застарілі, однак важливо перевірити наскільки документи актуальні, оскільки занадто застаріла інформація може ввести в оману процес аналізу. Крім того, перегляд документів може зайняти багато часу, особливо для складних систем із великою документацією [17]. Аналіз документів служить додатковим інструментом поряд з інтерв’ю, анкетами та спостереженнями. Наприклад, організаційні документи можуть прояснити неоднозначні відповіді на інтерв’ю або покращити розуміння спостережуваної поведінки користувачів.

Використання аналізу документів як метода збору вимог також має свої недоліки. Основним з них є неповнота або недостовірність представлення вимог у документах. Це полягає у відсутності можливості отримати контекст або пояснення до вимог, які виявлені в документах. Без можливості діалогу зі стейкхолдерами, як це можливо під час інтерв'ю або фокус-груп, інтерпретація та розуміння вимог може бути обмеженою. Це може призвести до неповного розуміння контексту або справжніх потреб стейкхолдерів, що в свою чергу, може призвести до непорозумінь або неправильного розроблення системи або продукту.

1.2.5 Семінари (воркшопи)

Воркшоп – це організована та керована зустріч, на якій ретельно відібрані зацікавлені сторони збираються разом, щоб вивчити, уточнити, визначити пріоритети, підтвердити та обговорити вимоги. Як правило, під керівництвом кваліфікованого фасилітатора, ці сесії є спільними та базуються на принципах спільного проєктування додатків (JAD) [11].

Спільне проєктування додатків (JAD) – це спільний підхід, який використовується в розробці програмного забезпечення та системному проєктуванні для залучення ключових зацікавлених сторін, у тому числі кінцевих користувачів, до процесу визначення вимог до системи та розробки рішень. Під час сесій JAD зацікавлені сторони, розробники та інші члени команди проєкту інтенсивно працюють разом, щоб визначити вимоги, уточнити очікування та окреслити характеристики та функціональні можливості системи, що розробляється.

Ключові компоненти сеансів JAD зазвичай проводяться в форматі воркшопів (структурованих семінарів), потребують активної участі, проводяться ітераційно та вимагають візуального моделювання результатів роботи [18].

Воркшопи пропонують різні переваги, такі як зменшення дефектів продукту на 20-50%, мінімізація збільшення масштабу до 70% і економія часу та зусиль на 5-15% протягом життєвого циклу проєкту [19]. Вони виявляються цінними, коли BA шукає консенсусу між зацікавленими сторонами, прагне колективного розуміння або прагне ефективно виявити вимоги.

Під час цих воркшопів можуть бути використані різноманітні техніки та допоміжні засоби, включаючи формулювання проблем або сценаріїв на папері чи дошці, фліп-чарти, листки, маркери, проектори, техніку MoSCoW для визначення

пріоритетів, мозковий штурм для генерації ідей, прототипи для вдосконалення та перевірки вимог [20].

Використання воркшопів як метода збору вимог може мати деякі недоліки. По-перше, цей підхід може бути часо- та ресурсовитратним. Проведення воркшопів вимагає значних зусиль з боку організаторів та учасників, включаючи підготовку, участь у зустрічах та обробку отриманої інформації. Крім того, залучення багатьох стейкхолдерів до воркшопів може бути викликом через їхню роз'єднаність у часі та просторі.

Другий недолік полягає в можливості домінування деяких учасників над іншими. Воркшопи можуть мати неформальну структуру, що дозволяє активним учасникам зайняти провідну роль у дискусіях та прийнятті рішень, тим самим пригнічуючи голоси менш активних учасників. Це може спричинити втрату цінної інформації від тих, хто менш активно бере участь, та спотворення результатів воркшопу.

1.2.6 Анкетування

Наявність широкого кола зацікавлених сторін у IT-проєкті створює труднощі для розуміння їхніх вимог. Чудово мати багато вхідних даних, але потрібно стежити за витратами на початкових етапах проєкту. Використання опитування або анкети для збору інформації щодо вимог проєкту є економічно ефективним і усуває обмеження, такі як географічно розосереджена база зацікавлених сторін або проблеми з часом.

Для того, щоб провести анкетування, першочергово, вибір, який потрібно зробити, полягає в тому, використовувати друковану чи електронну копію для збору інформації.

Онлайн-електронна копія має явні економічні переваги, оскільки не потрібно друкувати чи вводити дані пізніше. Однак друкована копія може збільшити доступність, якщо не всі зацікавлені сторони мають надійний комп’ютер та доступ до Інтернету.

Далі представлено різні типи запитань, які можна використати:

– шкали Лайкерта, які дозволяють респондентам зробити лінійний вибір, наприклад, «повністю згоден, згоден, нейтрально, не згоден, категорично не згоден», що допоможе зрозуміти ставлення, але може дати спотворені позитивні відповіді;

– оцінки та рейтинги дозволяють респондентам вибрати відповідь за шкалою 0-5 або 0-10, яку легко зрозуміти та проаналізувати, але ця інформація стосується лише визначеного списку, а не є незалежними даними;

– поле для прапорців і скидання пропонують список попередньо визначених відповідей, що забезпечує контроль над даними для більш відкритих запитань, але обмежує креативність відповідей, які можна отримати;

– текстові поля дають вашим респондентам повний контроль над тим, що вони говорять, надаючи точні відповіді, але їх аналіз і класифікація можуть зайняти багато часу [21].

Використання анкетування як метода збору вимог також має свої недоліки, які можуть обмежити його ефективність. По-перше, анкети можуть бути обмеженими у своїй можливості детально вивчати вимоги. Оскільки анкети часто базуються на стандартизованих питаннях, вони можуть не захоплювати всіх аспектів потреб стейкхолдерів, що може призвести до втрати цінної інформації.

Другий недолік полягає у відсутності можливості отримати додаткові пояснення. У порівнянні з інтерв'ю, де інтерв'юер може задавати додаткові питання для отримання розширених відповідей, анкети не надають цієї можливості. Це може призвести до непорозумінь або неповного усвідомлення

вимог, оскільки деякі деталі можуть бути втрачені через відсутність можливості діалогу з учасниками.

1.2.7 Мозковий штурм

Мозковий штурм – це метод групової роботи, спрямований на швидке збирання ідей, рішень та пропозицій від учасників команди або зацікавлених сторін для визначення існуючих вимог. Цей процес організується у колективному середовищі під керівництвом модератора, який стимулює творчий потенціал учасників і відбирає ідеї для подальшого аналізу.

Мозковий штурм складається з двох основних етапів:

– сесія збору ідей: учасники групи пропонують будь-які ідеї, що приходять їм на думку, без обговорення або критики, і ці ідеї записуються безпосередньо на дошці або в електронному форматі;

– аналіз ідей: після закінчення сесії збору ідей модератор разом з командою або учасниками проводить аналіз зібраних ідей, що може включати класифікацію, ранжування, фільтрацію або комбінування ідей для визначення найкращих варіантів [22].

Мозковий штурм як метод генерації необхідних ідей може бути використаний на різних етапах життєвого циклу проєкту, таких як:

– збір ідей щодо мети, обсягу, цілей та інших ключових аспектів проєкту;

– виявлення потенційних ризиків та пропозиції щодо їх управління або уникнення;

– визначення стратегій виконання проєкту, ресурсів та графіка робіт;

– пошук альтернативних шляхів вирішення проблем, що виникають під час реалізації проєкту.

