

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра системного аналізу та обчислювальної математики

Пояснювальна записка

до дипломного проєкту (роботи)

магістра

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА РОЗРОБКА СИСТЕМИ
ОЦІНЮВАННЯ ВАРТОСТІ ІТ-ПРОЄКТІВ

Виконав: студент 2 курсу, групи КНТ-812м
спеціальності _____

124 Системний аналіз

(код і найменування спеціальності)

Освітня програма (спеціалізація)

«Інтелектуальні технології та

прийняття рішень в складних системах»

ЩЕРБАКС. О.

(ПРИЗВИЩЕ та ініціали)

Керівник ТЕРЕЩЕНКОЕ.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЛОЗОВСЬКА Л. І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій

Кафедра «Системний аналіз та обчислювальна математика»

Ступінь вищої освіти магістерський

Спеціальність 124 Системний аналіз
(код і найменування)

Освітня програма (спеціалізація) «Інтелектуальні технології та прийняття рішень в складних системах»
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

В. о. зав. кафедри Терещенко Е.В.

“ _____ ” _____ 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ЩЕРБАКА Семена Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та розробка системи оцінювання вартості ІТ-проєктів

керівник проєкту (роботи) к. ф. м., доцент, ТЕРЕЩЕНКО Еліна Валентинівна
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від “06” грудня 2023 року № 492

2. Строк подання студентом проєкту (роботи) 20 грудня 2023 року

3. Вихідні дані до проєкту (роботи) вимоги до програмного продукту від замовника

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити

1) Етапи та методики формування оцінок затрат часу розробників програмних систем;

2) Розробка методики оцінки розробки програмних систем;

3) Розробка комп'ютерної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	ТЕРЕЩЕНКО Е. В., в. о. зав. каф.		
нормоконтроль	ШИРОКОРАД Д.В., доцент		

7. Дата видачі завдання 01.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Проведення опису визначення оцінки розробки ІТ-проєктів	01.10.2023 р.	
2	Визначення етапи формування оцінки	10.10.2023 р.	
3	Проведення аналізу вимог до проєкту та їх класифікація	20.11.2023 р.	
4	Проведення аналізу методик формування оцінок затрат часу розробників	01.11.2023 р.	
5	Проведення класифікації методик та їх порівняльний аналіз	05.11.2023 р.	
6	Розробка методики розрахунку попередньої вартості розробки ІТ-проєктів	15.11.2023 р.	
7	Дослідження методики на 2-х проєктах	25.11.2023 р.	
8	Розробка комп'ютерну систему для розрахунку попередньої вартості розробки ІТ-проєктів	01.12.2023 р.	
9	Дослідження методики з використанням комп'ютерної системи на 2-х проєктах	05.12.2023 р.	
10	Оформлення отриманих результатів у ПЗ	10.12.2023 р.	

Студент



(підпис)

Семен ШЕРБАК
(ім'я, ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ **Еліна ТЕРЕЩЕНКО**
(підпис) (ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

ПЗ: 85 с., 26 рис., 16 табл., 17 джерел, 1 додаток.

FUNCTION POINT, PERT, POKER ESTIMATION, ВИМОГИ, ЕКСПЕРТНА ОЦІНКА, ЖИТТЄВИЙ ЦИКЛ, ОЦІНКА, МЕТОДИКИ ФОРМУВАННЯ ОЦІНОК, МЕТОДОЛОГІЯ МОНТЕ-КАРЛО, ОЦІНКА ПО АНАЛОГІЇ, РИЗИКИ, СВІТЛОФОРНА МАТРИЦЯ, КОМП'ЮТЕРНА СИСТЕМА

Метою даного дипломного проєкту є дослідження універсальної методики розрахунку попередньої вартості ІТ-проєктів та розробка системи оцінювання вартості ІТ-проєктів.

Об'єктом розробки є методика розрахунку початкової вартості ІТ-проєктів.

Пояснювальна записка містить три розділи.

У розділі 1 подано визначення оцінки вартості розробки ІТ-проєктів, етапи формування оцінки, аналіз вимоги до ІТ-проєкту та класифікація вимог, аналіз методів оцінки часу, витраченого на розробку, класифікацію та порівняльний аналіз методів оцінювання вартості ІТ-проєктів.

У розділі 2 розроблено та представлено у вигляді алгоритму методику розрахунку попередньої вартості розробки ІТ-проєктів. Описано тестування розробленої методики на двох проєктах.

У розділі 3 розроблено комп'ютерну систему для розрахунку початкової вартості ІТ-проєктів, яка реалізує методику розрахунку кошторисів ІТ-проєктів за аналогією.

ЗМІСТ

Завдання	2
Реферат	4
Вступ.....	7
1 Етапи та методики формування оцінок затрат часу розробників ІТ-проєктів	9
1.1 Визначення оцінки розробки ІТ-проєктів.....	9
1.1.1 Феномен «омана планування»	9
1.1.2 Імовірна характеристика оцінки	11
1.2 Етапи формування оцінки	13
1.2.1 Аналіз вимог до ІТ-проєктів	13
1.2.2 Визначення оцінки витрат часу розробників ІТ-проєктів	18
1.2.3 Залежність оцінки від ризиків ІТ-проєктів.....	18
1.2.4 Аналіз результатів формування оцінки	25
1.3 Аналіз існуючих методик формування оцінок.....	25
1.3.1 Експертна оцінка	27
1.3.2 Оцінка за аналогією	26
1.3.3 Методика оцінки «зверху вниз»	27
1.3.4 Методика оцінки «зверху вниз»	27
1.3.5 Методика оцінки функціональних точок.....	27
1.3.6 Триточкова оцінка або методика PERT	30
1.3.7 Методика Pokerestimate	30
1.3.8 Методологія Монте-Карло	32
1.3.9 Оціночні програми	32
1.4 Класифікація методик.....	35
1.5 Порівняльний аналіз існуючих методик.....	37
2 Розробка методики оцінки розробки ІТ-проєктів.....	40
2.1 Алгоритм методики оцінки розробки ІТ-проєктів.....	40
2.2 Дослідження методики оцінки розробки ІТ-проєктів	51

	6
3 Розробка комп'ютерної системи	61
3.1 Архітектура комп'ютерної системи	61
3.2 Проєктування системи	63
3.3 Розробка системи.....	64
3.4 Аналіз розробленої системи	66
Висновки	75
Перелік посилань.....	78
Додаток А. Результати дослідження методики.....	81

ВСТУП

ІТ-індустрія України є однією з найбільших галузей промисловості країни, а українські ІТ є ключовим компонентом її майбутнього економічного успіху. Наразі експорт послуг та розробки програмного забезпечення з України становить близько 4 мільярдів доларів США щорічно.

Є два типи ІТ-компаній, які спеціалізуються розробкою програмного забезпечення: аутсорс і продуктова.

Продуктові компанії - це компанії, які займаються розробкою власного програмного забезпечення. Аутсорсингові компанії - це ті, що надають власні ресурси на зовнішній ринок.

В аутсорсингових компаніях перший етап проєкту передбачає контакт між замовником і виконавцем, де замовник встановлює вимоги до програмного забезпечення. На основі наданих вимог виконавець повинен визначити, скільки коштуватиме замовнику розробка цього програмного продукту. Під час процесу оцінки можуть бути проведені додаткові інтерв'ю з замовником для уточнення вимог. Після цього слід прийняти рішення:

- а) вимоги до ІТ-проєкту, що визначають методику ведення проєкту;
- б) вимоги до програмного продукту, а саме:
 - 1) функціональні вимоги (описують поведінку системи за різних умов);
 - 2) нефункціональні вимоги, (описують якість продукту та додаткові вимоги, тобто функції, які не відображаються користувачам).

Обсяг робіт для ІТ-проєкту найчастіше формалізується у вигляді класичного "залізного трикутника" вартості, функціональності та часу. Як насправді ставляться до цих понять замовники та підрядники? Замовники зацікавлені в тому, щоб розробити якомога більше функцій, якомога швидше, за мінімально можливою вартістю. Розробники зацікавлені в максимальному бюджеті від замовника і якомога меншій функціональності, оскільки вони зацікавлені в прибутковості. Як наслідок, обидві сторони повинні досягти певної домовленості

щодо умов ІТ-проєкту ще до початку його реалізації. Виникає питання, як розробник може якомога точніше оцінити витрати, необхідні для успішної реалізації ІТ-проєкту, і в той же час забезпечити прибуток і не втратити клієнтів, тобто гарантувати, що замовник погодиться на умови розробника. Для цього розробники повинні якомога точніше оцінювати розвиток ІТ-проєкту.

У контексті кошторису програмних проєктів - це прогноз того, скільки ресурсів і часу знадобиться для розробки проєкту. Оцінки безпосередньо залежать від наступних ключових факторів

- вимог до програмного продукту та проєкту в цілому;
- життєвого циклу ІТ-проєкту;
- ризиків пов'язаних з веденням ІТ-проєкту.

Вимоги диктуються безпосередньо підрядником. Життєвий цикл описує, як відбуватиметься комунікація між учасниками проєкту, і вказує на час, необхідний для реалізації ІТ-проєкту. Наприкінці процесу оцінки, коли отримано попередній кошторис, необхідно виявити ризики, які можуть вплинути на проєкт, і врахувати їх при формуванні остаточного кошторису. Сторона-виконавець переводить кошторис у певну вартість, залежно від того, скільки часу окремі учасники проєкту планують витратити на реалізацію ІТ-проєкту.

Незважаючи на те, що методології управління проєктами та способи оплати за роботу різняться, часто саме на фінальних етапах проєкту виникають певні проблеми та непорозуміння. Щоб уникнути цього, важливо правильно розрахувати вартість розробки ІТ-проєкту до підписання контракту між сторонами, але це не точна наука, оскільки існує багато невизначеностей, таких як якість вимог замовника, кваліфікація і людський фактор співробітників з боку виконавця і ризики, пов'язані з ІТ-проєктом. Це робить розробку універсальної комп'ютерної системи для розрахунку попередньої вартості розробки ІТ-проєкту проблематичною.

1 ЕТАПИ ТА МЕТОДИКИ ФОРМУВАННЯ ОЦІНОК ЗАТРАТ ЧАСУ РОЗРОБНИКІВ ІТ-ПРОЄКТІВ

1.1 Визначення оцінки розробки ІТ-проектів

Оцінка розробки ІТ-проекту визначає ресурси, витрати і час, необхідні для створення певної системи або програмного продукту. Кошторис базується на таких показниках:

- рівень і якість деталізації вимог;
- фаховий досвід роботи учасників ІТ-проекту;
- ризики веденням ІТ-проекту.

1.1.1 Феномен «омана планування»

Найдорожчим ресурсом у процесі розробки ІТ-проекту є людино-години. Проблемою при оцінці людино-годин є явище "омана планування".

"Омана планування" - це явище, при якому прогнози щодо часу, необхідного для виконання завдання в майбутньому, мають оптимістичний ухил. Це явище відбувається незалежно від того, чи знає людина, що подібні завдання в минулому займали більше часу, ніж планувалося. Це упередження впливає лише на власну оцінку завдання, а коли зовнішній спостерігач прогнозує час, необхідний для виконання завдання, людина демонструє песимістичне упередження і переоцінює необхідний час [1].

«Омана планування» передбачає, що прогнози часу, необхідного для виконання поточних завдань, є більш оптимістичними, ніж уявлення про час, необхідний для виконання подібних завдань у минулому, і, як наслідок, більш оптимістичними, ніж фактичний час, необхідний для виконання поточних завдань. У 2003 році Ловалло і Канеман запропонували розширене визначення тенденції недооцінювати час, витрати та ризики майбутніх дій, переоцінюючи при цьому вигоди від таких дій. Згідно з цим визначенням, помилки планування призводять не лише до перевитрат часу, але й до перевитрат коштів та недосягнення запланованих результатів [1].

У дослідженні 1994 року 37 аспірантів-психологів попросили оцінити, скільки днів їм знадобиться для завершення дисертації. Середній показник склав 33,9 днів. Вони також оцінили, скільки часу їм знадобиться, "якщо все піде добре" (в середньому 27,4 дня) і "якщо все піде погано" (в середньому 48,6 дня). Реальний час, необхідний для виконання роботи, в середньому склав 55,5 днів, і лише близько 30% студентів завершили роботу в межах прогнозованого терміну [1].

В іншому дослідженні студентів попросили оцінити, коли будуть завершені їхні навчальні проєкти. Тобто вони оцінювали завершення з ймовірністю 50%, 75% і 99%, а потім відстежували статус закінчення. Результати виявилися наступними:

- проєкт завершили до часу, який зазначили з 50% ймовірністю - 13% студентів;
- проєкт завершили до часу, який зазначили з 75% ймовірністю - 19% студентів;
- проєкт завершили до часу, який зазначили з 99% ймовірністю - 45% студентів;
- решта не вписалася і в цей час [1].

Наступні методи зменшення похибки феномену «омани планування» пропонують Амос Тверські та Деніел Канеман:

- плани впровадження;
- прогнозування за аналогією;
- ефект сегментації.

Плани впровадження - це план, який визначає, коли, де і як буде здійснюватися та чи інша дія. Різні експерименти показали, що наявність плану виконання дозволяє людям краще зрозуміти завдання в цілому і передбачити всі можливі результати. На початкових етапах такого планування прогнози щодо часу виконання завдання є навіть більш оптимістичними. Однак вважається, що процес планування виконання "безпосередньо задіює силу волі", тобто сприяє більшій залученості людей до виконання завдання. Ті, хто склав план виконання

під час експерименту, швидше впоралися із завданням, їх рідше переривали і вони були менш оптимістичними у своїх подальших прогнозах, порівняно з тими, хто не склав план виконання. Також було виявлено, що менша кількість перерв зменшує оптимістичне упередження [1,3].

Прогнозування за аналогією ґрунтується на правильному кількісному аналізі фактичних наслідків минулої поведінки, подібної до запланованої [4].

Ефект сегментації проявляється в тому, що час, виділений на завдання в цілому, значно менший, ніж сума часу, виділеного на окремі підзадачі цієї задачі. У 2008 року в дослідженні Форсайта було перевірено, чи можна використовувати цей ефект для зменшення помилок планування, і три експерименти підтвердили його наявність. Однак ефект сегментації вимагає значних когнітивних ресурсів і тому не дуже добре підходить для повсякденних ситуацій [1,2].

1.1.2 Імовірна характеристика оцінки

Оцінки завжди носять імовірнісний характер. Статистика показує, що близько 3/4 ІТ-проектів перевищують свої початкові оцінки. Ймовірність того, що будь-який ІТ-проект буде завершено в рамках бюджету і вчасно, не є стовідсотковою. Ця різниця повинна бути мінімальною і є найважливішим питанням у сфері оцінки ІТ-проектів.

Зазвичай оцінки ІТ-проектів подаються у вигляді простих чисел, наприклад, що ІТ-проект триватиме 14 тижнів. Такі оцінки називаються точковими і є досить неточними, оскільки не містять інформації про ймовірності, пов'язані з точковими оцінками. Оцінка припускає ситуацію, показану на рисунку 1.1, тобто, що існує лише один можливий результат, тобто ця точка.

Точна оцінка враховує, що ІТ-проекти схильні до ряду невизначеностей. У сукупності ці невизначеності призводять до того, що результати ІТ-проекту мають розподіл ймовірності, причому деякі результати є більш ймовірними, ніж інші, а найбільш ймовірні результати сконцентровані в середині розподілу (рис. 1.2).

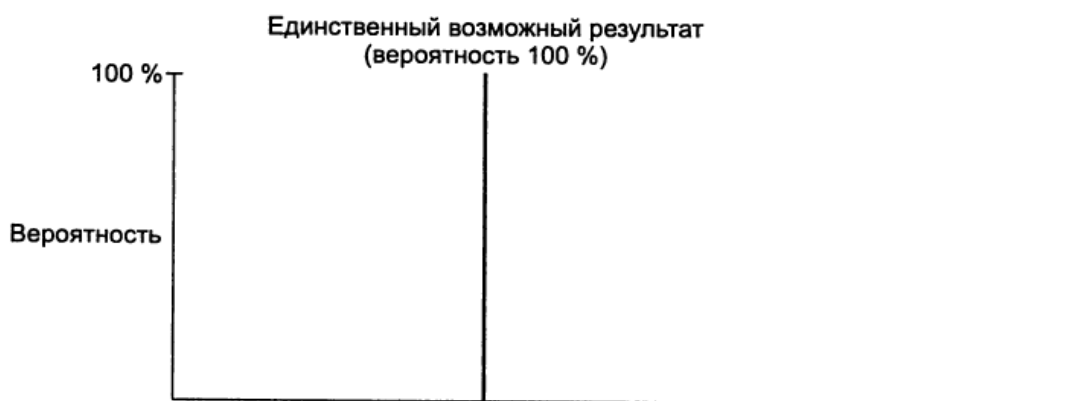


Рисунок 1.1 - Точкова оцінка термінів/витрат/робочого часу



Рисунок 1.2 - Нормальний розподіл термінів/витрат/робочого часу

Кожна точка на кривій представляє ймовірність того, що ІТ-проект буде завершено до точної дати (або точної суми). Площа під кривою представляє загальну стовідсоткову ймовірність. Цей тип розподілу ймовірності враховує можливий діапазон результатів. Однак припущення, що розподіл результатів симетричний відносно середнього значення, є неправильним. Якість виконання ІТ-проекту є обмеженою. Це означає, що лівий кінець розподілу є усіченим, а не нескінченно спадаючим, як у нормальному розподілі. Крива ймовірності йде дуже далеко вправо, тому що результат "успішного" ІТ-проекту обмежений, в той час як можливість "провалу" нескінченна. На рисунку 1.3 показано розподіл ймовірностей результатів ІТ-проекту.



Рисунок 1.3 - Уточнення уявлення можливих результатів ІТ-проектів

1.2 Етапи формування оцінки

Процес оцінювання складається з п'яти основних етапів:

- аналіз вимог до ІТ-проекту;
- оцінювання витрат часу на розробку та розміру продукту;
- оцінювання ризиків, що можуть виникати під час ведення ІТ-проекту;
- аналіз графіка розробки ІТ-проекту;
- формування вартості ІТ-проекту.

1.2.1 Аналіз вимог до ІТ-проектів

Найпоширеніша проблема з точністю оцінки - це вимоги. Вимоги - це опис характеристик, яким має задовольняти програмний продукт, що розробляється. Вони описують поведінку програмного продукту, характеристики або атрибути. Вони діють як обмеження в процесі розробки. Це визначення враховує різні типи інформації, які в сукупності називаються вимогами [5]. Вимоги охоплюють як бачення замовника та зовнішню поведінку системи (функціональні вимоги), так і погляд розробника на деякі внутрішні характеристики (нефункціональні вимоги). Вимоги включають як поведінку системи за певних умов, так і характеристики, які роблять систему корисною і повністю задовольняють кінцевого користувача.