Мозковий штурм – це популярна практика збору вимог у багатьох організаціях і проєктах. Її популярність пояснюється кількома факторами. По-перше, вона стимулює творчий процес, дозволяючи учасникам вільно виражати свої ідеї та думки. По-друге, мозковий штурм дозволяє ефективно використовувати час, збираючи велику кількість ідей за короткий період. По-третє, цей метод залучає до процесу збору вимог учасників з різних фахових груп та рівнів, що дозволяє отримати різноманітні погляди та ідеї.

Проте, мозковий штурм має й свої недоліки. Зокрема, існує ризик групової думки, коли учасники можуть підкорятися думці більшості або вибирати перші ідеї, не розглядаючи альтернативні варіанти. Також брак структури та відсутність чіткого керівництва можуть призвести до хаосу та втрати часу. Ідеї, що виникають під час мозкового штурму, можуть бути суб’єктивними та необґрунтованими, що ускладнює їх оцінку та використання. Нарешті, мозковий штурм зазвичай спрямований на швидке збирання ідей, а не на їх ретельний аналіз, що може призвести до пропуску важливих деталей.

Для візуалізації та пріоритезації ідей, що виникли під час мозкового штурму, можна використовувати декілька існуючих методів, таких як складання матриці прийняття рішень, матриці ІКА (важливість, ймовірність, витрати), матриці MoSCow (обов’язково, бажано, необов’язково, критично), а також методи SWOT-аналізу або аналізу Pareto.

1.2.8 Сценарний аналіз

Сценарії надають цінну можливість створення контексту для виявлення потреб користувачів. Вони дають інженерам структуру для розгляду завдань користувача шляхом використання запитань "що, якщо" і "як це зробити". Найпоширенішим типом сценарію є опис варіанту використання [23].

Варіант використання (use case) – це метод, що використовується для виявлення вимог, який служить детальним описом того, як користувачі взаємодіють із системою для виконання певного завдання чи мети. Він описує покрокову взаємодію між користувачем і системою, виділяючи різні сценарії, входи, виходи та умови, пов’язані з досягненням бажаного результату. Варіанти використання не обмежуються лише описом функціональних вимог; вони також охоплюють нефункціональні аспекти, такі як вимоги до продуктивності, безпеки та зручності використання. Хоча основна увага часто приділяється функціональним аспектам, сценарії використання забезпечують комплексне уявлення про поведінку системи та взаємодію користувачів, охоплюючи як функціональні, так і нефункціональні вимоги [24].

Написання історій користувачів також можна віднести до сценарного аналізу, так само як і варіанти використання, оскільки обидва підходи спрямовані на опис поведінки системи з точки зору користувача. У обох випадках акцент робиться на тому, що користувач хоче зробити і як система має реагувати на це.

Історії користувачів (user stories) – це стислі неформальні описи функціональності системи з точки зору кінцевого користувача. Зазвичай вони пишуться простою мовою та відповідають певному шаблону, часто структурованому так: «Як [роль користувача], я хочу [мета], щоб [користь]». Історії користувачів зазвичай використовуються в гнучких методологіях

розробки програмного забезпечення, щоб фіксувати вимоги ітеративно та спільно.

Основна відмінність між історіями користувачів і сценаріями використання полягає в їхньому рівні деталізації та фокусі, що показано на рисунку 1.3:

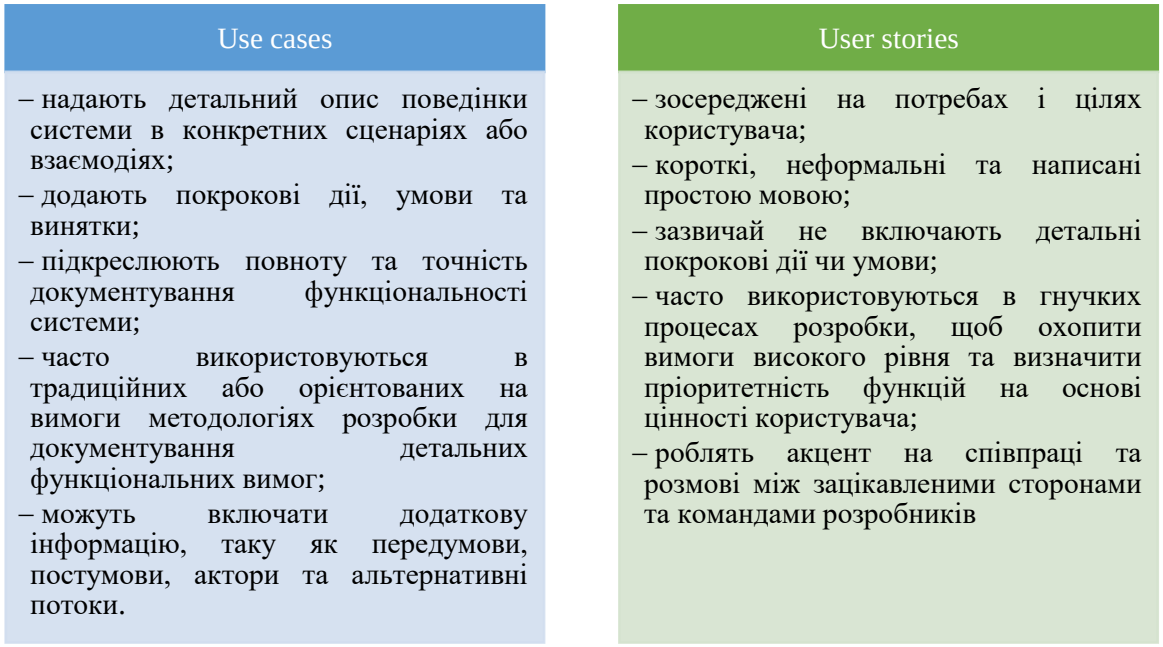

Рисунок 1.3 – Опис особливостей методів збору вимог Use cases & User Stories

Хоча варіанти використанні і історії користувачів служать подібним цілям для визначення системних вимог, вони не є взаємозамінними, і є ситуації, коли одне може бути більш прийнятним, ніж інше. Однак, вони часто доповнюють один одного. У деяких випадках історії користувачів можуть використовуватися для початкового охоплення вимог високого рівня, які потім розробляються в детальних варіантах використання в ході проєкту. Крім того, історії користувачів можуть стати способом визначення пріоритетів функцій і керівництва ітераціями розробки, тоді як варіанти використання надають детальні специфікації для реалізації та тестування. Вибір між їх використанням окремо чи в поєднанні

залежить від конкретних потреб проєкту, використовуваної методології розробки та вподобань команди проєкту.

Варіанти використання та історії користувачів вважаються методами виявлення, а не просто документування вимог, оскільки вони передбачають активну взаємодію із зацікавленими сторонами для визначення, розуміння та вдосконалення вимог. На відміну від простого документування вимог методи виявлення, як-то випадки використання, передбачають інтерактивні сесії, семінари та інтерв'ю для отримання вимог безпосередньо від зацікавлених сторін і користувачів [25]. Тобто цей метод не може бути використаний окремо від інших, які вже були розглянуті вище.

1.2.9 Аналіз інтерфейсів

Аналіз інтерфейсів використовується для виявлення місць, часу, причин, способів і одержувача інформації під час її передачі між компонентами системи або між межами системи.

До видів інтерфейсів відносяться:

– користувацькі інтерфейси, включаючи користувачів, напряму взаємодіючих з рішеннями всередині організації;

– зовнішніх по відношенню до рішення людей, таких як зацікавлені сторони або регулятори;

– бізнес-процеси;

– інтерфейси обміну даними між системами;

– інтерфейси програмних додатків (API);

– інтерфейси фізичних пристроїв.

Аналіз інтерфейсів визначає і проясняє наступне:

– хто буде використовувати інтерфейс;

– яка інформація буде передаватись через інтерфейс і об’єм даних;

– коли і як часто буде передаватися інформація;

– де буде відбуватися обмін інформацією;

– навіщо потрібен даний інтерфейс;

– як інтерфейс реалізований або повинен бути реалізований.

Раннє визначення інтерфейсів надає BA контекст для виявлення більш детальних вимог зацікавлених сторін, тим самим визначаючи необхідне функціональне охоплення рішення. Передчасне визначення інтерфейсів вказує, які сторони отримують користь або залежать від різних компонентів рішення, що допомагає аналітику визначити, які зацікавлені сторони мають бути присутніми в інших методах виявлення вимог. Схематичний опис роботи інтерфейсів показано на рисунку 1.4 [17].

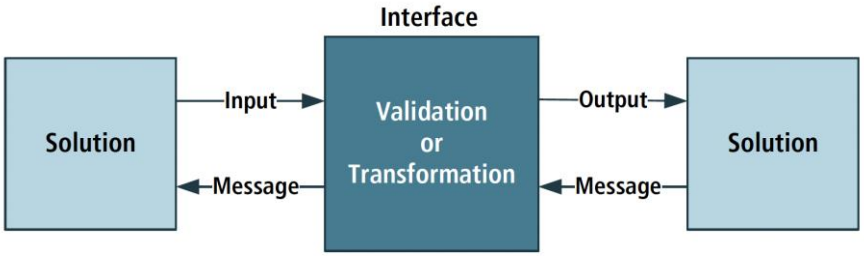

Рисунок 1.4 – Опис інтерфейсу

В цілому, переваги цього підходу виявляються в його спроможності впливати на планування та управління процесом впровадження, сприяти співпраці між різними сторонами та уникненні проблем інтеграції. Однак важливо розуміти, що цей підхід не забезпечує повної картини всіх аспектів

рішення та може вимагати додаткових методів для оцінки внутрішніх компонентів та аналізу інших аспектів рішення.

Однак, використання аналізу інтерфейсів як метода збору вимог може мати обмежену ефективність через низку недоліків. По-перше, інтерфейси можуть бути розроблені з урахуванням вже існуючих вимог та обмежень, що може ускладнити виявлення нових або змінених потреб користувачів. Це може призвести до непорозумінь або пропуску ключових деталей, що не враховуються у вже існуючих інтерфейсах. Крім того, відсутність можливості взаємодії з реальними користувачами може обмежити здатність аналізувати їхні реальні потреби та проблеми, які вони можуть зазнавати при використанні інтерфейсів.

1.2.10 Фокус-групи

Щоб ефективно зібрати вимоги до проєкту, фокус-групи дають цінну інформацію про ключові елементи проєкту шляхом залучення різноманітних зацікавлених сторін і точок зору. За допомогою фокус-груп визначаються можливості, потреби та виклики проєкту, перевіряються та вдосконалюються ідеї, раніше зібрані іншими способами. Проведення фокус-групи вимагає ретельного розгляду кількох факторів.

По-перше, стратегічний вибір учасників має вирішальне значення, орієнтований на різноманітність зацікавлених сторін і кінцевих користувачів, зазвичай залучаючи 6-12 осіб.

По-друге, формулювання цілеспрямованих запитань має важливе значення для ефективного ведення розмови. Запитання мають бути конкретними, щоб викликати конструктивний зворотній зв’язок, уникаючи надто відкритих запитань, які можуть призвести до непродуктивних дискусій.

Хоча фокус-групи пропонують багату якісну інформацію для розуміння вимог до проєкту, існують потенційні підводні камені, які слід враховувати. До них належать потреба в обробці перед аналізом, ризик того, що менталітет натовпу вплине на результати, і ймовірність того, що сесії переростуть у негативні дискусії без належного керівництва та запитань.

Незважаючи на ці труднощі, фокус-групи пропонують численні переваги, зокрема отримання глибокого розуміння потреб зацікавлених сторін, охоплення різноманітних точок зору користувачів і отримання більш цілеспрямованих результатів порівняно з сеансами мозкового штурму.

Підсумовуючи, фокус-групи служать цінним інструментом на останніх етапах збору вимог, надаючи прямий внесок від кінцевих користувачів для ефективного вдосконалення інформації про проєкт. Підготовка та кваліфікована модерація є важливими для забезпечення успішних результатів фокус-групових сесій, що призведе до чіткого та дієвого розуміння проєкту [26].

1.3 Дослідження типів ІТ-проєктів

Множину проєктів можна класифікувати за різними ознаками [27]. Загальноприйнятими класифікаційними ознаками є:

а) спрямованість (або сфера діяльності);

1) технічна;
2) організаційна;
3) економічна;
4) соціальна;
5) змішана;

б) масштаб (за обсягами фінансування) визначається трудомісткістю проєктних робіт, складністю та обсягом робіт, розміром загальних інвестиційних витрат;

1) малий;

2) середній;

3) великий;

в) тривалість реалізації проєкта;

1) короткострокові;

2) середньо-строкові;

3) довгострокові;

г) складність реалізації:

1) проста;

2) складна;

3) унікальна;

д) призначення проєкта;

1) інноваційне;

2) інвестиційне;

3) науково-дослідне;

4) навчально-освітнє;

5) змішане;

е) характер учасників;

1) вітчизняний (державний, регіональний, місцевий);

2) зарубіжний;

3) спільний;

ж) вимоги до якості проєкта:

1) бездефектний;

2) модульний;

3) стандартний;

з) географічна ознака проєкта;

1) територіальна;

2) регіональна;

3) державна;

4) міжнародна.

Термін «ІТ-проєкт» використовують для позначення діяльності, пов’язаної з використанням або створенням деякої інформаційної технології. ІТ-проєкти охоплюють такі сфери діяльності:

- розробку і розвиток програмних застосунків;
- впровадження інформаційних систем (ІС);
- розгортання ІТ-інфраструктури.

Для конкретизації інформації для кваліфікаційної роботи описано більш детально малі за масштабом ІТ-проєкти будь-якої сфери діяльності. Малі проєкти дозволяють спростити процес проєктування та впровадження, включаючи формування команди проєкту, за допомогою швидкого перерозподілу ресурсів. Однак, відсутність часу для виправлення помилок може ускладнити ситуацію, тому важливо докладно визначити обсяг проєкту, учасників та їхні методи роботи. Це особливо стосується етапів збору та аналізу вимог, графіка проєкту, форм звіту та умов контракту [28].

1.4 Постановка задачі дослідження

Після проведення аналізу методів збору вимог та класифікації ІТ-проєктів видокремлено низку ключових аспектів, які потрібно врахувати. По-перше, однією з основних проблем, що виникають у процесі застосування різних методів збору вимог у ІТ-проєктах, є неоднорідність та неповнота вимог. Це може

виникнути через недостатню увагу до деталей під час збору вимог або через нечіткість у формулюванні вимог. В результаті може виникнути ситуація, коли вимоги не відображають повністю потреби замовника або недостатньо чітко визначають очікувані результати проєкту.

Додатковою проблемою є невідповідність між очікуваннями замовника та реальними можливостями розробки. Це може виникнути внаслідок недостатньої комунікації між командою та замовником або через недоречний вибір методів збору вимог, які не враховують технічний контекст або обмеження, пов'язані з реалізацією проєкту. В результаті може виникнути розрив між очікуваннями стейкхолдерів та можливостями проєкту, що може призвести до затримок у виконанні, недоліків у продукті або навіть до провалу проєкту. Метою кваліфікаційної роботи є дослідження та порівняння існуючих методів збору вимог та визначення необхідних критеріїв для вибору методів на ІТ-проєктах.