Якщо у замовника є багато невизначеності щодо вимог і він має нечітке уявлення про ймовірну функціональність, у підрядника виникнуть проблеми при оцінці обсягу робіт, оскільки незрозуміло, яку саме роботу потрібно оцінювати. В такому випадку оцінки можуть призвести до великого діапазону часу роботи розробника над ІТ-проектом, наприклад, від 20 до 120 годин. В такому випадку мінімальна і максимальна оцінки можуть відрізнятись на 600%, що є дуже великим розкидом, і оцінка може вважатися неточною. Якщо замовник чітко розуміє, чого він хоче, розробник може зробити більш точну оцінку. У цьому випадку різниця між мінімальним і максимальним кошторисом становить менше 20%. Отже, чим точніше сформульовані вимоги до програмного забезпечення, що розробляється, тим більша ймовірність того, що буде отримано точний кошторис. Тому при підготовці кошторису необхідно уточнити вимоги у замовника за допомогою анкетування та інтерв'ю із замовником. Якщо це неможливо, розробники не мають іншого вибору, окрім як базувати свої оцінки на припущеннях щодо майбутньої функціональності.

Також необхідно зафіксувати вимоги та обмеження, які можуть встановити часові рамки розробки, тобто терміни та можливий бюджет від замовника. Ці обмеження безпосередньо впливають на вибір методів управління проектом і на склад команди, яка бере участь у проекті. Детально про аналіз вимог, категоризацію вимог та методи визначення вимог і їхньої якості розглянуто в 2-му розділі.

1.2.1.1 Класифікація вимог

Вимоги можна поділити на три рівні:

- 1-ий рівень - бізнес-вимоги, бізнес-правила;
- 2-ий рівень - вимоги користувача;
- 3-ій рівень - функціональні вимоги, нефункціональні вимоги, перехідні вимоги.

1-ий рівень описує бізнес-вимоги та правила щодо того, навіщо замовнику потрібна такий програмний продукт, тобто цілі, яких організація має намір досягти за допомогою його. Його основним змістом є бізнес-цілі організації або

замовника, який замовляє програмний продукт. Розуміння бізнес-вимог допомагає розробникам чітко інтерпретувати вимоги замовника і робити правильні припущення щодо вимог до IT-проєкту.

Бізнес-вимоги описують основні цілі розробки IT-проєкту з точки зору бізнесу.

Бізнес-правила включають корпоративну політику, державні норми, галузеві стандарти та обчислювальні алгоритми. Бізнес-правила - це не прямі вимоги до програмної системи, а скоріше обмеження для програмної системи, що визначають, яку функціональність повинна мати програмна система, що дотримується правил.

2-ий рівень описує цілі та завдання, які користувачі повинні вирішити за допомогою програмного продукту. Іншими словами, продукт, що розробляється, повинен бути корисним для користувача. Область вимог користувача також включає опис атрибутів і характеристик програмного продукту, важливих для задоволення потреб користувача. Розуміння вимог користувача сприяє кращому розумінню того, якою має бути програмна система, і визначенню пріоритетності завдань розробки. Все це разом збільшує ймовірність успішної реалізації IT-проєкту.

3-ій рівень визначає, як програмний продукт повинен поводитися в конкретних умовах. Він визначає, що підрядники повинні створити, щоб користувачі могли виконувати задачі (користувацькі вимоги) в рамках бізнес-вимог.

Функціональні вимоги описують функції, які повинно виконувати розроблена програмна система. Функціональні вимоги є похідними від вимог користувача [6].

Нефункціональні вимоги - вимоги визначають характеристики, які повинна мати програмна система, або обмеження, яких необхідно дотримуватися, але які не пов'язані з роботою програмної системи.

Перехідні вимоги - визначають умови, які повинні бути виконані для того, щоб впровадити новий програмний продукт в організації. Це не вимоги для того,

щоб програмна система працювала, а для того, щоб вона була запущена у виробництво.

Всебічне розуміння вимог на всіх рівнях забезпечує найкраще розуміння програмної системи, що розробляється, і передбачає всі можливі небезпеки.

1.2.1.2 Способи визначення вимог

Виявлення вимог - це спільний аналітичний процес, що включає такі види діяльності, як збір, ідентифікація, виявлення та визначення вимог. Результатом етапу виявлення вимог є узгоджене уявлення про потреби всіх зацікавлених сторін IT-проєкту. Чим краще розробники розуміють ці потреби, тим легше запропонувати альтернативні способи їх задоволення.

Методи виявлення вимог можливо поділити на:

- колективні - в яких беруть участь зацікавлені особи;
- незалежні - коли ви самостійно виявляєте інформацію.

Колективні методи зосереджені на визначенні вимог користувача та бізнесу. Незалежний метод доповнює вимоги, надані користувачем, і визначає функції, про які кінцевий користувач може не знати.

Найпоширенішими методами виявлення вимог є інтерв'ю, семінари, фокус-групи, спостереження, анкетування, аналіз системного інтерфейсу, аналіз користувацького інтерфейсу та аналіз документів.

Інтерв'ю є традиційним джерелом інформації про вимоги як до серійних продуктів, так і до інформаційних систем у будь-якій методології розробки програмного забезпечення.

Семінари - спеціально організовані зустрічі з багатьма зацікавленими сторонами та формальними ролями, такими як організатор або секретар.

Фокус-групи - це репрезентативні групи користувачів, які зустрічаються в рамках керованої справи зі з'ясування вимог, щоб надати інформацію та ідеї щодо функціональних та якісних вимог.

Спостереження - дослідження робочих процесів користувачів у робочому середовищі.

Опитувальні листи - це один із способів опитування великих груп користувачів і вивчення їхніх потреб.

Аналіз системного інтерфейсу є незалежним методом визначення вимог і передбачає аналіз систем, з якими взаємодіє система, що розробляється.

Аналіз інтерфейсу користувача - це незалежний метод аналізу існуючих програмних систем для визначення функціональних вимог користувача.

Аналіз документації вивчає всю доступну документацію та визначає можливі вимоги до програмного продукту.

1.2.1.3 Визначення якості вимог

Різними джерелами по-різному визначені характеристики якості вимог до програмного продукту і вони є загальновизнаними (таблиця 1.1) [5].

Таблиця 1.1 - Характеристики якості вимог

Характеристики	Опис
Одиничність	Вимога описує одну і тільки одну річ
Завершеність	Вимога повністю визначена в одному місці і вся необхідна інформація присутня
Послідовність	Вимога не суперечить іншим вимогам і повністю відповідає зовнішньої документації
Атомарність	Вимога «атомарно». Тобто вона не може бути розбита на ряд більш детальних вимог без втрати завершеності
Відстеження	Вимога повністю або частково відповідає діловим потребам як заявлено зацікавленими особами і документовано
Актуальність	Вимога не застаріла з плином часу
Здійснимість	Вимога може бути реалізована в межах проекту
Недвозначність	Вимога коротко визначена без звернення до технічної термінології, скорочень та іншим прихованим формулюванням. Воно виражає об'єктивні факти, не суб'єктивні думки. Можлива одна і тільки одна інтерпретація. Визначення не містить нечітких фраз. Використання негативних тверджень і складових тверджень заборонено
Обов'язковість	Вимога представляє певну зацікавленою особою характеристику, відсутність якої призведе до неповноцінності рішення, яка не може бути проігнорована. Необов'язкова вимога - протиріччя самому поняттю вимоги
Верифікованість	Реалізованість вимоги може бути визначена через один з чотирьох можливих методів: огляд, демонстрація, тест чи аналіз

1.2.2 Визначення оцінки витрат часу учасників розробки ІТ-проектів

Після етапу аналізу вимог формується перелік функціоналу, який потрібно оцінити кількісно. Іншими словами, необхідно розуміти, скільки часу фахівці витрачають на виконання поставлених перед ними завдань. Ця величина виражається в людино-місяцях, людино-тижнях або людино-годинах. Для отримання найбільш точних результатів найчастіше використовують значення людино-години.

Спочатку оцінюють роботу розробників, на яких припадає більша частина ресурсних витрат. Після аналізу вимог складається перелік функцій, які необхідно розробити, визначаються розробники, які можуть бути залучені до цього ІТ-проекту, та оцінюється кількість людино-годин, необхідних для розробки. Крім розробників, які працюють над ІТ-проектом, до оцінки людино-годин розробки можуть залучатися інші експерти та інші розробники.

Окрім розробників, за необхідності оцінюється робота інших учасників ІТ-проекту (дизайнерів, тестувальників та менеджерів ІТ-проекту). Однак у випадку з менеджерами ІТ-проектів це значною мірою залежить від обраного підходу до управління ІТ-проектом.

У третьому розділі наведено аналіз того, як розробляти оцінки часу учасників ІТ-проекту.

1.2.3 Залежність оцінки від ризиків ІТ-проектів

Необхідно визначити ризики, які можуть негативно вплинути на процес розробки програмного забезпечення.

Ризик - це невизначені події або умови, які впливають на життєвий цикл ІТ-проекту. Іншими словами, кожен ризик має ймовірність і вплив, які разом визначають величину ризику. Наприклад, якщо над ІТ-проектом працює 100 співробітників, існує дуже висока ймовірність того, що один з них захворіє, але це матиме дуже малий вплив на ІТ-проект. З іншого боку, працівник може завершити роботу раніше запланованого терміну. Аналізуючи ці приклади, можна зрозуміти, що ризики можуть бути як позитивними, так і негативними. Позитивних ризиків часто набагато менше, ніж негативних, і немає сенсу зосереджуватися на

позитивних ризиках, оскільки розробка ІТ-проєкту спрощується лише тоді, коли вони є очевидними. Негативні ризики, з іншого боку, вимагають пильної уваги, оскільки вони можуть мати значний вплив на прибутковість ІТ-проєкту [7].

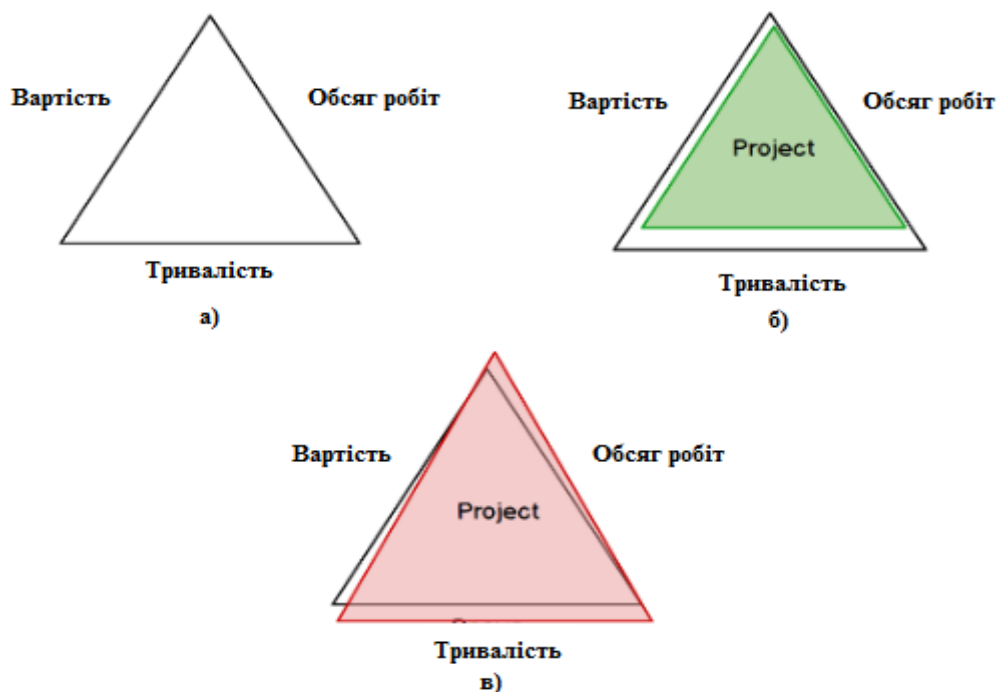


Рисунок 1.4 – а) «Залізний трикутник» ІТ-проєкту, б) позитивний результат, в) негативний результат

Ризики мають безпосередній вплив на «Залізний трикутник» ІТ-проєкту, тобто на обсяг, час і вартість ІТ-проєкту (рис. 1.4). На всіх етапах ІТ-проєкту значення цього трикутника можуть змінюватися, але вони завжди є постійними. Наприклад, на самому початку ініціації ІТ-проєкту, по закінченню з'ясування вимог із замовником, замовнику представляють вартість ІТ-проєкту, дату завершення ІТ-проєкту, обсяг робіт і якість робіт. Також на інших етапах, коли відбувається управління вимогами, створюється новий, але все ще постійний трикутник. Наприкінці життєвого циклу ІТ-проєкту обидві сторони задоволені, якщо всі показники знаходяться в межах залізного трикутника, тобто якщо замовник отримує програмний продукт в обіцяні терміни або раніше, за розумну ціну. Якщо ризик впливає на значення трикутника, тобто розробник зазнає збитків або замовник незадоволений результатом через терміни, вартість або якість

програмного продукту, обидві сторони зазнають втрат, що також впливає на репутацію. Тому ризик є дуже важливим елементом життєвого циклу ІТ-проєкту.

Часто визначаються лише негативні ризики. Це пов'язано з тим, що позитивні ризики приносять користь лише зацікавленим сторонам ІТ-проєкту, тоді як негативні ризики можуть підвищити рівень витрат і безпосередньо вплинути на прибуток розробника.

Після аналізу вимог до ІТ-проєкту та оцінки часу, витраченого учасниками ІТ-проєкту, слід скласти перелік ризиків, які можуть негативно вплинути на ІТ-проєкт. Після цього можна додати буферний час для розробників, залежно від ймовірності настання ризику та його впливу. Цей буферний час дозволяє включити негативні ризики у вартість ІТ-проєкту і дозволяє підряднику більш передбачувано розрахувати прибуток від ІТ-проєкту. Більше інформації про те, як визначаються, класифікуються та аналізуються ризики, наведено в 4-му розділі.

1.2.3.1 Класифікація ризиків

Існує велика кількість ризиків, і будь-який ризик можна класифікувати різними способами. Навіть досвідчені ризик-менеджери можуть ідентифікувати один і той самий ризик за різними класифікаціями [8]. В даному випадку класифікуємо ризики на основі їх виникнення:

а) ризики, які пов'язані з роботою з замовником:

- 1) невизначеність вимог;
- 2) зменшення з боку замовника бюджету на ІТ-проєкт;
- 3) тривалий час підтвердження макетів та дизайну;
- 4) нестабільне фінансування, тощо;

б) ризики, які пов'язані з комунікацією учасників ІТ-проєкту:

- 1) спілкування між розробниками;
- 2) координація робіт;
- 3) час для введення та залучення нових учасників ІТ-проєкту;
- 4) уточнення певних вимог, тощо;

в) технічні ризики:

- 1) невідповідність вимогам ІТ-проєкту ІТ-архітектури;

- 2) вибір неправильних стек технологій;
- 3) впровадження інших технологій, тощо;

г) ризики з боку певного учасника ІТ-проєкту:

- 1) відсутність достатньої кваліфікації;
- 2) необхідність здобуття нових навичок;
- 3) виникнення помилок;
- 4) хвороби, травми, незаплановані відпустки, непередбачені особисті обставини, звільнення, тощо;

обставини, звільнення, тощо;

д) зовнішні фактори (технічні, погодні, законодавчі умови, тощо).

1.2.3.2 Аналіз методик виявлення ризиків

Щоб управляти ризиками, їх потрібно спочатку ідентифікувати. Ідентифікація ризиків починається ще до початку ІТ-проєкту, на етапі початкової оцінки вартості програмного продукту. Ідентифікація ризиків - це ітеративний процес. Він відбувається протягом усього життєвого циклу програмного продукту під час ІТ-проєкту, аж до етапу, коли кінцевий програмний продукт впроваджується на стороні замовника. Дуже важливо виявити всі ці ризики на етапі початкової оцінки вартості програмного продукту. Виявлення якомога більшої кількості ризиків може захистити розробника від незапланованих витрат на ІТ-проєкт. Після того, як розробники врахували всі можливі ризики, вони можуть захистити себе, створивши певний буфер ризику.

Ідентифікацією ризиків займається проєктний менеджер або бізнес-аналітик на етапі початкової оцінки ІТ-проєкту. Є велика кількість методик для ідентифікації ризиків:

- мозковий штурм;
- Mind Mapping;
- аналіз документації;
- інтерв'ю;
- аналіз контрольних списків;
- метод аналогії;
- опитування експертів з великим досвідом роботи над ІТ-проєктами;

- метод Delphi;
- метод номінальних груп.

Мозковий штурм для спільного визначення потенційних ризиків, які необхідно оцінити.

Mind Mapping використовуються для визначення та категоризації потенційних ризиків і розуміння їх взаємозв'язків.

Аналіз документації передбачає аналіз проєктної документації, підготовленої до проведення експертизи. Аналізується якість плану, узгодженість плану, відповідність вимогам замовника, припущенням ІТ-проєкту, базовим планам щодо дотримання термінів, витрат та інших показників проєктного ризику.

Інтерв'ю проводяться для з'ясування вимог і невизначеностей, а також для того, щоб зрозуміти, які ризики розглядають зацікавлені сторони та різні фактори, що призводять до цих ризиків.

Контрольний список - це перелік ризиків, складений на основі інформації та знань, отриманих з попередніх подібних ІТ-проєктів.

Для ідентифікації ризиків метод аналогії використовує накопичені знання і плани з управління ризиками інших аналогічних ІТ-проєктів;

Метод Delphi схожий на метод мозкового штурму, але учасники не знають один одного. Фасилітатор використовує перелік запитань, щоб виявити ідеї, пов'язані з проєктними ризиками, і зібрати відповіді експертів. Потім відповіді експертів аналізуються, класифікуються і повертаються експертам для подальших коментарів. Після кількох ітерацій цього процесу отримують консенсус і список ризиків. Метод Delphi усуває страх тиску з боку колег або збентеження при висловленні ідей.

Метод номінальної групи дозволяє ідентифікувати ризики та ранжувати їх за ступенем важливості. У цьому методі формується група з 7-10 експертів. Кожен учасник перераховує ризики ІТ-проєкту індивідуально, без обговорення. Потім всі визначені ризики обговорюються спільно, і окремі ризики повторно перераховуються в порядку їх важливості.

В таблиці 1.2 представлені переваги та недоліки описаних вище методик.

Таблиця 1.2 – Порівняльний аналіз методик виявлення ризиків

Метод ідентифікації	Переваги	Недоліки
Мозковий штурм	Сприяє взаємодії членів групи. Швидкий. Недорогий	Може проявитися переважання однієї особистості. Можна зосереджуватися тільки в конкретних областях. Вимагає сильного ведучого. Для оцінки необхідно контролювати схильності групи
MindMapping	Не ресурсовитратний. Дозволяє впорядкувати і структурувати ризики	Високе завантаження аналітика. Можлива упереджена оцінка
Аналіз документації	Не ресурсовитратний. Дозволяє бачити явні ризики на основі вимог	Високе завантаження аналітика. Можливо упереджена оцінка
Інтерв'ю	Дозволяє максимально уникнути від невизначеності і явно визначити ризики	Вимагає прямого контакту замовника або представника з фірмою
Контрольні списки	Конкретний і упорядкований. Легко використовувати	Упередженість. Може не містити конкретних елементів для даного проекту
Метод аналогії	Використовує минулий досвід для виключення проблем в майбутньому. Подібні проекти містять багато схожих рис	Потребує багато часу. Легко отримати результати, що не підходять для даного випадку. Аналогія може бути некоректною
Опитування експертів	Використовується минулий досвід	Експерт може бути упередженим. Потребує багато часу
Метод Delphi	Немає домінування однієї особистості. Може проводитися дистанційно через електронну пошту. Виключається проблема ранньої оцінки. Вимагає участі кожного члена групи	Займає багато часу. Високе завантаження ведучого
Метод номінальних груп	Зменшується ефект домінуючої особистості. Забезпечує взаємодію учасників. Дає упорядкований список ризиків	Потребує багато часу. Високе завантаження ведучого

Основною проблемою в управлінні ризиками та прийнятті рішень щодо буферів ризиків є розмір переліку ризиків, отриманого на етапі ідентифікації. Це пов'язано з тим, що неможливо управляти всіма виявленими ризиками, що потребувало б величезних фінансових та людських ресурсів. Основним завданням якісного аналізу є розподіл ризиків на групи та визначення їх пріоритетності.

Для кожного ідентифікованого ризику оцінюється ймовірність настання та вплив на основі експертної оцінки. У деяких випадках ризики з явно низькою ймовірністю настання та впливу не включаються до рейтингу, але вносяться до переліку ризиків, що підлягають моніторингу в майбутньому.

Матриця ймовірності та наслідків або «Світлофорна» матриця є найбільш поширеним і широко використовуваним інструментом для визначення величини кожного ризику (табл. 1.3). Величина ризику є орієнтиром для реагування на ризик. Наприклад, ризики, розташовані в зоні високого ризику в матриці, вимагають превентивних операцій і проактивних стратегій реагування. Для загроз, розташованих у зоні низького ризику, превентивні операції можуть не знадобитися; скоріше за все, їх потрібно лише відстежувати. Важливо також розуміти, що ризики можуть змінювати свою величину і переміщатися вздовж цієї матриці протягом життєвого циклу продукту [9].

Таблиця 1.3 – «Світлофорна» матриця ризиків

Імовірність		Рівень збитку				
		1	2	3	4	5
		Несуттєві	Низькі	Середні	Суттєві	Катастрофічні
5	Вельми ймовірно					
4	Ймовірно					
3	Можливо					
2	Малоймовірно					
1	Дуже малоймовірно					

Результатом якісного аналізу є перелік ризиків та їх вагових коефіцієнтів.

На етапі кількісного аналізу ризику оцінюються кількісно (квантифікуються). Кількісний аналіз ризиків зазвичай проводиться для ризиків, які були кваліфіковані в результаті якісного аналізу. Під час кількісного аналізу аналізуються ризики з високою та середньою вагою. Забезпечуючи дуже невеликий буфер для груп ризиків з низькою вагою, кількісний аналіз може бути повністю абстрактним.

Кількісний аналіз ризиків є одним з найважливіших етапів в управлінні ризиками. Оскільки ризик має імовірнісний характер, він не може бути визначений на 100% кількісно за допомогою методики, але може бути максимально наближеним до точної оцінки. Загалом, методики можна розділити на два типи: експертна оцінка та аналіз статистичних даних. До першого типу належать мозковий штурм, метод Delphi та експертна оцінка. До другого - метод Монте-Карло та метод аналогії. Перший тип методик призводить до суб'єктивних та упереджених оцінок і вимагає значних ресурсів. Другий тип вимагає менше ресурсів, але потребує досить великої кількості статистичних даних.

1.2.4 Аналіз результатів формування оцінки

Основна частина витрат на розробку ІТ-проєкту - це заробітна плата працівників підрозділу-виконавця. Тому необхідно оцінити час, що витрачається на проєктні роботи в традиційних підрозділах. Оцінити ризики, пов'язані з розробкою програмного продукту, і визначити, скільки ресурсів потрібно для їх усунення. Спрогнозувати можливу тривалість життєвого циклу розробки та відредагувати час, необхідний для реалізації ІТ-проєкту відповідно до методики управління ІТ-проєктами.

Для отримання загальної оцінки необхідно також підрахувати ресурси, витрачені на розробку ІТ-проєкту, які не пов'язані із заробітною платою учасників ІТ-проєкту, та врахувати розмір маржі виконавчої компанії. Таким чином кінцева вартість розробки ІТ-проєкту складається із заробітної плати учасників ІТ-проєкту, витрат на інші ресурси та маржі компанії-виконавця.

1.3 Аналіз існуючих методик формування оцінок

Методологія формування оцінки означає, що можна визначити вартість розробки різних програмних систем на основі вхідних вимог від замовників,

отримуючи параметри, які в подальшому можуть бути переведені в грошовий еквівалент.

Існує велика кількість методик формування оцінки, а саме:

- експертна оцінка;
- оцінка за аналогією;
- зверху вниз;
- знизу вгору;
- функціональних точок (FunctionPoint);
- триточкова оцінка або PERT;
- покер оцінка (Pokerestimate);
- методологія Монте-Карло;
- використання програмного забезпечення SLIM, COCOMO.

1.3.1 Експертна оцінка

Підхід експертної оцінки передбачає залучення експертів у необхідних галузях для оцінки вартості ресурсів. Це може бути один або кілька експертів. Наприклад, у деяких випадках достатньо, щоб ІТ-проект оцінював один експерт, тоді як у випадку великих ІТ-проектів може знадобитися зібрати команду експертів, що складається з працівників з різних дисциплін, залучених до реалізації оцінюваного ІТ-проекту.

Експерти роблять власні припущення щодо оцінювання (строків і витрат). Потім вони повинні скласти оцінки для всіх типів працівників і в ході обговорення прийняти єдине рішення [10, 11].

Якість отриманої таким чином оцінки залежить від досвіду експертної групи та її розміру. Чим більший досвід мають експерти в необхідній предметній області і чим більша їх кількість, тим більша ймовірність того, що буде зроблена точна оцінка.

Перевагами експертної оцінки є висока точність оцінки, яка залежить від якості експертної команди. Недоліком цього підходу є великі витрати ресурсів на отримання оцінки.

1.3.2 Оцінка за аналогією

Оцінка за аналогією - це метод, який використовує базу історичних даних як основу для оцінки подібних ІТ-проектів. Цей метод дозволяє визначити кількість ресурсів, необхідних для виконання завдання, на основі історичних даних і відповідно до того, скільки ресурсів було потрібно для виконання аналогічних завдань. Історичні дані також можуть показати ризики та небезпеки, з якими стикалася команда, що дозволить їм витратити додатковий час або уникнути їх у майбутньому [3].

Будь-який співробітник компанії може виконати цей метод оцінки, але якість оцінки залежить не тільки від наявності історичних даних, але й від рівня кваліфікації та досвіду співробітника. Дуже складно знайти задачу, яка дозволяє провести дуже точну оцінку серед якомога більшої кількості схожих історичних даних. Особа, яка проводить оцінку, повинна розуміти відмінності між історичними даними та поточним завданням і редагувати оцінку відповідно до рівня цих відмінностей.

У випадках, коли є якісна база історичних даних досить і одної особи для отримання точної оцінки. Недоліком методики за аналогією є наявність якісної бази історичних даних і її наповнення. Перевага це висока точність оцінки та мала витрата ресурсів, але при наявності якісної історичної бази даних.

1.3.3 Методика оцінки «зверху вниз»

Методика оцінки "зверху вниз" (top down estimate) означає, що оцінки готуються на основі вимог дуже високого рівня. Ключовим моментом таких агрегованих експертних оцінок є те, що вони проводяться в узагальненому вигляді на основі досвіду виконання подібних завдань. Оцінювачами часто виступають або топ-менеджмент компанії, або експерти, які мають досвід виконання подібних завдань.

Методика оцінки вартості «зверху вниз» використовуються для оцінки витрат на ранніх стадіях ІТ-проекту, коли інформація про ІТ-проект ще дуже обмежена.

При формування оцінки методом "зверху вниз" оцінки менш точні, але час, витрачений на їх формування, мінімальний і практично не залежить від рівня деталізації вимог.

Перевага методики оцінки "зверху вниз" полягає в тому, що не залежить від рівня деталізації вимог та займає мало часу. Основний недолік це низька точності оцінки і погане розуміння про можливі ризики ІТ-проєкту [12].

1.3.4 Метод оцінки «знизу вгору»

При метод оцінки «знизу вгору» завдання розбиваються на менші, більш керовані частини, поки їх не можна буде оцінити вручну з необхідною точністю. Залежно від необхідної точності, завдання розбиваються на окремі завдання або об'єднуються в завдання більш високого рівня. Індивідуальні витрати на кожне завдання потім додаються разом для отримання повного кошторису ІТ-проєкту [13].

Рівень деталізації, необхідний для декомпозиції завдання, повинен відповідати рівню деталізації вимог. Якщо рівень деталізації вимог недостатній, необхідно уточнити вимоги у замовника за допомогою опитувань та інтерв'ю або покластися на припущення оцінювача.

Ця методика найчастіше використовується на завершальних етапах ініціювання проєкту, коли більшість вимог чітко визначені і необхідно отримати найточніший остаточний кошторис.

Оцінка «знизу вгору» вимагає багато часу на підготовку декомпозиції завдання. Однак цей метод може забезпечити високу точність та повне розуміння усіх ризиків розробки і це є його основна перевага.

Недоліки - вимагає достатнього рівня деталізації вимог, а також багато часу на підготовку декомпозиції та оцінку завдання.

1.3.5 Методика оцінки функціональних точок

Методика оцінки функціональних точок (FunctionPoint) визначає та підраховує функції на рівні користувацького інтерфейсу на основі аналізу вимог до функцій, які необхідно розробити [14].

Перш за все, необхідно визначити межі системи (абстрактний інтерфейс між програмою та всіма її користувачами). Потім необхідно визначити функціональні точки і призначити вагові коефіцієнти відповідно до складності функціональних точок. Існують наступні типи функціональних точок:

- EI (external in puts) - зовнішні вхідні транзакції;
- EO (external out puts) - зовнішні вихідні транзакції;
- EQ (external in quiries) - зовнішні запити;
- ILF (internal logical file) - внутрішні логічні файли;
- EIF (external logical file) - зовнішні файли інтерфейсу даних користувача.

Зовнішні вхідні транзакції - елементарні операції з обробки даних та керуючої інформації, що надходять до системи ззовні. Слова, які ідентифікують зовнішні вхідні транзакції: makeinactive, add, modify, assign, allocate, record, associate, create, delete, import, remove, change, reset, reverse, set, undo, update, upload, withdraw.

Зовнішні вихідні транзакції - елементарні операції для генерації даних або керуючої інформації, яка виходить за межі системи, включаючи специфічну логіку для обробки або обчислення інформації з одного або декількох ILF. Слова, які ідентифікують зовнішні вихідні транзакції: adjust, report, export, print, generate, notify, summary.

Зовнішні запити - елементарні операції для отримання даних або керуючої інформації з ILF або EIF у відповідь на зовнішній запит. Слова, які ідентифікують зовнішні запити: extract, inquire, browse, display, list, enquire, picklist, view.

Внутрішні логічні файли, які користувач призначає логічно пов'язаним групам даних або блокам керуючої інформації, керованим всередині продукту.

Зовнішній файл даних інтерфейсу користувача. Файл, зовнішній для відповідної функції, призначений користувачем для логічно пов'язаних груп даних або блоків керуючої інформації, на які посиляється виріб, але які управляються за межами виробу.

У компанії повинна бути власна статистика по витратах ресурсів на розробку для обробки функціональних точок.

Методика функціональних точок дозволяє визначити розмір програмної системи та оцінити часові витрати на основі статистики подібних реалізацій функціональних точок.

У випадках, коли є якісна база історичних даних можна отримати високу точність оцінки. Недоліком цього підходу є те, що він дещо складний у використанні. Особа, яка проводить оцінку, повинна мати достатні технічні навички. Методика функціонально-точкового аналізу нічого не говорить про складність оцінюваного продукту, не враховує ризики і не вимагає адекватної історичної бази даних.

1.3.6 Триточкова оцінка або методика PERT

При методиці PERT (Program Evaluation and Review Technique) вимагає суворих вимог до статистичної незалежності оцінки та компетентності експерта. Надмірний оптимізм або, навпаки, надмірний песимізм в оцінці може суттєво вплинути на кінцеву оцінку [15].

Термін PERT зазвичай використовується по відношенню до мережеских діаграм. Однак це офіційна назва методології оцінювання, яка використовує середньозважене значення трьох оцінок як остаточний бал. Згідно з цією методологією, людина починає з трьох оцінок:

- найпесимістичнішої (П), якщо справи йдуть погано;
- найоптимістичнішої (О), якщо справи йдуть добре;
- найвірогіднішої (В), якщо виникають звичайні проблеми або звичайні можливості.

Підсумкова оцінка PERT дорівнює $(O+4*V+П)/6$. Наприклад, було оцінено, що на виконання завдання знадобиться приблизно 10 годин. Найкращий сценарій - 6 годин, а найгірший - 26 годин. У цьому випадку оцінка PERT буде $(6+4*10+26)/6 = 12$ годин.

1.3.7 Методика Pokerestimate

Методика Pokerestimate (покер оцінка) - це методика оцінки на основі консенсусу, який використовується в основному при розробці програмного

продукту для оцінки складності майбутньої роботи і відносного обсягу завдань, які необхідно вирішити [16].

В основі методики Pokerestimate лежать наступні принципи:

- колективне обговорення;
- колективна згода;
- групування і відносність розмірів;
- залучення експертів;
- накопичення і використання досвіду попередніх оцінок.

Покер планування вимагає визначення наступних трьох позицій: попередня підготовка, командна робота, відносні одиниці вимірювання.

Під час попередньої підготовки аналізуються вимоги на основі інформації, наданої замовником. Потім ці вимоги формалізуються у вигляді функцій, які описуються розробнику для оцінки.

Командна робота заохочує обговорення переліку завдань, які необхідно виконати, щоб досягти згоди щодо оцінки обсягу робіт. Команда може складатися з розробників, дизайнерів, продакт-менеджерів та інших експертів. У більшості випадків команду очолює бізнес-аналітик або, за відсутності бізнес-аналітика, продакт-менеджер. Мінімальна кількість людей, необхідна для проведення функціональної оцінки, - це доповідачі та розробники функціональної оцінки, незалежно від того, чи є вони бізнес-аналітиками або продукт-менеджерами. Для того, щоб отримати найбільш точні результати під час оцінки, повинні бути присутніми кілька розробників, в ідеалі - всі фахівці, які працюють над ІТ-проєктом.

Відносні одиниці виміру означають, що команда абстрагується від прямої грошової вартості завдання при підготовці кошторису. Найпоширенішою та найоптимальнішою одиницею виміру є людино-години для виконання різних завдань.

Цією методикою користуються такі компанії як GeneralElectric, Cisco, Adobe, Amazon, IBM, CocaCola, Tesla та інші.

1.3.8 Методологія Монте-Карло

Методологія Монте-Карло - це загальний термін для групи чисельних методів, заснованих на отриманні великої кількості фактичних вимірювань стохастичного (випадкового) процесу, який формується таким чином, щоб його стохастичні властивості відповідали аналогічним значенням розв'язуваної задачі [17].

Багато систем є занадто складними, щоб вивчати вплив невизначеності за допомогою аналітичних методів. Однак такі системи можна вивчати, розглядаючи вхідні дані як випадкові величини і повторюючи певну кількість обчислень N (ітерацій) для отримання результатів необхідної точності.

У сфері попередньої оцінки вартості розробки програмного продукту метод Монте-Карло використовується в поєднанні з методом триточкової оцінки. Він видає випадкові результати часу, витраченого на вирішення різних завдань, від оптимістичних до песимістичних результатів. Після отримання випадкових результатів обчислюється сума всіх завдань, яка відображає завершення ІТ-проекту випадковим чином. Ці розрахунки виконуються N разів і обчислюється статистика для визначення ймовірності того, що проект буде завершено до певного терміну.

1.3.9 Оціночні програми

В даний час існує безліч оціночних програм, найбільш поширені з яких: SLIM, Cocomo 2, Check-points, Estimacs, SelectEstimator, Price-S, SEER-SEM.

В таблиці 1.4 наведено порівняння програм оцінювання. Аналіз ґрунтується на факторах, що впливають на вартість ІТ-проекту, які були або не були враховані в цих моделях.

Було встановлено у процесі аналізу, що дві моделі SEER-SEM і Cocomo 2 мають найбільшу повноту обліку чинників, а моделі SLIM і Price-S мають достатнє охоплення факторів.

Таблиця 1.4 - Порівняння оціночних програм

Група	Фактор	SLIM	Check-points	Price-S	Estimacs	SEER-SEM	SelectEstimator	Cocomo 2
Атрибути розміру	Кількість інструкцій в коді	+	+	+	-	+	-	+
	Функціональні точки	+	+	+	+	+	-	+
	Об'єктно-орієнтовані метрики	+	+	+	-	+	+	+
Атрибути програми	Тип / домен	+	+	+	+	+	+	-
	Складність	+	+	+	+	+	+	+
	Мова	+	+	+	-	+	+	+
	Повторне використання	+	+	+	-	+	+	+
	Необхідна надійність	-	-	+	+	+	-	+
Атрибути комп'ютера	Обмеження ресурсів	+	-	+	+	+	-	+
	Стійкість платформи	-	-	-	-	+	-	+
Атрибути персоналу	Можливості персоналу	+	+	+	+	+	+	+
	Зміна кадрів	-	-	-	-	-	-	+
	Досвід персоналу	+	+	+	+	+	-	+
Атрибути проекту	Інструментарій і техніки	+	+	+	+	+	+	+
	Розриви	+	+	+	-	+	+	+
	Обмеження трафіку	+	+	+	+	+	+	+
	Зрілість проекту	+	+	-	-	+	-	+
	Згуртованість команди	-	+	+	-	+	+	+
	Питання безпеки	-	-	-	-	+	-	-
Етапи	Початок робіт	-	+	+	+	+	-	+
	Розробка	+	+	+	+	+	+	+
	Створення	+	+	+	+	+	+	+
	Підтримка	+	+	+	+	+	+	+

Модель SEER-SEM має переваги, а саме:

- можливість підключення до інших систем;
- гнучкість по відношенню до інших доменів;

- можливість калібрування з використанням як історичних даних, так і організаційної інформації;

- підтримка планування;

- безліч звітів.