Проблеми, описані вище, надають змогу визначити предмет, об'єкт та мету даного дослідження.

Предметом дослідження в рамках кваліфікаційної роботи є процес вибору методів збору вимог на ІТ-проєкті.

Об’єктом дослідження є найпоширеніші методи збору вимог.

Метою даної роботи є порівняльний аналіз методів збору вимог, аналіз їх спільних та відмінних рис, а також вирішення задачі вибору методів збору вимог, базуючись на об’єктивних критеріях.

Для досягнення мети кваліфікаційної роботи необхідно вирішити наступні задачі дослідження:

- проаналізувати існуючі методи збору вимог;
- розробити класифікацію методів збору вимог;
- розробити критерії оцінювання методів збору вимог;

– розробити функцію вигоди для задачі вибору методів збору вимог для малих ІТ-проєктів;

– розробити методологію застосування функції вигоди при виборі методів збору вимог;

– експериментально перевірити отримані результати вирішення задачі вибору методів збору вимог для певного ІТ-проєкту.

2 ПОСТАНОВКА ЗАДАЧІ ВИБОРУ МЕТОДІВ ЗБОРУ ВИМОГ

2.1 Задача класифікації методів збору вимог

Використання всіх методів збору вимог на будь-якому ІТ-проєкті, зокрема малому з обмеженими ресурсами, буде надмірним та не матиме раціонального пояснення.

Класифікація методів збору вимог за схожим принципом збирання дозволить доцільно використовувати наявні ресурси на проєкті, а також мінімізує надмірність використання тих методів, що в результаті надають однакову інформацію. Можна визначити такі класи методів збору вимог: вербальні, аналітичні та візуалізаційні. Таке розбиття методів збору вимог на класи, дозволяє забезпечити комплексний підхід до збору вимог, використовуючи різні підходи для отримання різноманітної інформації від стейкхолдерів. Обираючи хоча б один метод із кожного класу для реалізації ІТ-проєктів для малого бізнесу, забезпечується збалансований підхід, який враховує як вербальні комунікаційні аспекти, так і аналітичні та технічні вимоги, що сприяє успішному виконанню проєкту та задоволенню потреб клієнта.

Класи методів збору вимог, такі як вербальні, аналітичні та візуалізаційні, описуються наступними атрибутами та їх можливими значеннями:

а) тип комунікації;

1) усний – пряма комунікація із стейкхолдерами або користувачами через живий діалог, телефонні розмови, відеоконференції тощо;

2) текстовий – комунікація через електронні листи, повідомлення чатів, текстові документи, зображення, графіки, діаграми тощо;

б) рівень залученості користувачів;

1) активний – рівень, що включає пряму участь користувачів або стейкхолдерів у процесі збору вимог;

2) пасивний – рівень, який вимагає меншої активної участі з боку користувачів;

в) рівень деталізації вимог;

1) низький – методи з низьким рівнем деталізації спрямовані на збір загальних користувацьких або бізнес вимог без деталізації;

2) середній – методи з середнім рівнем деталізації уточнюють конкретні потреби користувачів без докладного переходу до технічних деталей;

3) високий – методи з високим рівнем деталізації, дозволяють отримати конкретні та деталізовані вимоги, наприклад системні вимоги.

Для того, щоб вирішити задачу класифікації методів збору вимог необхідно прописати ознаки того чи іншого класу, які представлені в таблиці 2.1.

Таблиця 2.1 – Значення атрибутів для кожного класу

Клас / Атрибут	Вербальний	Аналітичний	Візуалізаційний
Тип комунікації	Усний (У)	Текстовий (Т)	Текстовий (Т)
Рівень залученості користувачів	Активний (А)	Пасивний (П)/Активний (А)	Пасивний (П)
Рівень деталізації вимог	Низький (Н)	Середній (С)	Високий (В)

Використовуючи таблицю 2.1, кожен із описаних в даному дослідженні методів збору вимог класифіковано. Результати класифікації представлені в таблиці 2.2.

Таблиця 2.2 – Класифікація методів збору вимог

Метод	Атрибут			Клас
	Тип комунікації	Рівень залученості користувачів	Рівень деталізації вимог	
Інтерв'ю	У	А	Н	Вербальний
Спостереження користувача	Т	П	С	Аналітичний
Прототипування	Т	П	В	Візуалізаційний
Аналіз документів	Т	П	С	Аналітичний
Семінари	У	А	Н	Вербальний
Анкетування	Т	А	С	Аналітичний
Мозковий штурм	У	А	Н	Вербальний
Сценарний аналіз	Т	П	В	Візуалізаційний
Аналіз інтерфейсу	Т	П	В	Візуалізаційний
Фокус-групи	У	А	Н	Вербальний

Дані результати класифікації дозволять головному менеджменту ІТ-проєкту для малого бізнесу при виборі найбільш підходящих методів збору вимог не втратити цілісність отриманих результатів.

2.2 Розробка критеріїв оцінки методів збору вимог

Далі виникає задача вибору методів збору вимог із створених класів. Для вирішення цієї задачі пропонується багатокритеріальний підхід: визначення критеріїв оцінки збору вимог, визначення їх кількісних значень та ваги в залежності від особливостей малого ІТ-проєкту.

Вхідна інформація, яка є у кожного ІТ-проєкту, та її фізичний сенс на етапі збору вимог:

– склад команди – мінімально необхідні спеціалісти для використання методу збору вимог;

– досвід команди – необхідні навички спеціалістів для використання методу збору вимог;

– строки – час, за який будуть представлені проаналізовані результати після використання методу збору вимог;

– бюджет – кошти, які необхідні при використанні методу збору вимог на часові затрати відповідного спеціаліста.

Послідовність вхідної інформації невипадкова, бо при реалізації ІТ-проєкту в малій компанії перше, на що спирається менеджмент, це на можливості учасників ІТ-команди, потім на час, за який ця конкретна команда представить результати, і, наприкінці, із вищезазначеної інформації розраховується бюджет на використання методу збору вимог.

Для того, щоб перетворити ці вхідні дані на критерії, необхідно надати їм конкретні кількісні показники.

Кількісні показники для критеріїв, які розробляються, можуть бути представлені двома типами: діапазонні значення та порогові значення. Діапазонні значення застосовуються для критеріїв строки та бюджет, бо час виконання

роботи при використанні різних методів може варіюватись в залежності від обсягу задачі та компетенції працівника, відповідно кошти на забезпечення виконання роботи залежать від використаного часу.

Порогові значення застосовуються при оцінці складу та досвіду команди. Пороговим значенням складу команди буде мінімальна кількість спеціалістів із потенційно необхідними хард-скілами. Підтвердження оцінки навичок спеціаліста відбувається при валідації його минулих результатів на проєктах – чим менше було ітерацій змін після його роботи, тим більше компетенції він має.

В таблиці 2.3 представлені середні значення для критеріїв вибору методів збору вимог. Значення затрачених коштів розраховувалось від усередненого значення заробітної плати IT-спеціаліста в Україні [29].