Серед недоліків моделі SEER-SEM:

- оскільки SEER-SEM спочатку була створена для комерційного використання, алгоритми моделі не є доступними через можливість створення сторонніх програмних продуктів на її основі;

- незалежних досліджень щодо використання цієї моделі не було опубліковано;

- для того, щоб використовувати інформацію від організації, модель потрібно відкалібрувати.

Таким чином, модель SEER-SEM, незважаючи на свій великий потенціал, є частиною програмного продукту "чорної скриньки" і алгоритми обчислень не розкриваються, тому модель не може бути використана в самостійному програмному продукті.

Моделі Сосото 2 - надійна налагоджена модель, добре відома і вивчена, проста для використання, підтримує сучасний процес розробки програмних продуктів та володіє достатньою точністю.

Недоліків моделі Сосото 2 це потрібно калібрувати в кожному конкретному випадку, а калібрування вимагає більше інформації про попередні ІТ-проекти, яка може бути недоступною.

Перевагами моделі SLIM є легкість адаптації до автоматизації, простота користувацького інтерфейсу, можливості аналізу ризиків, аналіз чутливості моделі та різноманітні звіти в програмній реалізації моделі.

Недоліки моделі SLIM полягають у тому, що коефіцієнт стиснення розкладу завищений, немає статичних доказів того, що показник ефективності, який використовується в моделі, має бути саме таким, а база даних ІТ-проекту не є загальнодоступною.

Перевагами моделі PRICE-S та відповідних програмних продуктів є використання в режимі реального часу, аналіз ризиків, підтримка визначення розмірів в точках коду і функціональних точках, хороше калібрування моделі, відмінний користувацький інтерфейс, підтримка операційних систем Windows і Unix та можливість експорту в MS Project.

Модель PRICE-S базується на властивостях, тобто життєвий цикл IT-проєкту описується в термінах таких властивостей, як кількість інсталяцій, очікуване зростання та якість. Такі описи є досить розпливчастими і можуть призвести до значних розбіжностей в оцінці IT-проєктів з однаковим життєвим циклом різними менеджерами IT-проєктів, якщо немає чітких положень щодо оцінки конкретних властивостей. База даних, що використовується для початкового калібрування моделі, недоступна для користувачів. Підвищена чутливість до певних факторів, зокрема, до факторів продуктивності, складності та платформи. Насправді, вплив цих факторів на різні типи IT-проєктів, з персоналом різного досвіду, може сильно відрізнятись, і точність оцінок у таких випадках може значно знижуватися.

Проаналізувавши програми оцінювання, можна зробити висновок, що Socomo 2 є найбільш функціональною та популярною програмою оцінювання, яка пропонує найбільше переваг.

1.4 Класифікація методик

Кожна з методик, перелічених у попередньому пункті, має свої сильні та слабкі сторони, але порівнювати їх між собою некоректно, оскільки деякі з них охоплюють весь процес оцінювання комплексно, тоді як інші охоплюють лише деякі аспекти процесу оцінювання і не враховують інші. Для того, щоб проводити порівняння, методик необхідно їх класифікувати.

Методології зверху-вниз і знизу-вгору описують, як підійти до аналізу потреб для складання кошторису, але не описують процес складання кошторису і те, як він виражається для подальшого переведення його в грошову вартість.

Методології експертного оцінювання описують підхід до формування оцінки, а не сам процес формування оцінки та вираження її в певній формі. У свою чергу, методологію аналогій та покер планування можна віднести до різновиду експертного оцінювання, оскільки вони передбачають наявність експертів.

Методики триточнової оцінки і Pokerestimate охоплюють лише етап оцінки вже перелічених завдань і не стосуються аналізу вимог замовника, аналізу ризиків та декомпозиції завдань ІТ-проєкту.

Методика оцінки функціональних точок і оціночні програми комплексно охоплюють всі етапи оцінки розвитку ІТ-проєкту.

Враховуючи, що неможливо розрахувати попередню вартість програмного продукту за єдиною методикою, класифікуємо існуючі методи за трьома рівнями (рис. 1.5):

- за типом способу аналізу завдань;
- за типом підходу до формування оцінки;
- за типом способу формування оцінки.

Перший рівень описує підхід до аналізу, необхідний для реалізації функції або завдання. Цей рівень дозволяє вирішити, як представити завдання, що підлягає оцінці. Перший рівень включає наступні методи: зверху вниз та знизу вгору.

Другий рівень описує, як створюються оцінки. Цей рівень відповідає за те, хто і як оцінює списки завдань. Другий рівень включає такі методи: експертна оцінка та аналогія.

Третій рівень визначає, як виражаються результати оцінювання. До третього рівня належать такі методи: триточкову та Pokerestimate.

1 рівень за типом способу аналізу завдань	Зверху вниз Знизу вгору		Метод функціональних точок (Function Point)	Оціночні програми: – SLIM; – Cocomo 2; – Check-points; – Estimacs; – Select Estimation; – Price-S; – SEER-SEM;
2 рівень за типом підходу до формування оцінки	Експертна оцінка	Poker estimate		
3 рівень за типом способу формування оцінки	Оцінка по аналогії Трьохточкову або PERT			

Рисунок 1.5 – Класифікація методик формування оцінки

1.5 Порівняльний аналіз існуючих методик

Після класифікації можна порівняти методики на кожному рівні

В 1-ому рівні методики зверху-вниз і знизу-вгору. Методика знизу-вгору має переваги перед методикою зверху-вниз, а саме, точність оцінки вище та легко ідентифікувати ризики.

Недоліки методики знизу-вгору в порівнянні з методикою зверху-вниз - більше витрат ресурсів, необхідний відповідний рівень деталізації, тривалий аналіз вимог.

Точність оцінки висхідного підходу та вища ймовірність виявлення ризиків пояснюється більш детальним аналізом вимог, що вимагає більше часу та ресурсів. Водночас, є краще розуміння функціональності, необхідної для розробки, а також краще розуміння технології та можливих ризиків, пов'язаних з її реалізацією. І навпаки, неможливо провести детальний аналіз вимог без необхідного рівня деталізації.

Якщо рівень деталізації вимог такий, що можна використовувати висхідний підхід, слід витратити більше ресурсів на оцінку вартості розробки ІТ-проєкту. Це допоможе уникнути незапланованих витрат у майбутньому. Підрядникам може знадобитися до кількох робочих днів, щоб розрахувати більш точний кошторис, але непередбачувані витрати можуть багаторазово зрости в майбутньому.

Якщо неточні оцінки є достатніми, краще використовувати низхідний підхід, але розкид між мінімальною та максимальною оцінками може бути дуже великим.

У світлі вищесказаного можна зробити висновок, що витрачання більшої кількості ресурсів на підготовку кошторисів та застосування висхідного підходу щодо рівня деталізації вимог дозволить запобігти зростанню витрат у майбутньому.

На 2-му рівні експертна оцінка порівнюється з методологією за аналогією.

Методологію за аналогією краще обирати, оскільки вона оцінює завдання на основі об'єктивних історичних даних, коли завдання максимально відповідає дійсності, тоді як методологія експертної оцінки передбачає суб'єктивне судження експерта. Якщо оцінка максимально об'єктивна, вона є якісною і точною

Якщо розглядати метод з точки зору навичок, необхідних для його застосування, то переважає аналогічний метод. Експертна оцінка передбачає, що оцінювач володіє достатніми технічними навичками для оцінки різних видів робіт. У цьому відношенні метод за аналогією не вимагає глибоких знань.

Недоліком методики аналогій порівняно з методикою експертних оцінок є те, що він вимагає наявності історичної бази даних, і чим більше вона спроектована таким чином, щоб можна було знайти різні типи завдань, тим краще. З іншого боку, експертам, які використовують метод експертних оцінок, не потрібно ніякої підготовки або інших матеріалів, якщо вони вже досліджували тип завдання, яке оцінюється.

Якщо експертна група складається з одного або декількох експертів, оцінка буде максимально наближеною до об'єктивної, але ресурси, витрачені на її отримання, будуть значно більшими.

Методика по аналогії в порівнянні з методикою експертної оцінки менше затратна по ресурсам і більш об'єктивна.

З вищесказаного можна зробити висновок, що краще використовувати методологію за аналогією, коли є якісна історична база даних, але ідеальний випадок - коли ця оцінка синхронізована і схвалена експертами, які можуть оцінити завдання.

На 3-му рівні порівнюємо методики триточнової оцінки та Pokerestimate.

Основною перевагою методу триточкового оцінювання порівняно з покер плануванням є стохастична природа оцінок, тобто вони описують розвиток подій від оптимістичного до песимістичного. З іншого боку, покер планування дає точкові оцінки, які часто не відповідають дійсності.

Перевага покер планування порівняно з методикою триточнової оцінки полягає в тому, що при покер плануванні формується команда з двох або більше експертів у даній галузі, які після постановки завдання анонімно оцінюють його. Отримані оцінки потім обговорюються і в ході діалогу формується одна узгоджена оцінка, тоді як методологія триточнової оцінки передбачає суб'єктивну думку одного експерта.

Методика покер планування більш затратна по ресурсам.

З вищесказаного можна зробити висновок, що методологія покер планування вимагає більше ресурсів і не враховує імовірнісний характер, але це дуже хороший підхід до дискусії. Найкращим варіантом було б запровадити покер планування, але зробити так, щоб експерти при формуванні своїх оцінок використовували діапазон від оптимістичного до песимістичного, а не оцінювали по пунктах.

2 РОЗРОБКА МЕТОДИКИ ОЦІНКИ РОЗРОБКИ ІТ-ПРОЄКТІВ

Аналіз теоретичного матеріалу показує, що важко знайти єдину методику розрахунку попередньої вартості розробки ІТ-проєкту, яка б враховувала всі етапи оцінки, тобто аналіз вимог, оцінку часових витрат учасників розробки ІТ-проєкту, оцінку можливих ризиків ІТ-проєкту тощо. Існуючі методології розглядають окремі етапи оцінювання ІТ-проєкту та за певних умов. Тому існує потреба у формулюванні універсальної методології оцінювання розробки ІТ-проєкту.

2.1 Алгоритм методики оцінки розробки ІТ-проєктів

Звертаючись до підрядників, замовники надають свої вимоги у різних формах, включаючи документацію, макети, інтерактивні прототипи та усно. Для того, щоб оцінити вартість розробки ІТ-проєкту, спочатку необхідно зрозуміти, яку саме систему хоче розробити замовник. Тому першим кроком у підготовці кошторису є аналіз вимог замовника.

Аналіз вимог - це визначення бізнес-вимог, цілей і завдань, які система повинна дозволяти користувачам виконувати, а також розуміння того, як система працює за різних умов.

Клієнт або представник клієнта, який готує вимоги, часто подає їх у попередній формі, що може викликати невизначеність або запитання щодо тлумачення вимог. У таких випадках питання слід перерахувати і, якщо це можливо, з'ясувати з клієнтом. Якщо це неможливо, на питання потрібно відповісти самостійно, виходячи з припущень. Якщо на функціональне питання неможливо відповісти і немає припущень, необхідно проаналізувати ринок конкуруючих продуктів, щоб знайти відповідь на питання. Оскільки виконавець і замовник можуть мати різне розуміння функціональності, будь-які зроблені

припущення повинні бути записані, щоб надати замовнику факт, що він оцінив функціональність на основі певних припущень.

Результатом аналізу вимог є розуміння, наскільки це можливо, функціональності та призначення програмного забезпечення, що розробляється, на основі наданих вимог. На рисунку 2.1 показано алгоритм аналізу вимог.

Далі, відповідно до аналізу вимог, необхідно скласти перелік робіт ІТ-проєкту. Декомпозиція - це поділ великих елементів ІТ-проєкту на менші, більш керовані елементи. Рівень деталізації елементів декомпозиції залежить від розміру та складності ІТ-проєкту. Вузли - це елементи ІТ-проєкту (наприклад, завдання, підзадачі, роботи), а ребра - це поділ від елементів вищого рівня до елементів нижчого рівня.

Метод декомпозиції використовується для визначення так званої ієрархічної робочої структури. Ієрархічна структура робіт поділяє ІТ-проєкт на ієрархічно пов'язані, керовані і контрольовані елементи (роботи) і полегшує створення системи управління всіма елементами ІТ-проєкту. Ієрархічна структура робіт є основою для підготовки майбутніх мережових графіків ІТ-проєкту, що характеризують терміни виконання окремих робіт.

Елементом найвищого рівня є мета ІТ-проєкту або продукт ІТ-проєкту, який є кінцевим результатом, отриманим в результаті всіх заходів і подій ІТ-проєкту. Потім мета ІТ-проєкту розбивається на більші елементи (завдання), а продукт ІТ-проєкту - на окремі компоненти, такі як окремі невеликі продукти або системи. Вони далі розбиваються на менші компоненти. Потім відбувається перехід від окремих підзадач і результатів ІТ-проєкту до заходів, необхідних для досягнення кожного результату. Діяльність ІТ-проєкту можна розбити на менші завдання, процедури і заходи, як показано на рисунку 2.2. Для оптимізації структури подібні завдання, необхідні для досягнення різних результатів, можна об'єднати в одну операцію (робочу систему). У реальному ІТ-проєкті може знадобитися введення додаткових рівнів, наприклад, поділ підзадач на заходи, а потім подальший поділ їх на роботи.

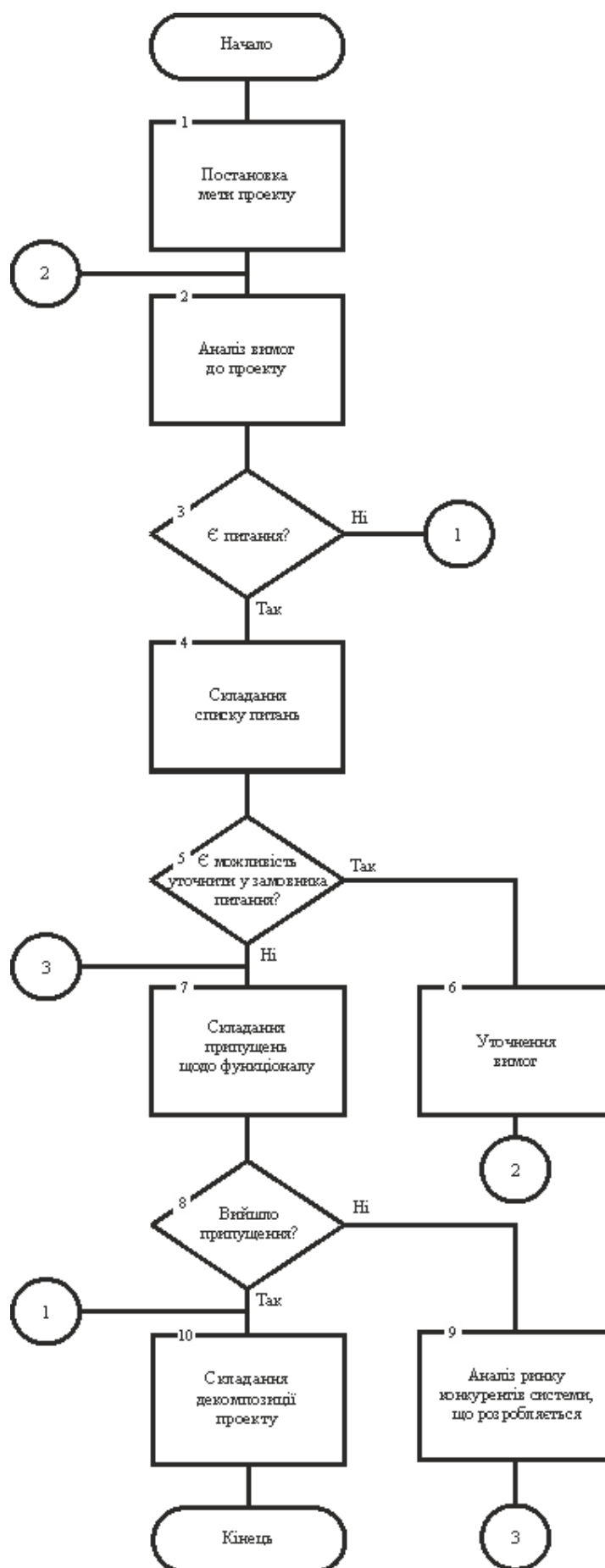


Рисунок 2.1 – Алгоритм аналізу вимог



Рисунок 2.2 - Схема декомпозиції ІТ-проєкту

При ієрархічній декомпозиції роботи кожен рівень зменшує складність кожного елемента, поки не буде досягнутий розумний рівень з точки зору управління. "Розумний" рівень можна визначити так: елементи на нижчих рівнях можуть бути призначені відповідальним особам, призначеним для виконання ІТ-проєкту, без необхідності додаткового управління і без необхідності подальших пояснень. Елементи нижнього рівня в декомпозиції є елементами управління ІТ-проєктом.

Деякі результати можуть потребувати різних рівнів декомпозиції на заходи. Деякі результати можуть бути декомпоновані лише на наступний рівень, щоб досягти рівня деталізації, необхідного для діяльності, в той час як інші можуть потребувати подальших рівнів декомпозиції. Чим більше елементів структури декомпоновано, тим більше можливостей для планування, управління та контролю роботи, але надмірна декомпозиція може призвести до ускладнення управління ІТ-проєктом, неефективного використання ресурсів і низької продуктивності.

Методику знизу-вгору використовуємо для складання декомпозиції.

Далі будується технологічний стек. Технологічний стек - це набір інструментів, які використовуються в ІТ-проєкті, включаючи мови програмування, фреймворки, системи управління базами даних і компілятори.

Стек технологій, обраний розробником, визначає продуктивність, вимоги до апаратного забезпечення та надійність програмного продукту. В результаті, для складних процесів і високонавантажених систем стек технологій, обраний розробником, визначає, наскільки доступним є впровадження та експлуатація системи і наскільки добре вона відповідає вимогам замовника.

Для формування технологічного стеку слід звернутися до експертів, які мають достатню кваліфікацію для вирішення подібних завдань, щоб пояснити бачення, цілі та функції IT-проєкту, а також вибрати найбільш підходящу технологію, яка відповідає системним вимогам і доступна для співробітників компанії. Вибір технологічного стеку безпосередньо визначає потенційних розробників, які можуть очолити цей IT-проєкт.

Наступним кроком є вибір методології управління IT-проєктом. У сучасній розробці IT-проєктів дуже часто в команді присутня роль менеджера IT-проєкту, в той час як роль бізнес-аналітика зустрічається рідше, але також поширена. Вибір методології управління IT-проєктами безпосередньо впливає на обсяг їхніх завдань і час, який вони витрачають на їх виконання.