Таблиця 2.3 – Критерії оцінки методів збору вимог

Клас	Метод	Склад команди	Досвід команди (навички)	Строки (години)	Бюджет ($/годину)
Вербальний	Інтерв'ю	1 BA/PM	Проведення інтерв'ю; Аналіз даних	3-5 годин	33-65$
	Семінари	1 PM/BA	Проведення семінарів та модерація дискусій; Знання предметної області	5-10 годин	55-130$
	Фокус-групи	1 BA/PM	Фасилітація; Знання предметної області	5-8 годин	11-104$

Продовження таблиці 2.3

Клас	Метод	Склад команди	Досвід команди (навички)	Строки (години)	Бюджет ($/годину)
Вербальний	Мозковий штурм	1 PM – модератор, 1 BA, 1-2 Software engineer, 1-2 QA engineer	Генерування ідей та командної роботи; Знання предметної області	2-3 години	96-216$
Аналітичний	Аналіз документів	1 BA	Читання та розуміння документів; Аналіз технічної документації	20-40 годин	260-520$
	Анкетування	1 BA	Розробка анкет; Аналіз даних	2-3 години	26-39$
	Спостереження користувача	1 BA 1 PM 1 SME	Фасилітація великої кількості людей; Дослідження користувачів та інтерпретація даних.	5-20 годин	170-680$
Візуаліза-ційний	Аналіз інтерфейсів	1 BA, 1 UI/UX дизайнер, 1 Software engineer або 1 QA engineer	Аналіз інтерфейсів – UX/UI-дизайну; Тестування	2 години на інтерфейс	Від 65$

Кінець таблиці 2.3

Клас	Метод	Склад команди	Досвід команди (навички)	Строки (години)	Бюджет ($/годину)
Візуаліза-ційний	Сценарний аналіз	1 BA	Знання предметної області; Базові навички UX/UI-дизайну; Інтерпретація даних дослідження користувачів	0,1 – 0,5 годин на один варіант використання або історію користувача	Від 2$
	Прототипування	1 BA, 1 UX/UI дизайнер, (1 Software engineer – якщо необхідна демо-версія реального продукту)	Проведення презентацій; UX/UI-дизайн; (створення архітектури програмного рішення)	Від 2 годин на простий прототип до 80 на інтерактивний	70-3120$

2.3 Розробка функції вигоди

Методом рішення багатокритеріальної задачі вибору методів збору вимог пропонується зробити функцію вигоди K_{profit} з використанням інтегральних адитивних критеріїв, де сума відхилень при використанні методу збору вимог

прагне до нуля. Обирати метод з класу методів збору вимог варто той, в якого сума відхилень буде найменшою. Функція вигоди має такий вигляд:

$$K_{profit} = \sum_{i=1}^{n} C_i\left(|K_i - K_i^P|\right) \to 0, \quad (2.1)$$

де n – кількість критеріїв, які розглядаються;

C_i – ваговий коефіцієнт i-го критерія;

K_i – числове значення локального критерія;

K_i^P – i-те нормоване значення локального критерія.

Ваговий коефіцієнт C_i визначається за допомогою експертного оцінювання, що рахується за формулою 2.2. Кожен проєкт може мати свої ваги для кожного критерію, в залежності від особливостей цього проєкту.

$$C_i = \frac{\sum_{j=1}^{m} x_j}{m}\ , \quad (2.2)$$

де x_j – оцінка j-го експерта;

m – кількість експертів.

Слід зауважити, що в даному випадку використовуються не чисто числові значення критеріїв, а діапазонні та порогові значення. Тому необхідно їх привести до нормованого вигляду для можливого аналізу відхилень.

Для діапазонних значень (строки та бюджет) відхилення буде рахуватись наступним чином: якщо $K_i^{min} \le K_i \le K_i^{max}$, де K_i^{min} – нормоване мінімальне значення діапазону, K_i^{max} – нормоване максимальне значення діапазону, тоді $\sigma = 0$. Якщо $K_i^{min} \ge K_i$, тоді $\sigma = \left|K_i - K_i^{min}\right|$. Якщо $K_i^{max} \le K_i$ тоді $\sigma = |K_i - K_i^{max}|$. Однак, для методів сценарного аналізу та аналізу інтерфейсів нормальне значення строків буде варіюватись в залежності від кількості функцій та

інтерфейсів відповідно. Сама кількість функцій або інтерфейсів стане відома після використання методів із вербального та аналітичного класів.

Також слід зауважити, що принцип використання даної функції базується на тому, що будь-яке відхилення призводить до негативних наслідків. З точки зору бізнесу, якщо витрати на метод були менші ніж заплановано, це позитивний результат, але в контексті даного дослідження такий результат також має вважатись негативним. Зменшення витрат на проєкт може призвести до погіршення якості продукту або послуги через скорочення обсягів робіт або використання менш ефективних рішень, що може відображатися на задоволенні клієнтів і спричинити негативне ставлення до бренду компанії. Також це може створити конфлікти з партнерами та постачальниками через скорочення оплати чи зміни умов контракту, що може погіршити відносини та вплинути на майбутню співпрацю.

Для визначення порогового значення «складу команди» пропонується оцінювати відхилення у відсотках значимості участі кожного члена команди при використанні кожного методу збору вимог. Тут 100% відповідає значенню 1, що є максимальною сумою всіх значень для кожного члена команди (таблиця 2.4).

Таблиця 2.4 – Нормоване значення критерію «Склад команди» для розрахунку відхилень

Метод	Склад команди	Відсоток значимості участі члена команди
Інтерв’ю	1 BA/PM	1
Семінари	1 PM/BA	1
Фокус-групи	1 BA/PM	1

Кінець таблиці 2.4

Метод	Склад команди	Відсоток значимості участі члена команди
Мозковий штурм	1 PM – модератор, 1 BA, 1-2 Software engineer, 1-2 QA engineer	0,25 0,25 0,25 0,25
Аналіз документів	1 BA	1
Анкетування	1 BA	1
Спостереження користувача	1 BA 1 PM 1 SME	0,5 0,3 0,2
Аналіз інтерфейсів	1 BA, 1 UX/UI дизайнер, 1 Software engineer або 1 QA engineer	0,25 0,40 0,35
Сценарний аналіз	1 BA	1
Прототипування	1 BA, 1 UX/UI дизайнер, (1 Software engineer – якщо необхідна демо-версія реального продукту)	0,4 (0,3 при участі Software engineer) 0,6 (0,4 при участі Software engineer) (0,3)

Для порогового значення «досвід команди» пропонується прийняти за 0 або 1 відсутність або наявність відповідно необхідної навички. Сума наявності або відсутності навичок і буде K_i значенням локального критерію «досвід команди».

3 РОЗРОБКА МЕТОДОЛОГІЇ ЗАСТОСУВАННЯ ФУНКЦІЇ ВИГОДИ ПРИ ВИБОРІ МЕТОДІВ ЗБОРУ ВИМОГ

3.1 Розробка матриці прийняття рішення вибору методів збору вимог

На базі результатів другого розділу кваліфікаційної роботи пропонується розробити загальну матрицю прийняття рішень для вибору методів збору вимог, яка вказана в таблиці 3.1, на етапі збору і аналізу вимог, де Cm_i – запропонований клас методів збору вимог (таблиця 2.2), m_i – конкретний метод збору вимог, K_i – критерій оцінки методу, $K_{profit\ i}$ – результат відхилення після застосування функції вигоди 2.1, $M = \min(K_{profit\ i})$ – конкретний метод із запропонованих у класі для використання на проєкті.

Таблиця 3.1 – Матриця прийняття рішень для вибору методів збору вимог

Клас	Метод	Склад команди	Досвід команди	Строки	Бюджет	σ	Висновки
Cm_i	m_i	K_1	K_2	K_3	K_4	$K_{profit\ i}$	M

Слід зауважити, що в даній роботі детально розглядається вибір методів збору вимог для малих ІТ-проєктів, тому вибір методів з класу обмежується тільки одним методом з багатьох. Однак, при необхідності, ресурсної можливості або вимогах замовника, є можливість обирати декілька методів із класу для проведення якісного етапу збору вимог, в яких будуть мінімальні відхилення після використання функції вигоди.