Підхід до управління IT-проєктом обирається насамперед відповідно до ступеня залучення клієнта або його представника. Другий етап - це рівень необхідної деталізації. Якщо початкові вимоги є максимально деталізованими і можуть бути чітко інтерпретовані, то можна використовувати методологію управління IT-проєктами за принципом водоспаду. Вибір цієї методології означає, що основна робота менеджера IT-проєкту і бізнес-аналітика припадає на початкову фазу і зводиться до мінімуму на наступних етапах.

Методологія Scrum передбачає регулярні дзвінки від команди, що працює над IT-проєктом, до замовника. Залежно від частоти таких дзвінків і тривалості IT-проєкту, вони можуть займати значну кількість часу. Наприклад, замовник може запитувати дзвінки з командою тричі на тиждень, а над IT-проєктом працює три розробники. В середньому такий дзвінок триватиме близько 30 хвилин. Якщо IT-проєкт триває чотири місяці, загальний час, витрачений розробниками, становитиме 72 години.

Після визначення методики управління IT-проектом можна планувати життєвий цикл розробки IT-проекту та визначати завдання проєктних менеджерів і бізнес-аналітиків.

Після визначення технологічного стеку та методики IT-проекту слід зібрати та визначити потенційну команду для роботи над IT-проектом, залежно від навичок, необхідних для досягнення цілей та завдань IT-проекту. Якщо потенційну команду для роботи над IT-проектом неможливо визначити заздалегідь, слід обрати групу експертів, які можуть оцінити час, який буде витрачено на виконання завдання.

Після того, як всі дані для оцінки доступні, особа, відповідальна за управління процесом оцінки, порівнює поточні завдання з завершеними IT-проектами, якщо є якісь історичні дані. Якщо є збіг, поточні завдання оцінюються в порівнянні з завершеними завданнями. Таким чином, при обговоренні з командою вони можуть більш об'єктивно оцінити час, витрачений на виконання завдання.

Після того, як команда або експерти зустрілися для підготовки кошторису, оцінювач IT-проекту повинен пояснити всі вимоги та бачення IT-проекту, щоб усі розуміли цілі та завдання IT-проекту.

Наступним кроком оцінювач описує завдання команді, яка оцінює IT-проект. Далі кожен самостійно оцінює час, витрачений на виконання завдання, використовуючи трибальну оцінку. Команда обговорює оцінки і приходить до консенсусу щодо того, як отримати одну оптимістичну, одну реалістичну і одну песимістичну оцінку. Якщо є схожі оцінки для завдання, команда порівнює ці оцінки і, за необхідності, коригує їх для отримання остаточного результату. Таким чином команда оцінює всі завдання IT-проекту і, якщо під час обговорення виявляються ризики, їх одразу ж ідентифікують та оцінюють.

Алгоритм формування оцінки часу витрат на вирішення певних задач показано на рисунку 2.3.

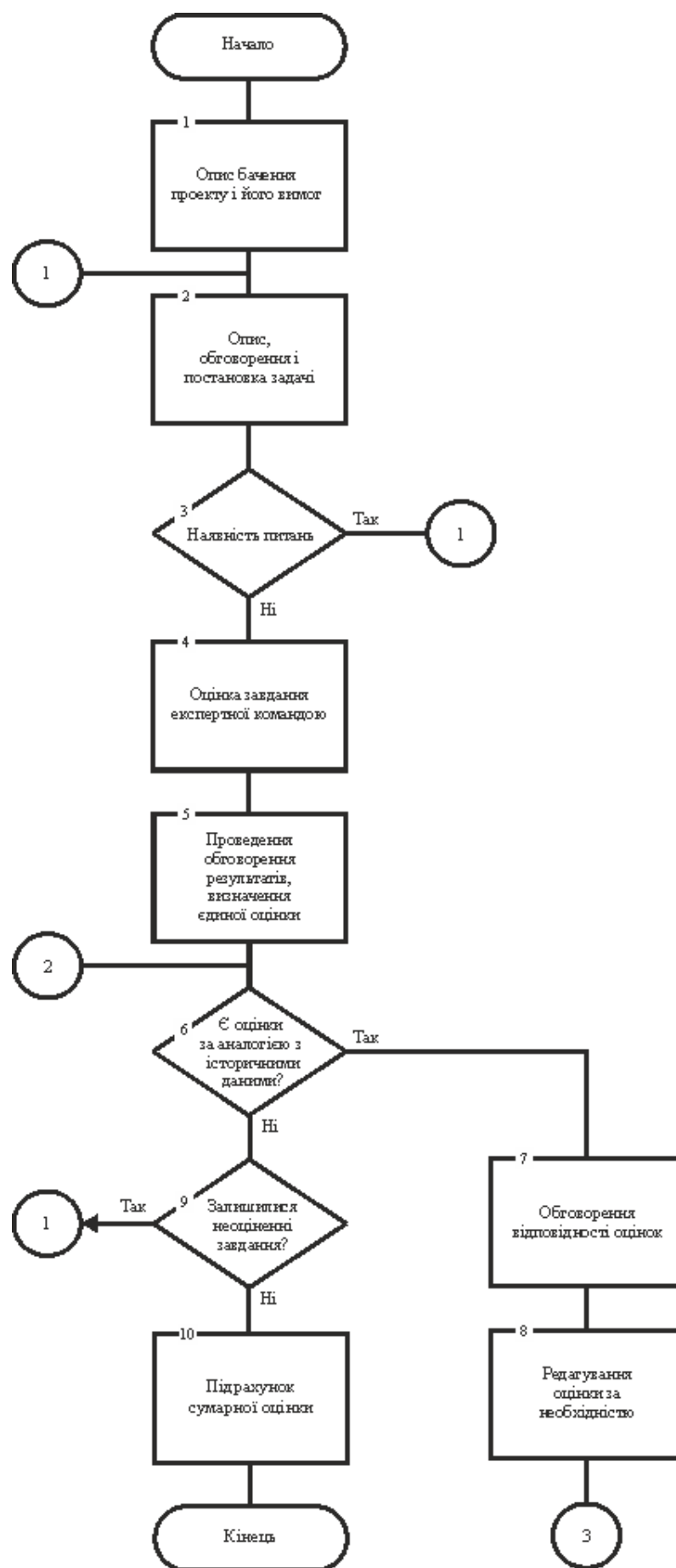


Рисунок 2.3 – Алгоритм формування оцінки

Далі необхідно проаналізувати ризики, які можуть вплинути на хід виконання ІТ-проєкту. Для цього розробляється сигнальна матриця ризиків ІТ-проєкту (табл. 1.3).

Оцінивши час, витрачений на виконання завдання, експерт або команда вирішує, скільки часу слід додати до загальної підсумкової оцінки, щоб отримати додатковий час для успішного усунення ризику без зайвих втрат.

Наступним кроком є перевірка результатів оцінки декомпозиції ІТ-проєкту за допомогою методу Монте-Карло. Метод Монте-Карло аналізує остаточну оцінку часу, витраченого на вже згенеровані завдання.

Суть методу Монте-Карло полягає в багаторазовому отриманні результатів випадкової події на основі триточнової оцінки всіх завдань ІТ-проєкту. Наприклад, завдання А оцінюється в 10 годин в оптимістичному випадку і 20 годин в песимістичному випадку. Метод Монте-Карло приймає випадкові значення в діапазоні від 10 до 20. Таким чином моделюються випадкові величини для всіх завдань ІТ-проєкту. Потім всі випадкові значення підсумовуються для отримання випадкового результату для ІТ-проєкту. Випадкові результати моделюються кілька разів. Потім відображається статистика, яка показує ймовірність того, що проєкт буде завершено через певний проміжок часу. Ймовірність виражається як кількість отриманих випадкових результатів. При використанні методу Монте-Карло для отримання результатів найкращою оцінкою вважається оцінка з ймовірністю 95%.

Алгоритм перевірки результатів оцінки декомпозиції ІТ-проєкту методологією Монте-Карло показано на рисунку 2.4.

Застосування методології Монте-Карло дозволяє отримати остаточну оцінку розбивки ІТ-проєкту, а перелік питань, уточнень і припущень, на яких базується оцінка, допомагає підряднику вести продуктивний діалог з клієнтом щодо функціональної розробки. Формулювання припущень також допомагає клієнту зрозуміти, що виконавець дійсно оцінив, що клієнт мав на увазі або не мав на увазі під вимогами ІТ-проєкту.

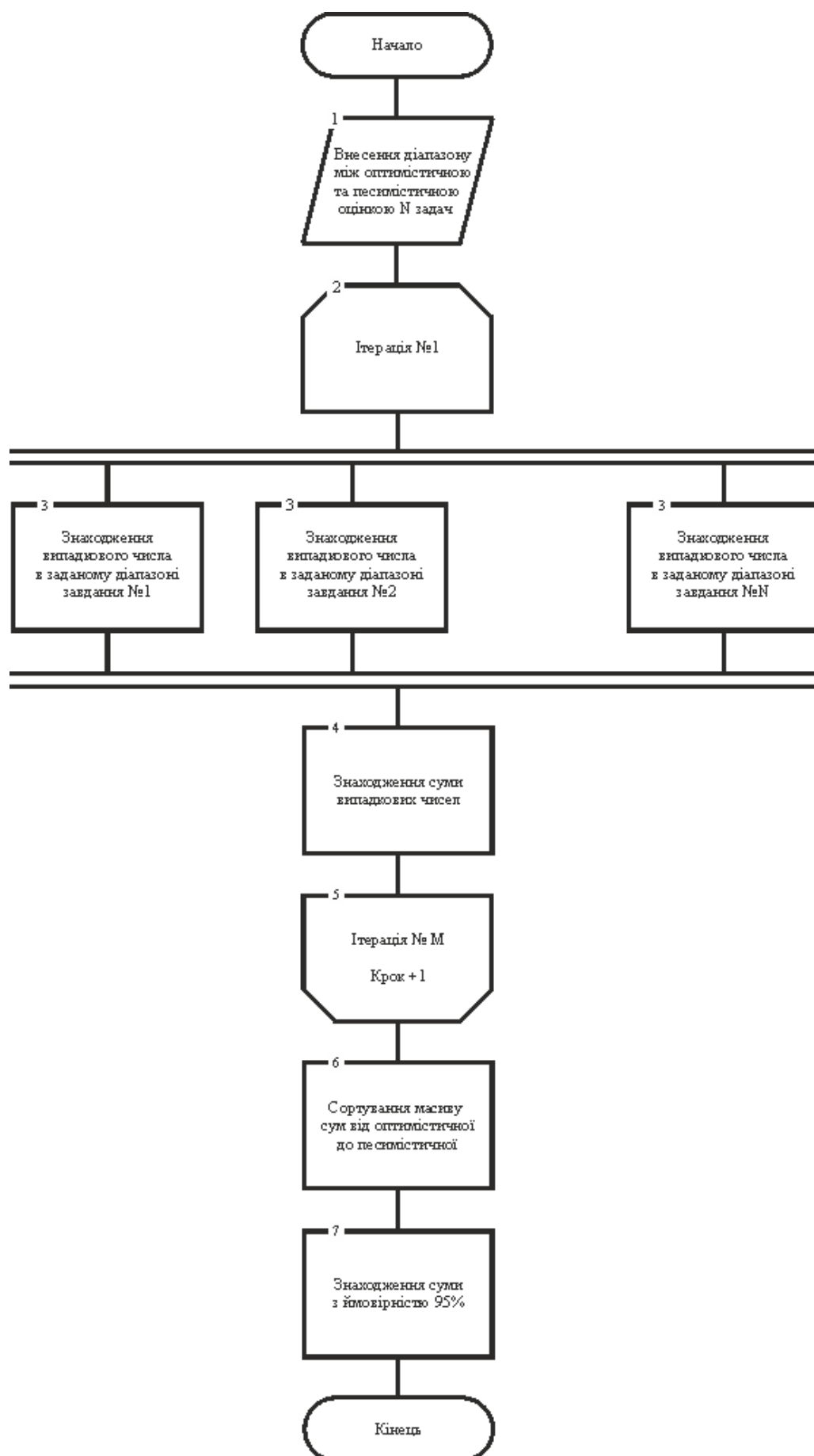


Рисунок 2.4 - Алгоритм перевірки результатів оцінки методологією Монте-Карло

При розробці ІТ-проектів приблизно 90% витрат припадає на заробітну плату працівників, які працюють над ІТ-проектом. Тому якісна оцінка декомпозиції завдань ІТ-проекту дає уявлення про загальний час, витрачений на розробку.

В кінці залишається провести конвертацію годин, витрачених на розробку, в грошовий еквівалент в залежності від заробітної плати співробітників, розрахунок ресурсів, не пов'язаних із заробітною платою працівників, що витрачаються на розробку ІТ-проектів, додавання маржі.

Таким чином, сума маржі компанії-виконавця, заробітної плати працівників та інших витрат на ресурси стає остаточною єдиною цифрою вартості розробки ІТ-проекту.

Алгоритм універсальної методики оцінки розробки ІТ-проекту подано на рисунку 2.5.

В результаті оцінки бізнес-аналітик повинен отримати:

- перелік питань, пояснень та припущень, на яких ґрунтувалася оцінка;
- оцінку декомпозиції завдань ІТ-проекту в триточкову вигляді;
- світлофорну матрицю ризиків з їх ваговими коефіцієнтами;
- методологією Монте-Карло аналіз оціненої декомпозиції ІТ-проекту.

2.2 Дослідження методики оцінки розробки ІТ-проектів

Протестувати розроблену методологію на попередніх витратах на розробку програмного продукту в реальному ІТ-проекті.

В якості вкладу від замовника ми надали письмові вимоги, включаючи бачення проекту, нефункціональні вимоги, вимоги до наступного етапу розробки та інтерактивні макети з коментарями.

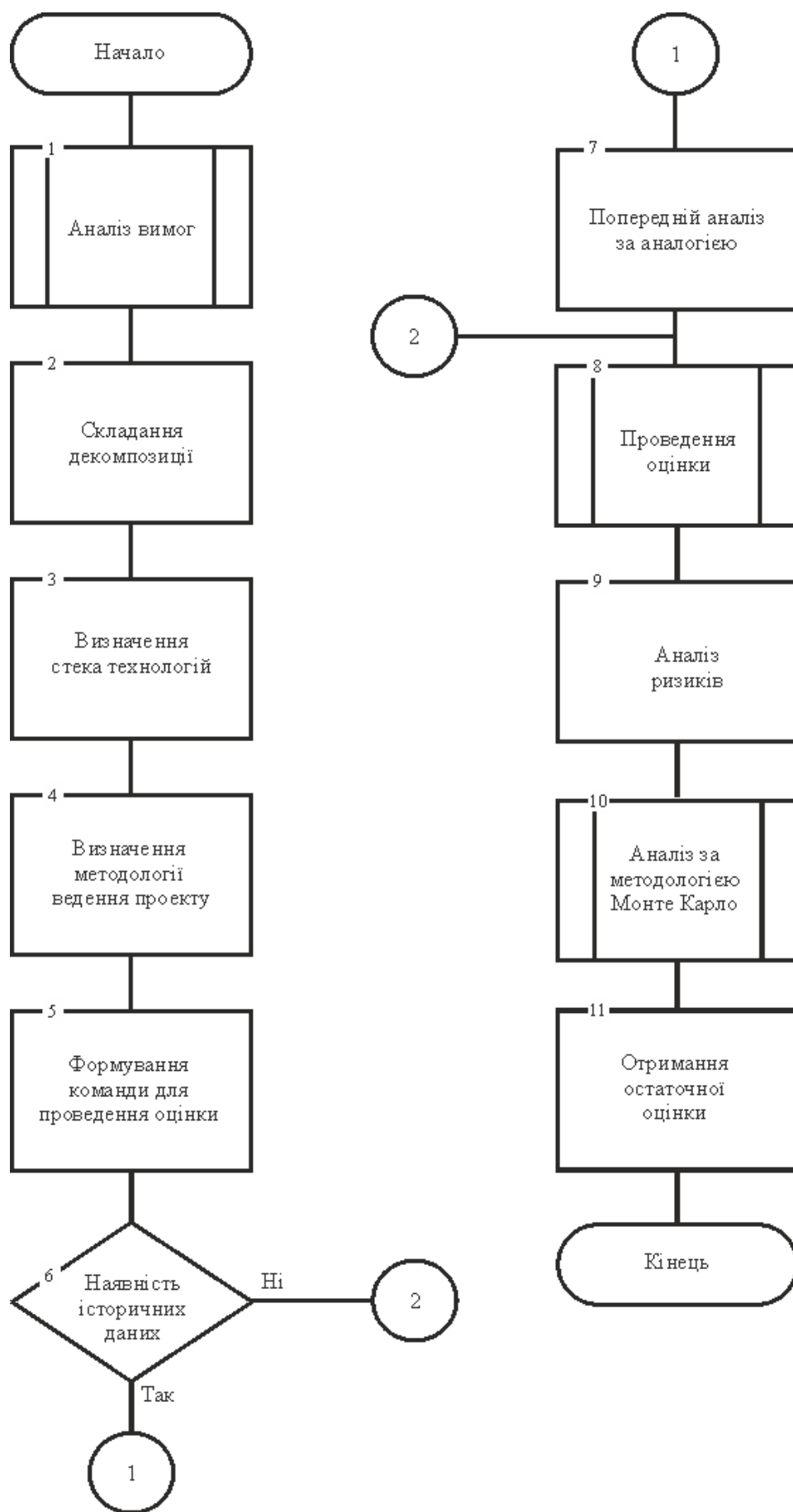


Рисунок 2.5 – Алгоритм методики оцінки розробки ІТ-проектів

Перший крок - аналіз вимог. В результаті аналізу вимог розуміються бізнес-вимоги, управління IT-проєктом, цілі та завдання, функціональні та нефункціональні вимоги.

Бізнес-вимоги IT-проєкту. Замовник хоче розробити веб-систему, яка може створювати інтерактивні анкети та вставляти скрипти в блоки HTML-коду для поширення на зовнішні сайти. Система повинна дозволяти замовнику пропонувати послугу використання системи за платною підпискою, яка надає індивідуальний обліковий запис і доступ до функціоналу системи. Перша версія додатку буде мати функціонал системи без можливості придбання підписки. На наступному етапі буде додана можливість платної підписки.

Розуміння управління IT-проєктами, цілей і завдань означає, що система повинна мати можливість створювати організаційні сутності, що містять інформацію про IT-проєкт. Проєкт - це інтерактивна анкета. Проєкти можуть бути опубліковані і вставлені в будь-яку HTML-сторінку за допомогою скриптів, що дозволяє користувачам проходити опитування на зовнішньому веб-сайті. Система повинна збирати результати інтерактивного опитування, відображати їх для дослідницьких цілей та експортувати в Excel. Система повинна дозволяти адміністратору організації створювати та видаляти інших користувачів системи всередині організації.

IT-проєкт містить 4 типи користувачів:

- користувачі, які заповнюють інтерактивні анкети на зовнішніх веб-сайтах (зовнішній користувачі);
- опитувальник - може проводити маніпуляції з проєктами і бачити результати певного проєкту.
- власник організації - має всі права, що і опитувальник, а також доступ до сторінки управління користувачами.
- головний адміністратор - має доступ до сторінки управління організаціями і має права власника.

Щоб зрозуміти функціональні вимоги, необхідно пояснити бачення IT-проєкту, а також встановити цілі та завдання IT-проєкту.

Нефункціональні вимоги це:

- висока продуктивність і доступність;
- стиснення відео для чудової продуктивності та невеликого обсягу пам'яті;
- вбудуйте вікторини на будь-який сайт з можливістю вибору розміру та автоматичної зміни розміру;
- платформа повинна бути побудована у вигляді хмарного рішення та розміщена на Amazon AWS;
- навколишнє середовище платформи має відповідати вимогам Європейського Союзу щодо захисту даних і конфіденційності (GDPR).

Після аналізу вимог сформовано декомпозицію функціональних вимог ІТ-проєкту. Результати декомпозиції представлені в колонках 1 та 5 таблиці А.1 Додатку А. Оскільки не було можливості уточнити вимоги та задати питання замовнику, у колонці "Коментарі та припущення" таблиці А.1 описано припущення щодо функціональності.

Після визначення загального обсягу робіт, наступним кроком є визначення стеку технологій разом з експертом. Для визначення стеку технологій експерт знайомиться з вимогами до ІТ-проєкту, баченням ІТ-проєкту, цілями та завданнями. Експерт сформував наступний стек технологій, а саме, Node.JS, React.JS, Redux, PostgreSQL, Sequelize, AWS S3 bucket, AWS Cognito, AWS EC2.

Далі керівник ІТ-проєкту та вся команда визначилися з методологією управління ІТ-проєктом, щоб встановити правильні оцінки. Було обрано гнучку методологію бачення ІТ-проєкту з елементами Scrum, оскільки вимоги замовника не були чітко визначені і потребували уточнення протягом життєвого циклу розробки системи.

Потім була визначена команда для проведення оцінки завдання, що складалася з двох розробників з необхідними навичками для обраного стеку технологій, менеджера проєкту та тестувальника.

По-друге, проведено попередній аналіз обсягу робіт за аналогією з завершеними ІТ-проектами. Оскільки історична база даних була відсутня, оцінка за аналогією з іншими ІТ-проектами не була можливою.

Після того, як команда була визначена, була проведена оцінка цілей ІТ-проекту. Для проведення оцінки всі члени команди були ознайомлені з баченням, вимогами, цілями та завданнями ІТ-проекту.

Всі завдання проекту були розглянуті в ітеративному порядку, починаючи з визначення завдань і закінчуючи обговоренням і визначенням єдиного обсягу оцінки завдань. Разом з розробниками менеджер проекту та тестувальники оцінювали функціональні вимоги та обсяг робіт з розробниками. При цьому були перераховані можливі ризики, пов'язані з розробкою ІТ-проекту.

Результати процесу оцінювання були представлені у вигляді діапазону від оптимістичних до песимістичних оцінок загальної кількості годин, необхідних кожному працівнику для виконання своїх завдань. Результати представлені в колонках 2-4 Таблиці А.1 в Додатку А. Загальний час, необхідний для всіх працівників, представлений у таблиці 2.1.

Таблиця 2.1 - Загальна кількість годин після формування оцінки

Загальна кількість годин розробника		
Оптимістична	Реалістична	Песимістична
577	794	1014

Оскільки розробка була менш складною, ніж навички розробника, і не було інтеграції з зовнішніми сервісами, то ризиків, пов'язаних з роботою розробника, практично не було і ймовірність їх виникнення практично дорівнювала нулю. У зв'язку з цим перелік ризиків був мінімальним і не було необхідності створювати матрицю світлофорів.

Наступним кроком була перевірка результатів оцінки декомпозиції ІТ-проекту за допомогою методу Монте-Карло. Цей метод використовувався для перевірки лише завдань розробників. Це пов'язано з тим, що помилки інших

працівників не впливають на результат ІТ-проєкту, оскільки їхнє навантаження невелике.

Для отримання результатів оцінки методом Монте-Карло використовувалася таблиця Excel. Всі завдання розробників у ІТ-проєкті були введені в цю таблицю, і для кожного завдання були отримані випадкові числа в діапазоні від оптимістичного до песимістичного результату. Була отримана сума кожного рядка всіх випадкових значень для завдання, яка відображає випадковий результат завершення ІТ-проєкту. Таких значень було 1000. Потім була отримана статистика, що відображає кількість ІТ-проєктів, завершених до певного часу. Результати показали, що 950 випадків, що становить 95% від 1000 випадків, були завершені, із загальним часом 804 години.

Останнім кроком є отримання остаточної оцінки на основі результатів оцінки, моделювання за методом Монте-Карло та аналізу ризиків.

Попередня оцінка розробників становила 794 години, тоді як результат за методом Монте-Карло - 804 години. Припускаючи мінімальну кількість ризиків і виходячи з часу розробки 804 години, остаточні оцінки вартості учасників ІТ-проєкту наведені в таблиці 2.2.

Таблиця 2.2 - Загальна кількість годин після формування остаточної оцінки

Загальна кількість годин розробника		
Оптимістична	Реалістична	Песимістична
580	804	1014

Загально витрачений робочий час на формування оцінки такий - бізнес аналітик - 14 годин, розробник - 8 годин.

Після оцінки часу, витраченого учасниками ІТ-проєкту, були підраховані витрати на ІТ-проєкт:

- годин всіх учасників ІТ-проєкту помножен на вартість їх роботи в годину;
- враховані витрати на офісні товари;

– маржа компанії.

Отримані результати роботи над проектом представлені у Додатку А таблиця А.2.

Для того, щоб порівняти попередню оцінку та отримані результати для кожного завдання розробника, було створено графік, як показано на рисунку 2.6. По осі абсцис відкладено всі завдання, відсортовані за зростанням тривалості для наочності на графіку; по осі ординат - тривалість кожного завдання.

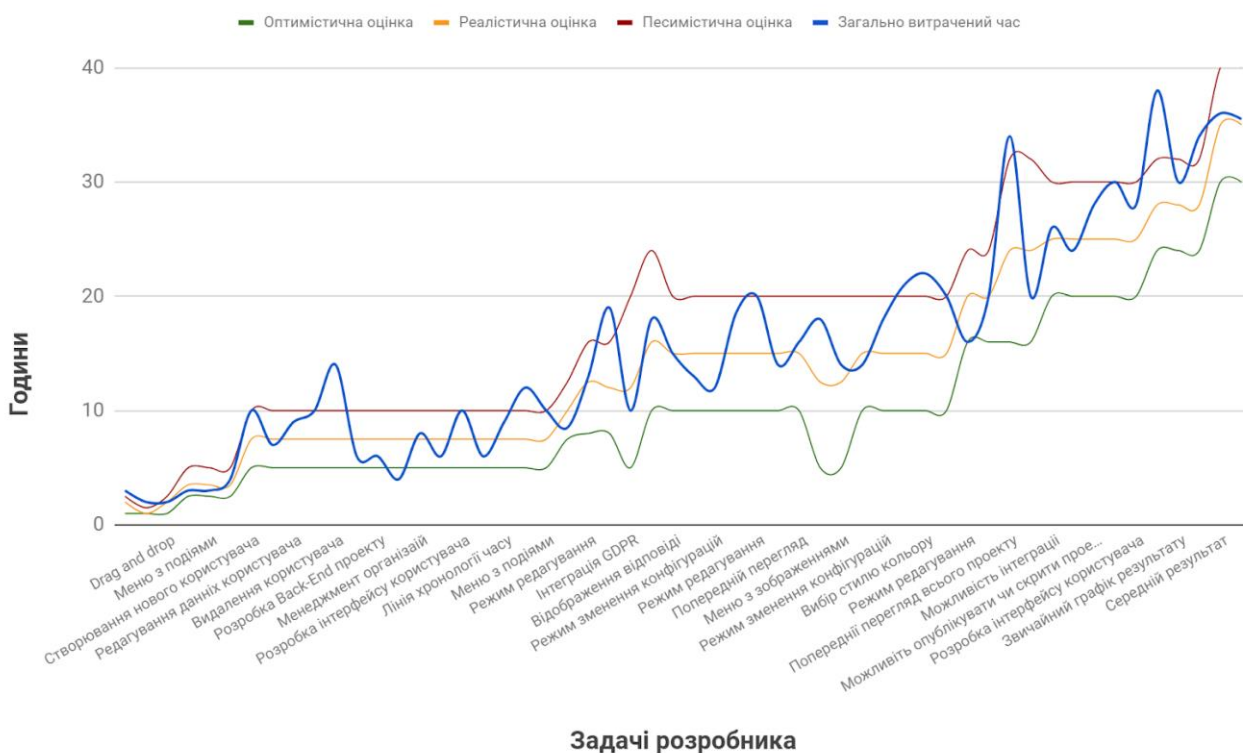


Рисунок 2.6 - Порівняння попередньої оцінки та отриманих результатів за кожним завданням розробника

На рисунку 2.7 на графік додано лінії тренду, які порівнюють попередні оцінки та результати, отримані для кожного завдання розробника за кожним індикатором. Можна зробити висновок, що загальний необхідний час є вищим за реалістичну оцінку, але нижчим за песимістичну.

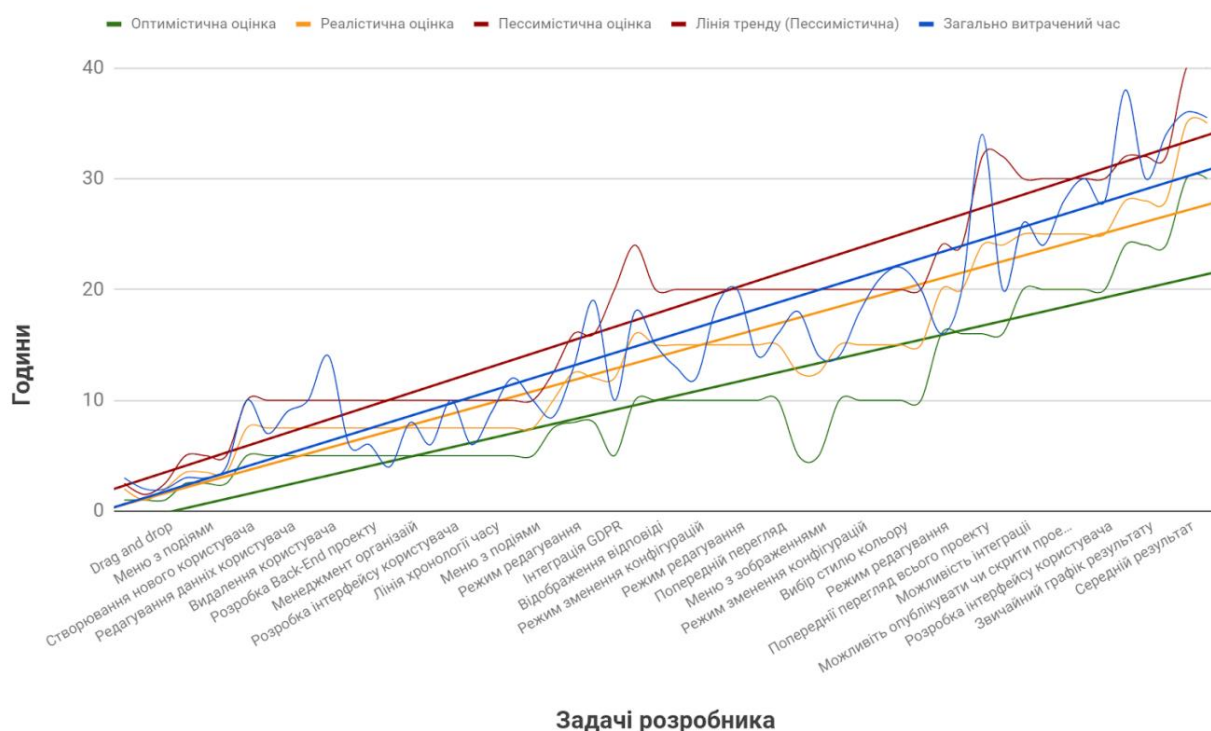


Рисунок 2.7 - Порівняння попередньої оцінки та отриманих результатів за кожним завданням розробника з лінією тренду за кожним показником

Порівняння попередньої оцінки та результатів, отриманих під час розробки ІТ-проєкту представлено в таблиці 2.3

Таблиця 2.3 - Порівняння результатів

Співробітник	Попередня оцінка	Витрачений час на розробку	Різниця
Розробник	804	816	12

Бачимо, що розробник вийшов за рамки реалістичної оцінки.

На рисунку 2.8 показано графік для розрахунку відсотка помилок. Зеленим кольором на графіку показано лінію тренду для середнього результату, який становить - 13%. Це означає, що середня абсолютна похибка часу, витраченого розробником на ІТ-проєкт, порівняно з реалістичною оцінкою становить 13%.

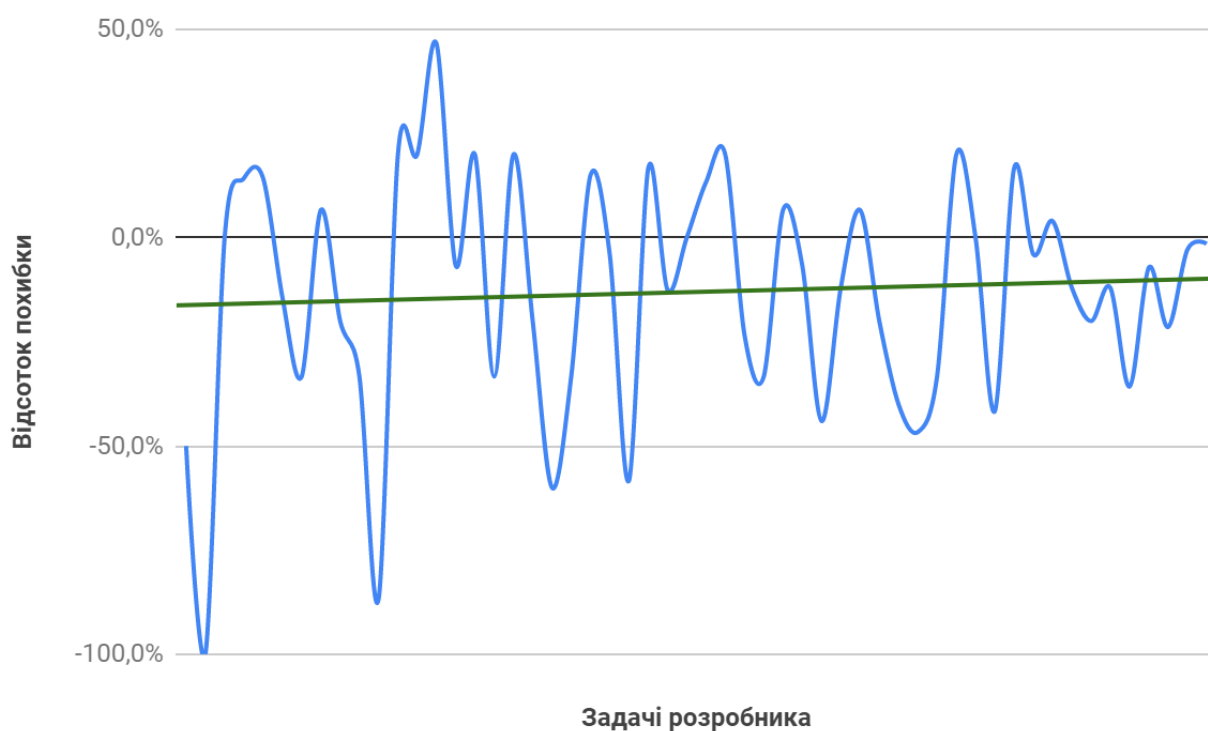


Рисунок 2.8 - Графік розрахунку відсотка похибки

Далі представлено аналіз методики на 2-ому ІТ-проекті, результати якого подано в таблиці 2.4.

Таблиця 2.4 - Загальна кількість годин після формування остаточної оцінки

Загальна кількість годин розробника		
Оптимістична	Реалістична	Песимістична
342	442,5	543

Загально витрачений робочий час на формування оцінки 2-го ІТ-проекту:

- бізнес-аналітик -6 годин;
- розробник -4 години.

Як показано на рисунку 2.9, для порівняння попередніх оцінок та результатів, отриманих для кожного завдання 2-ого ІТ-проекту, було створено графік. По осі x відкладено всі завдання 2-ого ІТ-проекту, відсортовані за

зростанням тривалості, щоб графік було легше читати. По осі у відкладено тривалість виконання кожного завдання 2-ого ІТ-проекту.

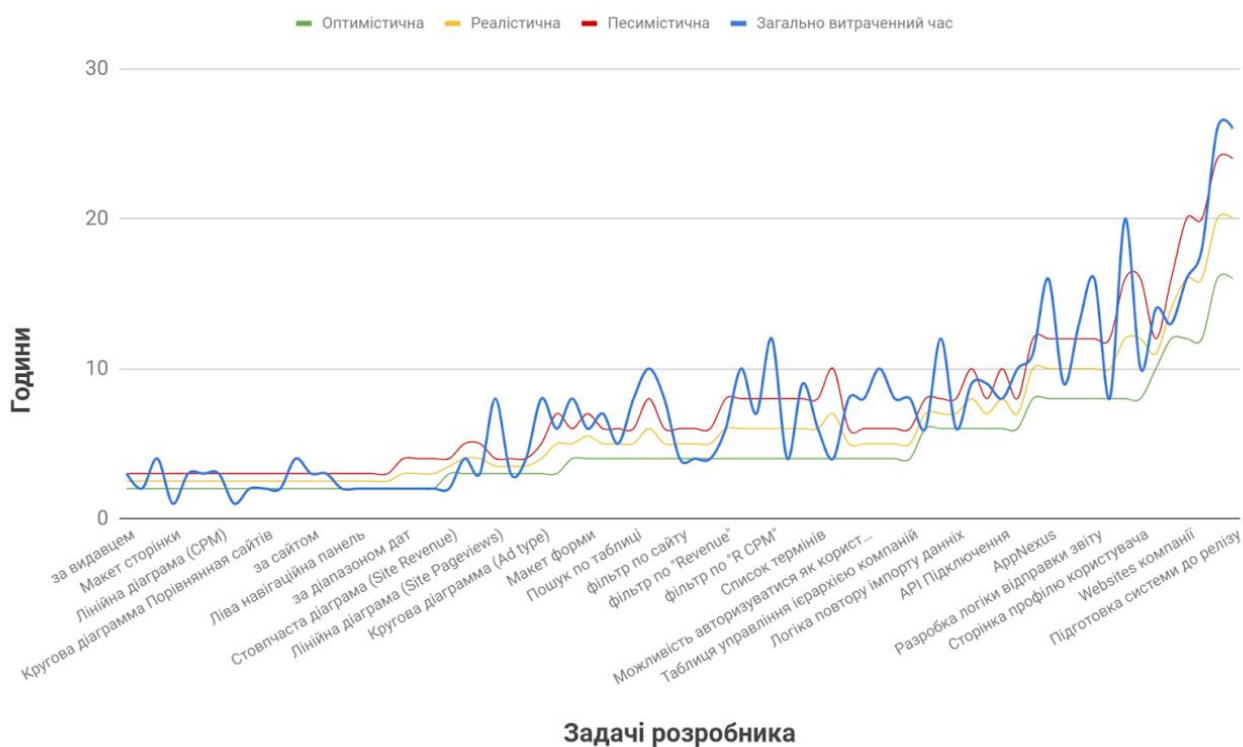


Рисунок 2.9 - Порівняння попередньої оцінки та отриманих результатів за кожним завданням розробника 2-ого ІТ-проекту

На рисунку 2.10 лінії тренду для кожного індикатора були додані до графіка, який порівнює результати, отримані за попередньою оцінкою та за кожним завданням 2-ого проекту.