3.2 Схема заповнення матриці прийняття рішень

Процес ІВ, незалежно від етапу життєвого циклу програмного забезпечення, вимагає уважного планування та оцінки. Однак, на етапі аналізу, коли вимоги мають стати більш конкретними та деталізованими, планування активностей набуває особливого значення.

Перед початком процесу інженерії вимог слід здійснити попередній етап планування. На цьому етапі визначається:

а) план активностей;

1) визначається які активності проводити і коли;
2) визначається список артефактів;
3) оцінюються ризики;
4) визначається система метрик;

б) аналізуються зацікавлені сторони (стейкхолдери);

в) розробляється план затвердження вимог;

г) обираються інструменти для документування.

Планування дозволяє заздалегідь спланувати процес збору вимог, визначити необхідний ресурсний потенціал та уникнути можливих недоліків чи збоїв у майбутньому. Саме на цьому етапі пропонується заповнити матрицю прийняття рішень по вибору методів збору вимог (таблиця 3.1). Шість етапів заповнення матриці описані нижче.

Етап 1. РМ на старті проєкту має документ, який містить розподіл ресурсів, в тому числі бюджет і команду цього проєкту. Використовуючи ці документи необхідно відокремити склад команди, яка буде займатись етапом збору та аналізом вимог, і занести ці значення в таблицю 3.1 в K_1.

Етап 2. Після визначення значення K_1, необхідно визначити K_2 – оцінити навички команди, яка потенційно може виконувати задачу збору вимог.

Необхідно відокремити всі можливі навички спеціалістів для використання методів збору вимог, які є значеннями атрибуту «досвід команди» в таблиці 2.3. Після цього PM або інша відповідальна особа, наприклад, HR має оцінити наявність або відсутність цих навичок, спираючись на їх попередні результати на проєктах.

Етап 3. Для визначення K_3 – часу на використання того чи іншого метода – PM необхідно провести мозковий штурм з членами команди для визначення часу виконання задачі або створити невелике анкетування спеціалістів, де вони можуть визначити свій суб'єктивний час на виконання задачі, а PM надасть цим значенням більш об'єктивну оцінку.

Етап 4. Конкретне значення критерія K_4 рахується за формулою:

$$K_4 = \sum_{i=1}^{n}(S_i * K_3), \tag{3.1}$$

де n – кількість необхідних спеціалістів;

S_i – заробітна плата i-го спеціаліста за годину;

K_3 – критерій «Строки».

Етап 5. Далі, використовуючи формулу 2.1, необхідно заповнити значення атрибуту відхилення σ для кожного методу збору вимог, що розглядається.

Етап 6. Спираючись на отримані дані на етапі 5, можна зробити висновки, тобто вибрати метод із запропонованих класів з мінімальним відхиленням.

Підсумовуючи, використання даної матриці прийняття рішення при виборі методів збору вимог на IT-проєкті дозволяє ще на етапі планування мінімізувати проблему неповноти вимог, бо кожний клас методів охоплює свої аспекти потенційних вимог, а функція вигоди допомагає з об'єктивним вибором методів, спираючись на вхідні фактори проєкту.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ МЕТОДОЛОГІЇ ВИБОРУ МЕТОДІВ ЗБОРУ ВИМОГ

4.1 Загальний опис ІТ-проєкта по роботизації проведення банківської виписки

Для прикладу проведено аналіз використання методів збору вимог на ІТ-проєкті по роботизації проведення банківської виписки.

Метою проєкту є автоматизація процесу проведення банківської виписки з використанням роботизованих процесів автоматизації (RPA). Це дозволить значно підвищити ефективність роботи, зменшити людські помилки та знизити операційні витрати. Проєкт розрахований на два місяці, протягом яких буде впроваджено програмне забезпечення для автоматичного оброблення банківських транзакцій, включаючи завантаження, перевірку та інтеграцію даних у внутрішні системи компанії.

Проведення банківської виписки включає в себе наступні кроки: спочатку необхідно зайти в банк та вивантажити виписки з банківського рахунку. Після цього ці виписки імпортуються в бухгалтерську систему компанії. На завершальному етапі проводяться непроведені документи згідно з встановленими спеціальними правилами.

Проєкт передбачає розробку та впровадження роботизованої системи, яка автоматизує всі ці кроки. RPA-боти будуть запрограмовані для автоматичного входу до банківської системи, завантаження банківських виписок, їхнього імпорту в бухгалтерську систему та проведення документів відповідно до визначених правил. Це дозволить зменшити ручну роботу, мінімізувати ризик помилок та прискорити процес проведення банківської виписки.

4.2 Аналіз вхідних даних на етапі збору вимог IT-проєкта проведення банківської виписки

Для порівняння двох альтернатив – вибір методів на основі історичних даних (A) та вибір методів за розробленою методологією (B) – можна використати A/B тестування, де кінцевим результатом ефективності буде кількість запитів на зміни на етапі збору вимог: чим вона менше, тим краще. Такий підхід дозволяє об’єктивно оцінити вплив використання методології на процес управління проєктом.

A/B-тестування – це спосіб порівняти дві версії чогось, щоб визначити, яка ефективніша. Хоча його найчастіше асоціюють із веб-сайтами та додатками, цьому методу майже 100 років, і це одна з найпростіших форм рандомізованого контрольованого експерименту [30].

Вхідними даними перед початком експерименту на етап збору вимог є склад команди, досвід команди, а також заробітна плата кожного члена команди, які вказані в таблиці 4.1. Ці дані будуть незмінні для обох вибірок при тестуванні альтернатив. Також зазначимо, що і перша і друга група тестування входила в рамки бюджету всього проєкту. Стейкхолдери зазначили, що результати їм потрібен в межах 20 годин.

Таблиця 4.1 – Вхідні дані ІТ-проєкту проведення банківської виписки

Склад команди	Досвід команди	ЗП/год
1 BA	проведення інтерв'ю; аналіз даних; інтерпретація даних дослідження користувачів; розробка анкет; генерування ідей та командної роботи; читання та розуміння документів; знання предметної області	10$
1 PM	проведення інтерв'ю; фасилітація великої кількості людей; генерування ідей та командної роботи; проведення презентацій; проведення семінарів та модерації дискусій	11$

4.2.1 Аналіз результатів ефективності альтернативи А

Команда А вирішила використовувати тільки інтерв’ю (2 години) та сценарний аналіз (4 години), на виході отримуючи технічне завдання з 5 функціями нової роботизованої системи:

- вхід в банки;
- експорт виписок;
- вхід в бухгалтерську систему підприємства;
- імпорт виписок;
- проведення виписок.

Актори процесу проведення виписки взаємодіють з двома інтерфейсами – сайтом банку та програмою бухгалтерського обліку, які беруть участь у бізнес-процесі. Після представлення результатів аналізу стейкхолдерам запитів на зміни було 3 та кількість альтернативних сценаріїв для останньої функції з 4 зросла до 10.

Тобто використання лише інтерв'ю та сценарного аналізу для збору вимог виявилось недостатнім для повного та точного визначення функцій нової роботизованої системи. Методи, що використовувалися, не забезпечили достатньої гнучкості для швидкого адаптування до змін у вимогах. Це підтверджується кількістю запитів на зміни після представлення результатів стейкхолдерам (їх було три), що вказує на необхідність переробки значної частини вимог. Кількість альтернативних сценаріїв для останньої функції зросла з 4 до 10, що свідчить про те, що початковий аналіз не врахував всіх можливих варіантів використання функції. Це може призвести до затримок у розробці та додаткових витрат.