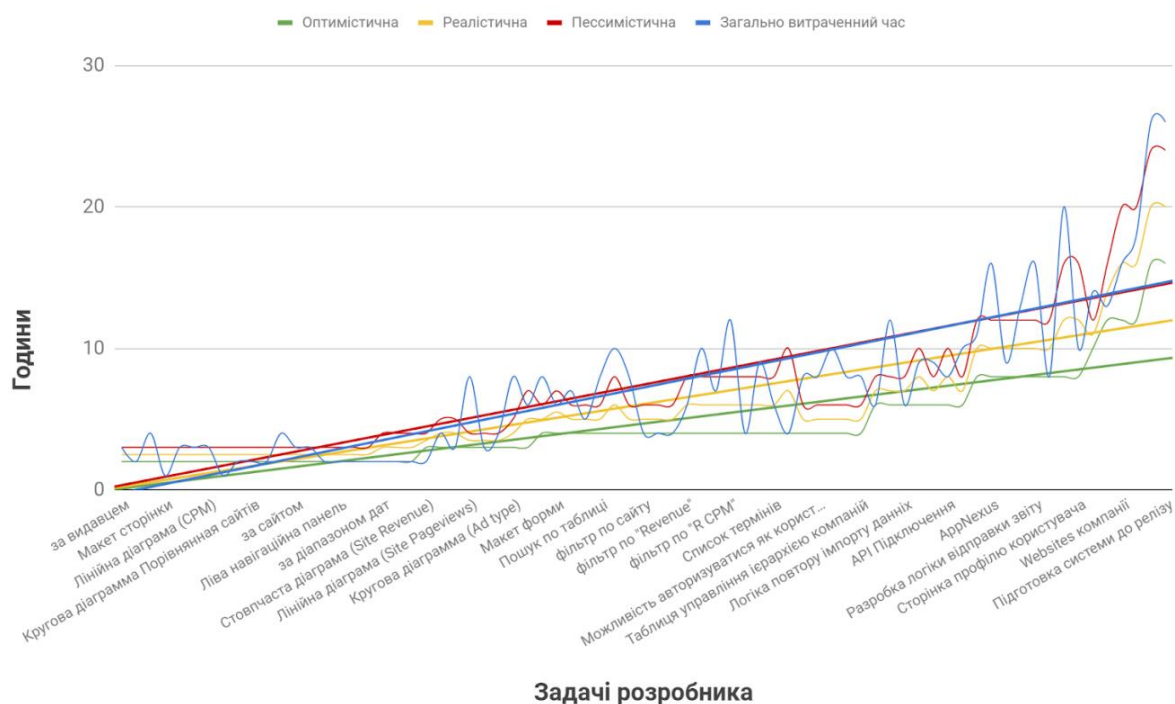


Рисунок 2.10 - Порівняння попередньої оцінки та отриманих результатів за кожним завданням розробника 2-ого ІТ-проєкту з лінією тренду за кожним показником

Порівняння результатів, отриманих під час попередньої оцінки та розробки 2-ого проєкту, представлено в таблиці 2.5. З порівняння результатів видно, що розробники перевищили реалістичні оцінки.

Таблиця 2.5 - Порівняння результатів для 2-ого ІТ-проєкту

Співробітник	Попередня оцінка	Витрачений час на розробку	Різниця
Розробник	442,5	521	78,5

На рисунку 2.11 наведено графік для розрахунку відсотка помилок. Зеленим кольором на графіку показано лінію тренду для середнього результату, який становить - 14,5%. Це означає, що середня абсолютна похибка часу, витраченого розробниками на 2-ий ІТ-проєкт, відносно реалістичної оцінки становить 14,5%.

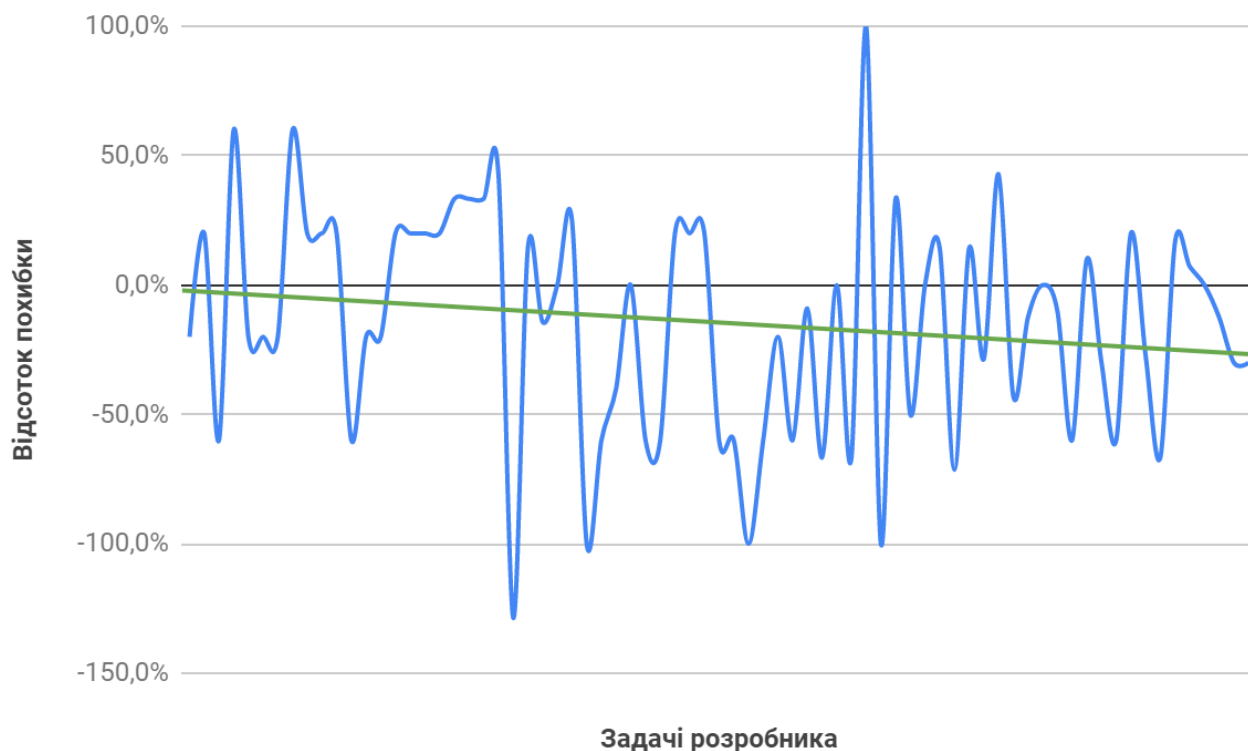


Рисунок 2.11 - Розрахунок відсотка похибки для 2-ого ІТ-проєкту

Розроблену методику було апробовано на прикладі двох ІТ-проєктів, в результаті чого було отримано попередню оцінку очікуваної кількості годин, які будуть витрачені на розробку, та розрахунок фактичної кількості годин, витрачених на розробку ІТ-проєкту.

Результати показали, що 1-ий ІТ-проєкт з реалістичною оцінкою 804 години роботи розробника потребує 14 годин роботи бізнес-аналітика і 8 годин роботи розробника. 2-ий ІТ-проєкт з реалістичною оцінкою 442,5 години роботи розробника потребував би 6 годин роботи бізнес-аналітика і 4 години роботи розробника.

Порівняння результатів абсолютних похибок у двох дослідженнях ІТ-проєктів дозволяє зробити висновок, що відхилення між попередніми та фактичними оцінками кількості годин, витрачених на розробку двох ІТ-проєктів, між показниками учасників розробки не перевищує 15% для реалістичних оцінок ІТ-проєктів.

3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ

У розділі 2 було розроблено та протестовано на двох реальних проєктах методику розрахунку попередньої вартості ІТ-проєктів. Розбіжність між попередніми та фактичними оцінками учасників щодо кількості годин, витрачених на розробку, не перевищувала 15% для реалістичної оцінки та на 10% менше для песимістичної. Висока ефективність розробленої методики дозволила впровадити її в ІТ-компанії «Freshcode».

На основі розробленої методики були розраховані попередні витрати на розробку ще кількох актуальних проєктів. Це дозволило накопичити дані про тривалість певних типових завдань, присутніх майже у всіх проєктах, які потім можна було використовувати для розрахунку наступних проєктів. Це означає, що в деяких випадках краще застосовувати дану методику оцінювання за аналогією, перевагою якої є висока точність і низьке споживання ресурсів.

Це призвело до розробки комп'ютерної системи оцінювання вартості ІТ-проєктів за застосування методики оцінювання за аналогією. Це значно скорочує час, що витрачається на розрахунок початкової вартості розробки ІТ-проєктів, і дозволяє максимально зменшити попередню оцінку кількості годин, витрачених на розробку.

3.1 Архітектура комп'ютерної системи

Було обрано програмне забезпечення Portal CRM для розробки комп'ютерної системи зберігання бази даних оцінок за аналогією, яка була розміщена в хмарному сховищі AWS (Amazon Web Services). Цей вибір був зумовлений тим, що саме з таким стеком технологій та типом архітектури працює

компанія «Freshcode», яка реалізує розроблену методику оцінювання вартості IT-проектів. На рисунку 3.1 зображена архітектура комп'ютерної системи.

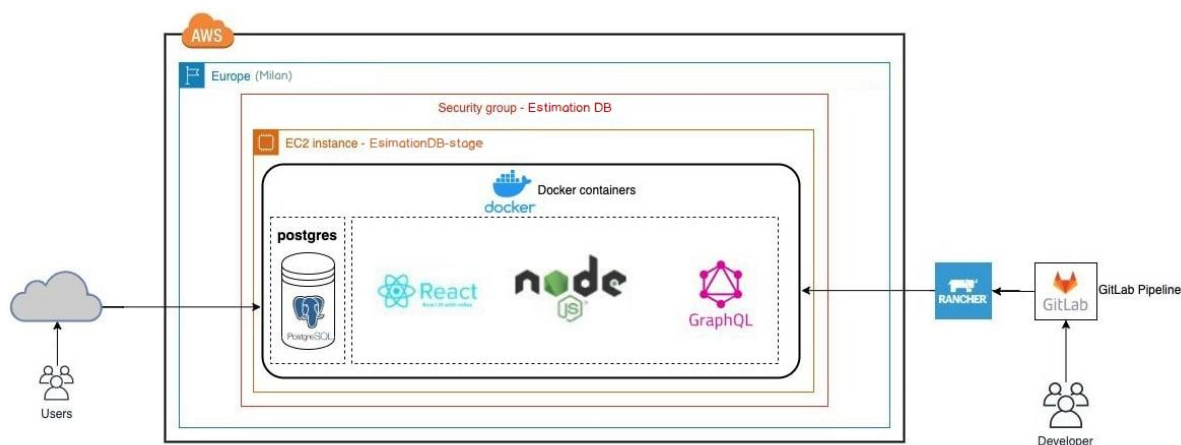


Рисунок 3.1 – Архітектура комп'ютерної системи

Програмне забезпечення Portal CRM базується на наступних технологіях:

- React.js;
- PostgreSQL;
- Node.js;
- GraphQL.

Обране хмарне сховище AWS має наступні характеристики:

- операційна система Amazon EC2 - Linux;
- 2-х ядерний процесор IntelScalable- 2,5 ГГц;
- обсяг пам'яті - 10 ГБ;
- зберігання для кожного екземпляра EC2 - SSD загального призначення (gp2);
- навантаження - послідовне, кількість випадків: 1;
- передача даних;
- Advance EC2 instance- t3.small;
- вартість передачі даних - 11.583;
- об'єм оперативної пам'яті - 2 Гб;

- потужність мережі - до 1 Гбит/с.

3.2 Проєктування системи

Розроблено базу даних для зберігання оцінок тривалості виконання певних типових завдань, накопичених у ході різних проєктів компанії «Freshcode».

База даних має одну сутність, яка складається з наступних атрибутів:

- Name (ім'я);
- Images (зображення);
- CustomEstimate (оцінка);
- Status (статус);
- Tags (тег);
- CustomSolution (опис).

Атрибут «Name» має тип даних «String» (рядок) та містить ім'я задачі.

Атрибут «Images» має тип даних «File» (файл) та може містити зображення, що демонструють приклади складності компонентів користувацького інтерфейсу.

Атрибут «CustomEstimate» має тип даних «String» (рядок) та містить триточкову реалістичну, оптимістичну та песимістичну оцінки витрат часу в годинах розробника.

Атрибут «Status» має тип даних «String» (рядок) та містить маркери «SuperRough», «Rough» та «Accurate», які вказують на достовірність оцінки. Маркер «SuperRough» означає, що оцінка береться лише з одного проєкту, маркер «Rough» означає, що оцінка береться з двох проєктів та маркер «Accurate» показують, що оцінка береться з результатів трьох проєктів.

Атрибут «Tags» має тип даних «String» (рядок) та містить ключові слова, які полегшують сортування всіх рядків бази даних, якщо користувач не пам'ятає назву задачі.

Атрибут «CustomSolution» має тип даних «String» (рядок) та містить описформування оцінки.

3.3 Розробка системи

В попередньому пункті було описано таблицю бази даних з властивістю її атрибутів та створено її за допомогою програмного забезпечення «Portal CRM».

Після аналізу двох проєктів, оцінкавартості яких проводилась за допомогою розробленої методики, було заповнено таблицю бази даних. Аналіз цих двох ІТ-проєктів базувався на існуючих задачах «Freshcode», з якими стикалися розробники під час реалізації ІТ-проєктів.

Під час реалізації двох ІТ-проєктів, оцінка вартості яких проводилась за допомогою розробленої методики, розробники зіткнулися з створення таблиць для веб-сайту, це типова задача для ІТ-проєктів. Приклади цих таблиць показані на рисунках 3.2 та 3.3.

Date	Demand	Site/Apps	Ad Request	Sold Impression	Revenue	eCPM	Page RPM	Pageviews	HB Initiations
08/2020	App Next	siteexample.com	12 316	25 655	\$70 542.50	0.78	0.78	44	0.78
08/2020	Google Ad	example.com	54 340	4 620	\$5 012.00	11	0.08	11	0.08
08/2020	Google ADX	siteexample.com	7 856	25 655	\$45 873.20	8.02	4.23	556	4.23
08/2020	App Next	siteexample.com	12 316	25 655	\$70 542.50	0.78	0.78	7	0.78
08/2020	Google Ad	example.com	54 340	4 620	\$5 012.00	11	0.08	11	0.08
08/2020	Google ADX	siteexample.com	7 856	25 655	\$45 873.20	8.02	4.23	98	4.23
08/2020	App Next	siteexample.com	12 316	25 655	\$70 542.50	0.78	0.78	80	0.78
08/2020	Google Ad	example.com	54 340	4 620	\$5 012.00	11	0.08	5	0.08
08/2020	Google ADX	siteexample.com	7 856	25 655	\$45 873.20	8.02	4.23	32	4.23
08/2020	Google Ad	example.com	54 340	4 620	\$5 012.00	11	0.08	78	0.08

Рисунок 3.2 – Таблиця 1-го ІТ-проєкту

Tenant First Name	Tenant Last Name	Tenant E-mail	Rental Street Address	Unit	City	State	ZIP	Country	Date Rentered from	Date to
Billy	Jean	billyjean@gmail.com	4517 Washington Ave. Manchester, Ke	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Floyd	Miles	billyjean@gmail.com	1901 Thornridge Cir. Shiloh, Hawaii 81063	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Billy	Jean	billyjean@gmail.com	4517 Washington Ave. Manchester, Kentucky 39495	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Floyd	Miles	billyjean@gmail.com	1901 Thornridge Cir. Shiloh, Hawaii 81063	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Billy	Jean	billyjean@gmail.com	4517 Washington Ave. Manchester, Kentucky 39495	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Floyd	Miles	billyjean@gmail.com	1901 Thornridge Cir. Shiloh, Hawaii 81063	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Billy	Jean	billyjean@gmail.com	4517 Washington Ave. Manchester, Kentucky 39495	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Floyd	Miles	billyjean@gmail.com	1901 Thornridge Cir. Shiloh, Hawaii 81063	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Billy	Jean	billyjean@gmail.com	4517 Washington Ave. Manchester, Kentucky 39495	1	Minneapolis	MN	55441	Hennepin	11/04/2020	
Floyd	Miles	billyjean@gmail.com	1901 Thornridge Cir. Shiloh, Hawaii 81063	1	Minneapolis	MN	55441	Hennepin	11/04/2020	

Рисунок 3.3 – Таблиця 2-го ІТ-проєкту

Розробники в ході виконання цих задач зіткнулися з наступними підзадачами:

- пошук для реалізації таблиці відповідної бібліотеки, витрачений час - від 1 до 3-х годин;
- підключення бібліотеки, витрачений час - від 1 до 2-х годин;
- налаштування стилів, витрачений час - від 5 до 7-х годин;
- для заповнення таблиці даними написання запитів до бази даних, витрачений час - від 2 до 4-х годин.

В підсумку для реалізації компонентів таблиці 1-го ІТ-проєкту (рис. 3.2) розробник витратив 10 годин, а для подібної задачі 2-го ІТ-проєкту (рис. 3.3) витратив 16 годин.

Вбазу даних по результатам аналізу цієї задачі, а саме реалізації компонентів таблиці, запишемо наступні дані:

- а) атрибут «Name» (ім'я) - «Таблиця»;
- б) атрибут «Images» (зображення) - (рис. 3.2);
- в) атрибут «CustomEstimate» (оцінка) - «10 - 16 годин»;
- г) атрибут «Status» (статус) - «Accurate»;
- д) атрибут «Tags» (тег) - «Table», «Web», «UI»;
- е) атрибут «CustomSolution» (опис):

- 1) пошук для реалізації таблиці відповідної бібліотеки, витрачений час - від 1 до 3-х годин;
- 2) підключення бібліотеки, витрачений час - від 1 до 2-х годин;
- 3) налаштування стилів, витрачений час - від 5 до 7-х годин;
- 4) для заповнення таблиці даними написання запитів до бази даних, витрачений час - від 2 до 4-х годин.