4.2.2 Аналіз результатів ефективності альтернативи В

Команда В з тими ж вхідними даними скористалась розробленою методологією та заповнила таблицю 3.1 для свого проєкту. Також після проведення опитування експертів: команди цього проєкту, результати якого вказані в таблиці 4.2, і було отримано ваги для кожного критерію, де 1 – найменш критичне дотримання, 5 – найбільш критичне дотримання.

Таблиця 4.2 – Ваги критеріїв вхідних даних ІТ-проєкту

Критерій	Результати					Вага критерію
	Найменш критично	Не сильно критично	Середня критичність	Сильно критично	Найбільш критично	
Бюджет	-	-	-	6	4	4,4
Строки	-	-	1	8	1	4
Склад команди	-	-		3	7	4,7
Досвід команди	-	-	1	5	4	4,3

Результати розрахунків та заповнення матриці прийняття рішень представлені в таблиці 4.3, де кожен метод має свій ідентифікатор:

– В1 – інтерв’ю;

– В2 – семінари;

– В3 – фокус-групи;

– В4 – мозковий штурм;

– А1 – аналіз документів;

– А2 – анкетування;

– А3 – спостереження користувача;

– Вз1 – аналіз інтерфейсів;

– Вз2 – сценарний аналіз;

– Вз3 – прототипування.

Таблиця 4.3 – Заповнена матриця прийняття рішень для ІТ-проєкту

Клас	Метод	Склад команди	Досвід команди	Строки	Бюджет	σ	Висновки
Вербальний	В1	1 ВА	проведення інтерв'ю; аналіз даних	3	30	13,2	В2 – Семінари
	В2	1 РМ	знання предметної області; проведення семінарів та модерації дискусій	7	77	0	
	В3	1 РМ	знання предметної області; фасилітація	4	66	4	
	В4	1 ВА 1 РМ	генерування ідей та командної роботи; знання предметної області	2	42	239,7	
Аналітичний	А1	1 ВА	читання та розуміння документів;	10	100	748,3	А2 – Анкетування
	А2	1 ВА	аналіз даних; розробка анкет	2	20	26,4	
	А3	1 ВА 1 РМ	інтерпретація даних дослідження користувачів; фасилітація великої кількості людей	6	126	194,44	
Візуалізаційний	Вз1	1 ВА	-	16	160	191,75	Вз2 – Сценарний аналіз
	Вз2	1 ВА	знання предметної області, інтерпретація даних дослідження користувачів	4	40	30,5	
	Вз3	1 ВА	-	8	80	187,52	

Для наочної візуалізації розраховано за формулою 2.1 відхилення для методу В1 – інтерв’ю. Результати розрахунків представлені нижче:

$$4{,}2\ (|1-1|) + 4{,}3\ (|2-2|) + 4(|3-3|) + 4{,}4(|30-33|) = 13{,}2$$

Для критерію «Склад команди» ефективність людини на проєкті дорівнює 100%, тому можна вважати, що відхилення по цьому критерію дорівнює 0. Всі необхідні навички для використання методу наявні у даного спеціаліста, тому відхилення по критерію «Досвід команди» також дорівнює 0. Строки, за які цей спеціаліст впорається з використанням цього методу та надасть результати, дорівнюють мінімальному значенню з нормального діапазону, тому відхилення цього критерію також є нульовим. Але, зважаючи на заробітну плату за годину цього працівника, бюджет, який необхідно використати на проведення інтерв'ю, менше за нормований. Однак, причиною такого результату може неправильне оцінювання часу виконання задачі, що в подальшому призведе до переробок. Тому результуючим значенням відхилення для методу інтерв'ю буде 13,2.

Подальші розрахунки проводяться аналогічно.

Підсумовуючи, команда В використала 3 методи, які були обрані згідно з методологією (семінари, анкетування та сценарний аналіз), виділила 5 функцій та 2 інтерфейси і отримала 1 запит на зміни, який додав 2 додаткових правила в альтернативний сценарій функції «Проведення виписки».

4.3 Аналіз результатів A/B тестування

Виходячи із даних, отриманих після A/B тестування, можна зробити висновок, що використання розробленої методології призвело до зменшення кількості запитів на зміни на 66% на проєкті та підвищило якість першої спроби задокументованих вимог. Наочне представлення результатів тестування вказано в таблиці 4.4.

Таблиця 4.4 – Результати A/B тестування

Альтернативи	Методи, які були використані	Кількість запитів на зміни	Години додаткової роботи
Альтернатива А	Інтерв'ю, Сценарний аналіз	3	16
Альтернатива В	Семінари, Анкетування, Сценарний аналіз	1	3

Також, можна відзначити, що через велику кількість доробок, фактичний час на завершення етапу збору вимог у команди А вийшов за рамки на 2 години.

Команда В, використовуючи семінари, анкетування та сценарний аналіз, досягла значно кращих результатів у зборі вимог для нової роботизованої системи. Це стало можливим завдяки декільком важливим чинникам, таким як широке залучення стейкхолдерів, структурований збір даних та глибокий аналіз сценаріїв.

Проведення семінарів дозволило команді В зібрати необхідну кількість стейкхолдерів в одному місці, що сприяло обговоренню вимог в режимі реального часу та позбавило проблеми неузгодженості потреб між різними стейкхолдерами на самому підприємстві.

Анкетування дозволило систематично збирати вимоги від всієї кількості Process Owners. Це дало змогу отримати детальну та різнобічну інформацію про потреби та очікування користувачів, надало майже всі виключення для проведення виписок, що забезпечило більш точне та комплексне формулювання вимог.

Поєднання сценарного аналізу з іншими методами дало змогу краще розуміти реальні умови використання системи. Це дозволило виявити потенційні

проблеми та врахувати їх на етапі планування, що значно знизило кількість запитів на зміни після впровадження вимог.

У результаті команда В виділила 5 функцій та 2 інтерфейси, що відповідали бізнес-процесам користувачів. Було отримано лише 1 запит на зміни, який додав 2 додаткових правила в альтернативний сценарій функції «Проведення виписки». Це свідчить про те, що початковий аналіз був проведений ретельно та врахував більшість можливих варіантів використання системи. Такий комплексний підхід до збору вимог забезпечив високу точність та релевантність отриманих даних, що значно підвищило ефективність проєкту в цілому.

ВИСНОВКИ

За завданням кваліфікаційної роботи було проведено дослідження методів збирання вимог на ІТ-проєктах та класифікації ІТ-проєктів. На основі цих досліджень було визначено предмет, об’єкт та мету даного дослідження.

У рамках кваліфікаційної роботи було доведено актуальність дослідження, проаналізовані існуючі підходи до вибору методів, обґрунтована мета розробки матриці прийняття рішень для використання у методології при виборі методів збору вимог. Також сформовані та нормалізовані критерії оцінки методів збору вимог.

Для досягнення мети кваліфікаційної роботи було вирішено наступні задачі дослідження: проаналізовано існуючі методи збору вимог, розроблено класифікацію методів збору вимог, розроблено критерії оцінювання методів збору вимог, розроблено функцію вигоди для задачі вибору методів збору вимог для малих ІТ-проєктів, розроблено методологію застосування функції вигоди при виборі методів збору вимог, а також експериментально перевірено отримані результати вирішення задачі вибору методів збору вимог для певного ІТ-проєкту. Робота виконана згідно вимог методичних вказівок [31] та оформлена згідно з стандартами [32, 33].

Результати дослідження були представлені на конференції [34], а також обговорені на Молодіжному форумі [35].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tiwari S., Rathore S. S., Gupta A. Selecting requirement elicitation techniques for software projects. 2012 CSI Sixth International Conference on Software Engineering (CONSEG), Indore, Madhay Pradesh, India, 5–7 September 2012. 2012. URL: https://doi.org/10.1109/conseg.2012.6349486 (дата звернення: 22.05.2024).

2. Glintz M. CPRE Online Glossary - CPRE - IREB – International Requirements Engineering Board. International Requirements Engineering Board. URL: https://www.ireb.org/en/cpre/glossary/ (дата звернення: 21.04.2024).

3. Kovitz B. Hidden skills that support phased and agile requirements engineering. Requirements Engineering. 2003. Vol. 8, no. 2. P. 135–141. URL: https://doi.org/10.1007/s00766-002-0162-9 (дата звернення: 21.04.2024).

4. Communication patterns of agile requirements engineering / Abdullah et N. N. B. al. the 1st Workshop, Lancaster, United Kingdom, 26 July 2011. New York, New York, USA, 2011. URL: https://doi.org/10.1145/2068783.2068784 (дата звернення: 27.05.2024).

5. Cao L., Ramesh B. Agile Requirements Engineering Practices: An Empirical Study. IEEE Software. 2008. Vol. 25, no. 1. P. 60–67. URL: https://doi.org/10.1109/ms.2008.1 (дата звернення: 21.05.2024).

6. Requirement Engineering Challenges in Agile Software Development / A. Rasheed et al. Mathematical Problems in Engineering. 2021. Vol. 2021. P. 1–18. URL: https://doi.org/10.1155/2021/6696695 (дата звернення: 21.04.2024).

7. Pohl K. Requirements engineering fundamentals: A study guide for the certified professional for requirements engineering exam, foundation level, IREB compliant. 2nd ed. 2015. 163 p.

8. A systematic literature review of requirements engineering education / M. Daun et al. Requirements Engineering. 2022. URL: https://doi.org/10.1007/s00766-022-00381-9 (дата звернення: 21.04.2024).

9. Дужак І. О. «РОЗУМНИЙ БУДИНОК». Automation Technological and Business - Processes. 2014. Т. 13, № 13-14. URL: https://doi.org/10.15673/2312-3125.13-14/2010.32920 (дата звернення: 21.04.2024).

10. Моделі, методи та інформаційна технологія розробки архітектури складних інформаційних систем на основі функціональних вимог. URL: https://nure.ua/yevlanov-maksim-viktorovich (дата звернення: 22.04.2024).

11. Requirement Gathering Methods. University of Missouri–St. Louis. URL: https://www.umsl.edu/~sauterv/analysis/F2015/Requirement%20Gathering%20Methods.html.htm (дата звернення: 21.04.2024).

12. Requirement gathering questions: Using interviews - PM Majik. PM Majik. URL: https://www.pmmajik.com/requirement-gathering-questions-using-interviews/ (дата звернення: 21.04.2024).

13. Kawulich, Barbara B. (2005). Participant Observation as a Data Collection Method [81 paragraphs].Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 6(2), Art. 43, http://nbn-resolving.de/urn:nbn:de:0114-fqs0502430.

14. Observation: Definition, Types & Research | StudySmarter. StudySmarter UK. URL: https://www.studysmarter.co.uk/explanations/social-studies/theories-and-methods/observation/#:~:text=What%20are%20the%204%20types,covert%20observation,%20and%20overt%20observation. (дата звернення: 21.04.2024).

15. Project Requirement Gathering: Prototyping - PM Majik. PM Majik. URL: https://www.pmmajik.com/project-requirement-gathering-prototyping/ (дата звернення: 21.04.2024).

16. Wiegers K. E., Beatty J. Software Requirements (3rd Edition) (Developer Best Practices). Microsoft Press, 2013. 672 p.

17. IIBA. A Guide to the Business Analysis Body of Knowledge. International Institute of Business Analysis, 2015. 512 p.

18. Awati R. What is joint application development (JAD)? | Definition from TechTarget. Software Quality. URL: https://www.techtarget.com/searchsoftwarequality/definition/JAD#:~:text=Joint%20application%20development,%20frequently%20shortened,collaborative%20workshops%20called%20JAD%20sessions. (дата звернення: 21.04.2024).

19. Requirements Workshop Technique: Exploring the Power of Collaboration – Business Analyst Learnings. Business Analyst Learnings. URL: https://www.businessanalystlearnings.com/ba-techniques/2013/2/28/requirements-workshop-technique-exploring-the-power-of-collaboration (дата звернення: 21.04.2024).

20. Gottesdiener E. Requirements by Collaboration: Workshops for Defining Needs. Addison-Wesley Professional, 2002. 368 p.

21. Project Requirement Gathering: Surveys and Questionnaires - PM Majik. PM Majik. URL: https://www.pmmajik.com/project-requirement-gathering-surveys-and-questionnaires/ (дата звернення: 21.04.2024).

22. Institute P. M. PMBOK Guide: The Project Management Body of Knowledge. Booksmith Publishing LLC, 2021. 760 p.

23. Bourque P., Society I. C., Fairley R. E. Guide to the Software Engineering Body of Knowledge: Version 3.0. IEEE Computer Society Press, 2014. 346 p.

24. Price J. R. Write a Use Case: Gathering Requirements that Users Understand. The Communication Circle, 2020. 232 p.

25. Kulak D., Guiney E. Use Cases: Requirements in Context, Second Edition. 2nd ed. Addison-Wesley Professional, 2003. 272 p.

26. Project Requirement Gathering: Focus Groups - PM Majik. PM Majik. URL: https://www.pmmajik.com/project-requirement-gathering-focus-groups/ (дата звернення: 21.04.2024).

27. Управління ІТ-проєктами: Загальні питання теорії управління ІТ-проєктами (конспект лекцій) Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / уклад.: Л. М. Добровська, О. С. Коваленко, О. А. Аверьянова; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 3,67 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2022 – 284 с.

28. Основи управління ІТ проектами [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки»/ КПІ ім. Ігоря Сікорського ; уклад.: В. О. Кузьміних, Р. А. Тараненко. – Електронні текстові дані (1 файл:1,998 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 75 с.

29. Dou. Зарплати. URL: https://jobs.dou.ua/salaries/?period=2023-12&position=Middle%20SE (дата звернення: 22.05.2024).

30. A Refresher on A/B Testing. URL: https://hbr.org/2017/06/a-refresher-on-ab-testing (дата звернення: 22.05.2024).

31. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи другого (магістерського) рівня вищої освіти за освітньо-науковою програмою «Управління проектами в галузі інформаційних технологій» / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2024. – 24 с.

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

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

34. Панфьорова І. Ю., Степченко А. О. дослідження та удосконалення технік типового процесу збирання вимог в ІТ-проєкті// Scientific progress: innovations, achievements and prospects. Proceedings of the 9th International scientific

and practical conference. MDPC Publishing. Munich, Germany. 2023. Pp. 173-176. URL: https://sci-conf.com.ua/ix-mizhnarodna-naukovo-praktichnakonferentsiya-scientific-progress-innovations-achievements-and-prospects-29-31-05- 2023-myunhen-nimechchina-arhiv/.

35. Степченко А.О. Дослідження методів збирання вимог в ІТ-проєктах.// Радіоелектроніка та молодь у XXI столітті. Т. 6 : Конференція "Інформаційні інтелектуальні системи" : матеріали 28-го Міжнар. молодіж. форуму, 16–18 квіт. 2024 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків: ХНУРЕ, 2024. – 958 с.