Таким чином після аналізу подібних задач даних двох проєктів, оцінка вартості яких проводилась за допомогою розробленої методики, отримали таблицю заповнену даними зображеними на рисунку 3.4.

No	Name	Images	Custom Estimate	Status	Tags	Custom Solution
1	Twitter authorization		5 - 7 - 8	SuperRough	UI API Web	-
2	API integration		6 - 8 - 12	Rough	BackEnd API	Нужно сначала убедиться, что API существует. Если его нет, то передать это Sales. Если API нашли, то: Первый API research: 2 -
3	Facebook authorization		5 - 7 - 8	SuperRough	UI API Web	-
4	Google authorization		5 - 7 - 8	SuperRough	UI API Web	-
5	Enviroment setup		6 - 8 - 12	Accurate	BackEnd	Mobile: 8 - 16 Web: 6 - 12
6	Login form		4 - 5 - 6	SuperRough	UI Web	UI: 2 - 3 Validation: 2 - 3
7	Bar chart		8 - 10 - 12	Rough	UI Web Dashboard Chart	Connect to a chart library: 2 - 3 Place on the page layout: 0,5 - 1 Simple chart UI: 0,5 - 1 Medium chart UI: 2 - 4 Advanced chart UI: 4
8	Filter Dropdown		4 - 6 - 8	Accurate	UI Filter Web	Simple dropdown UI: 1 - 3 Medium dropdown UI: 2 - 4 Advanced dropdown UI: 3 - 4 Page layout placement: 0,5 - 1 Responsive: 0,5
9	Header		3 - 5 - 8	Rough	UI Web	Header UI: 2 - 4 Responsive: 1 - 3 Search UI: 1 - 2
10	Left Navigation menu		4 - 6 - 8	Rough	UI Web	-

Рисунок 3.4 –Таблиця бази даних з даними

3.4 Аналіз розробленої системи

Проведемо аналіз розробленої методики, яка використовує базу даних, описану вище.

На зараз база даних містить невелику кількість задач. Розробники можуть зіткнутися з задачами, опис яких ще не внесені до бази даних. Тому, треба враховувати, що треба виділити час розробника на формування попередньої оцінки витрат годин на розробку таких задач, опис яких не внесені до бази даних.

В таблиці 3.1 можна побачити результати аналізу методики, що розроблялася, з використанням даних бази даних по 3-му проєкту.

Таблиця 3.1 - Загальна кількість годин після формування оцінки 3-го ІТ-проєкту

Загальна кількість годин розробника		
Оптимістична	Реалістична	Песимістична
198,5	245,5	293

Загалом на формування оцінки 3-го ІТ-проєкту витрачено робочих годин:

- 5 годин – бізнес-аналітик;
- 0,5 години - розробник.

На рисунку 3.5 сформовано графік, де можна побачити порівняння попередньої оцінки та результатів отриманих за кожним завданням 3-го ІТ-проєкту. Завдання 3-го ІТ-проєкту подані по осі x та відсортовані по зростанню тривалості виконання для наочності у графіку. По осі y подано тривалість кожного завдання 3-го ІТ-проєкту.



Рисунок 3.5 - Порівняння попередньої оцінки отриманих результатів за кожним завданням розробника 3-го ІТ-проєкту

До порівняння попередньої оцінки отриманих результатів за кожним завданням 3-го ІТ-проєкту було додано лінію тренду за кожним показником (рис. 3.6).



Рисунок 3.6 - Порівняння попередньої оцінки отриманих результатів за кожним завданням розробника 3-го ІТ-проєкту з лінією тренду за кожним показником

В таблиці 3.2 представлені порівняння попередньої оцінки результатів розробки 3-го ІТ-проєкту. Виходячи з порівняння результатів можна сказати, що розробник вийшов за межі реалістичної оцінки.

Таблиця 3.2 - Порівняння результатів 3-го ІТ-проєкту

Співробітник	Попередня оцінка	Витрачений час на розробку	Різниця
Розробник	245,5	255	9,5

Будуємо графік розрахунку відсотка похибки (рис. 3.7). На графіку зеленим кольором зображено лінію тренду середнього результату, що становить 8%. Це говорить, що середня абсолютна похибка витраченого часу розробника 3-го ІТ-проєкту відносно реалістичної оцінки становить 8%.

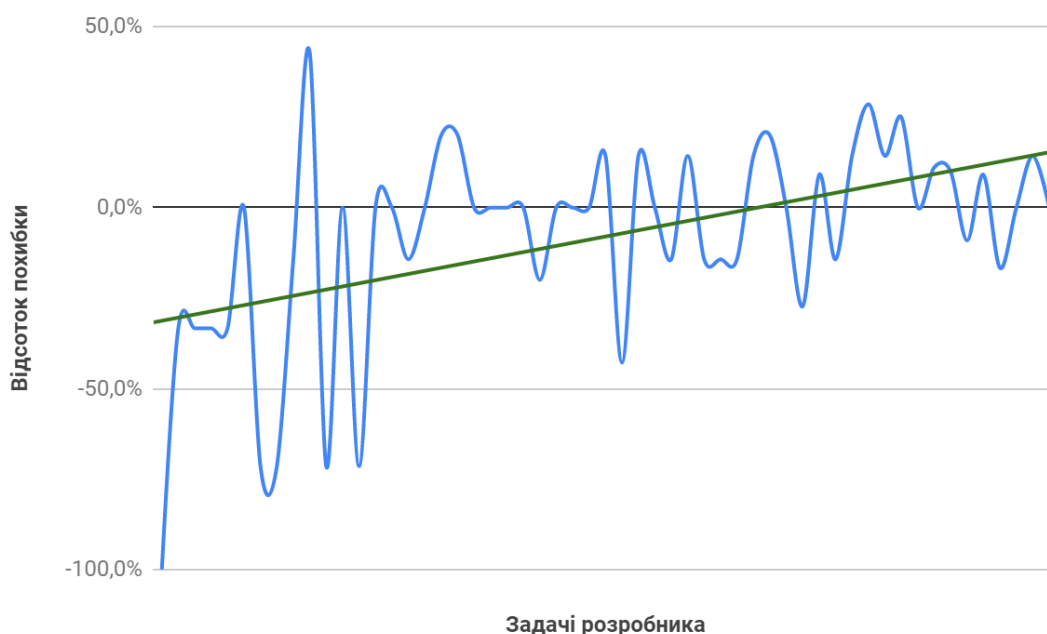


Рисунок 3.7 - Розрахунок відсотка похибки 3-го ІТ-проекту

В таблиці 3.3 подано результати аналізу методики з використанням даних бази даних по 4-му ІТ-проекту.

Таблиця 3.3 - Загальна кількість годин після формування остаточної оцінки по 4-му ІТ-проекту

Загальна кількість годин розробника		
Оптимістична	Реалістична	Песимістична
448	560	680

Загалом на формування оцінки 4-го ІТ-проекту витрачено робочих годин:

- 9 годин—бізнес-аналітик;
- 1 година - розробник.

На рисунку 3.8 подано графік порівняння попередньої оцінки отриманих результатів за кожним завданням розробника 4-го проекту. Завдання 4-го ІТ-проекту подані по осі x та відсортовані по зростанню тривалості виконання для наочності у графіку. По осі y подано тривалість кожного завдання 4-го ІТ-проекту.

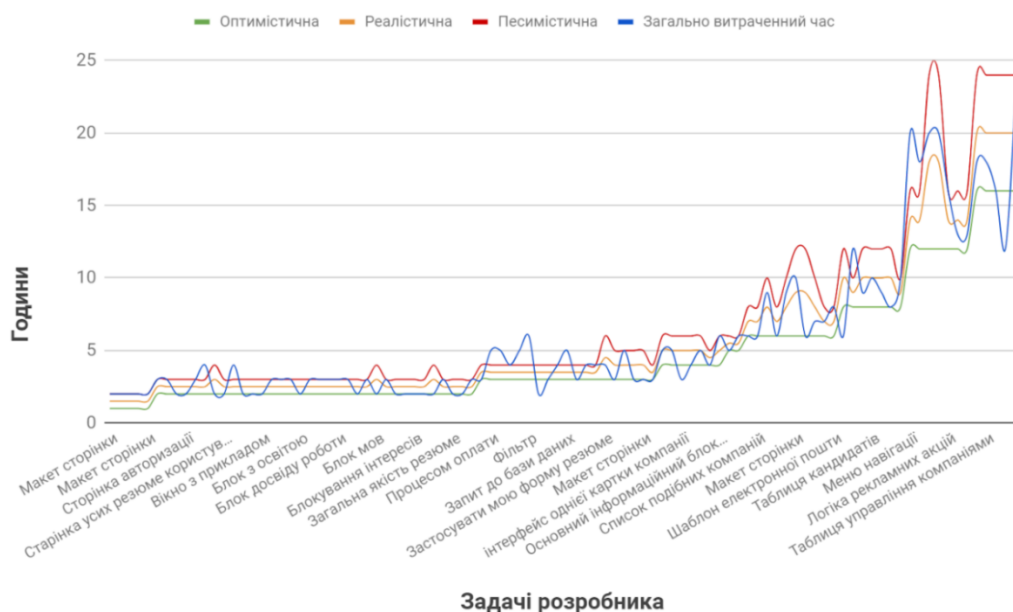


Рисунок 3.8 - Порівняння попередньої оцінки отриманих результатів за кожним завданням розробника 4-го проєкту

До порівняння попередньої оцінки отриманих результатів за кожним завданням 4-го ІТ-проєкту було додано лінію тренду за кожним показником (рис. 3.9).

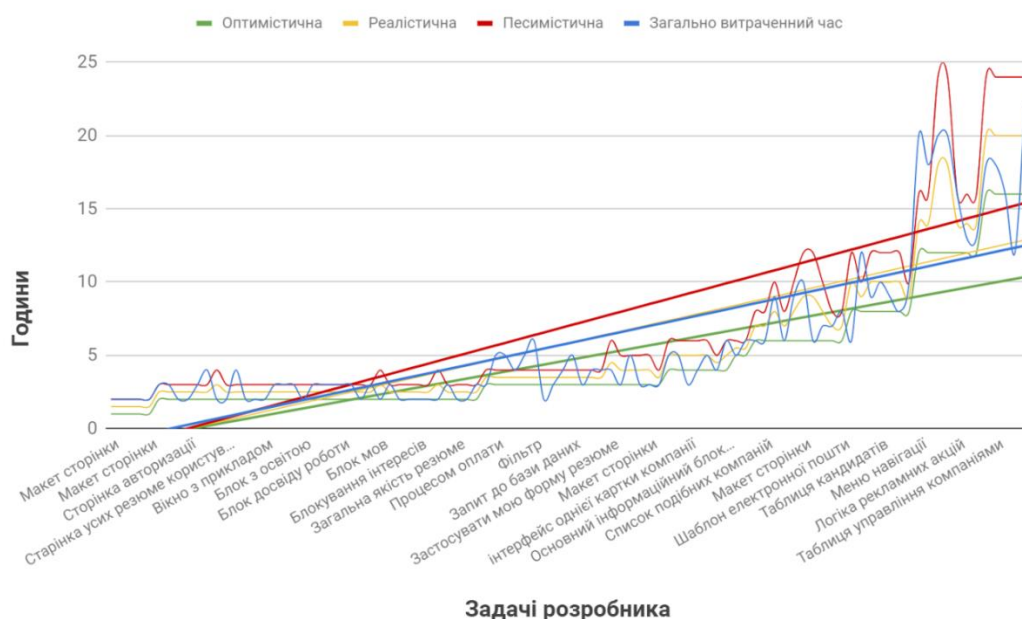


Рисунок 3.9 - Порівняння попередньої оцінки отриманих результатів за кожним завданням 4-го ІТ-проєкту з лінією тренду за кожним показником

В таблиці 3.4 представлені порівняння попередньої оцінки результатів розробки 4-го ІТ-проекту. Виходячи з порівняння результатів можна сказати, що розробник вийшов за межі реалістичної оцінки.

Таблиця 3.4 - Порівняння результатів для 4-го ІТ-проекту

Співробітник	Попередня оцінка	Витрачений час на розробку	Різниця
Розробник	560	564	4

Будуємо графік розрахунку відсотка похибки (рис. 3.10). На графіку зеленим кольором зображено лінію тренду середнього результату, що становить 2,7%. Це говорить, що середня абсолютна похибка витраченого часу розробника 4-го ІТ-проекту відносно реалістичної оцінки становить 2,7%.

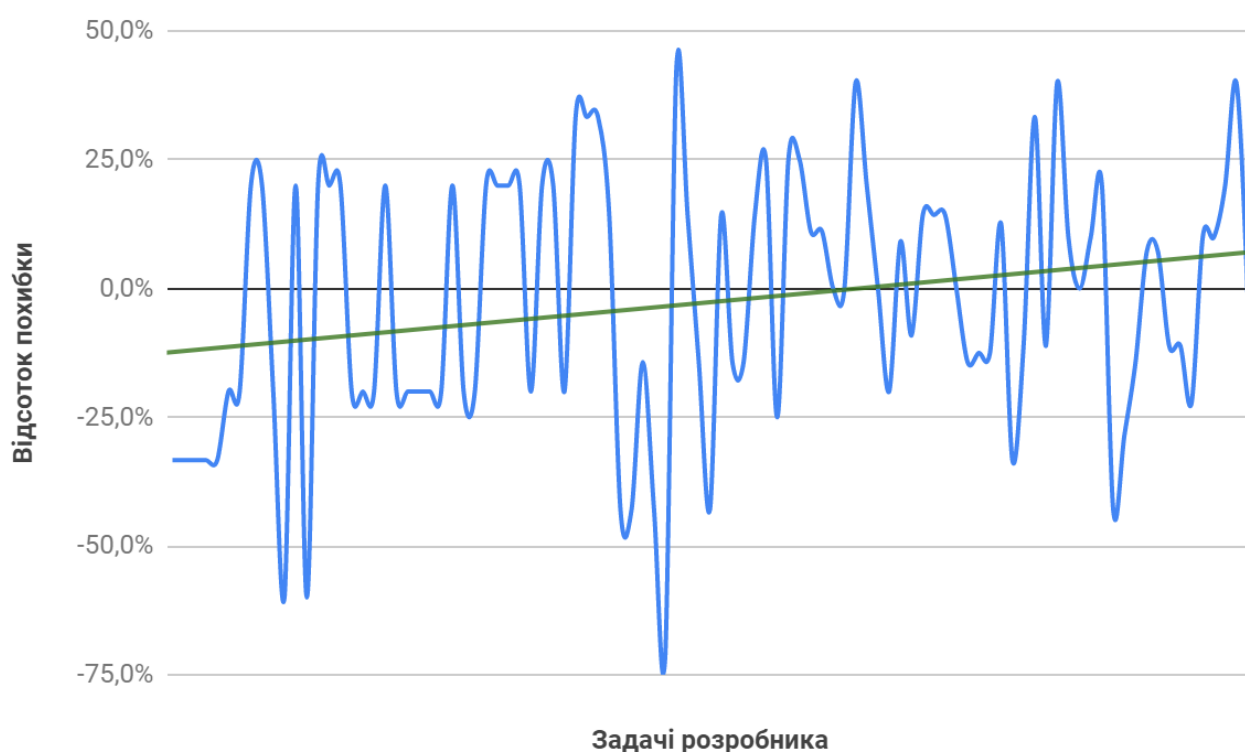


Рисунок 3.10 - Розрахунок відсотка похибки для 4-го проекту

В результаті було протестовано розроблену методика, з даними двох проєктів. Результати показали попередню оцінку кількості годин, які планувалося витратити на розробку, та розрахунок фактичної кількості годин, які були фактично витрачені на розробку ІТ-проєкту. Порівнюючи результати, можна зробити висновок, що відхилення між показникам учасників розробки у попередній та фактичній оцінках кількості годин, витрачених на розробку двох проєктів, не перевищує 8% для реалістичних оцінок проєктів. Це дає можливість стверджувати, що точність розрахунків підвищилася.

Далі необхідно ввести певний норматив - реалістичну оцінку роботи розробника, для порівняння усіх чотирьох проєктів та одержання остаточних висновків. Реалістичну оцінку роботи розробника приймаємо за 500 годин.

В 1-му ІТ-проєкті (реалістична оцінка якого дорівнює 804 години) на отримання оцінки кількості годин витрачених на розробку було витрачено:

- 14 годин – бізнес-аналітика;
- 8 годин - розробника.

Відповідно пропорційно до 500 годин реалістичної оцінки час витрачений на формування оцінки розробника та бізнес-аналітика становить 5 та 8,7 годин відповідно.

В 2-му ІТ-проєкті (реалістична оцінка якого дорівнює 442,5 години) на отримання оцінки кількості годин витрачених на розробку було витрачено:

- 8 годин – бізнес-аналітика;
- 4 годин - розробника.

Пропорційно до 500 годин реалістичної оцінки час витрачений на формування оцінки розробника та бізнес-аналітика 2-го ІТ-проєкту становить 4,5 та 9 годин відповідно.

Для розрахунку середньої кількості годин, витрачених на формування кошторису за розробленою методикою, було використано середню кількість експертних годин, отриману пропорційно розміру проєктів 500 годин реалістичної оцінки роботи розробника. Результат становить:

- 8,8 годин – бізнес-аналітика;

– 4,75 годин - розробника.

Аналогічно для формування оцінки за допомогою розробленої методики та системи було розраховано середню кількість годин роботи бізнес аналітика та розробника та середню похибку часу витраченого розробником відносно реалістичної оцінки. Таким чином, необхідно витратити 0,95 години роботи розробника та 9,1 години роботи бізнес-аналітика для формування оцінки проєкту розміром 500 годин реалістичної оцінки. Похибка результату оцінки при цьому становить 8% більше ніж реалістична оцінка ІТ-проєкту.

Порівняння результатів отриманих в процесі аналізу розробленої методики з використанням бази даних та без її використання подані в таблиці 3.5.

Таблиця 3.5 - Порівняння результатів аналізу з використанням розробленої методики та без

Метрика	Використання методики без бази даних	Використання методики з базою даних	Різниця
Витрачені години роботи бізнес аналітика	8,8 годин	9,1 годин	- 0,3 години
Витрачені години роботи розробника	4,75 годин	0,95 годин	3,8 години
Похибка реального часу розробника відносно реалістичної оцінки	15%	8%	7%

Тестування розробленої системи, дані якої містять на основі результатів реалізації двох проєктів, показало результати розрахунку попередньої оцінки кількості годин, які були заплановані на розробку, та фактичну кількість годин, які витратили розробку двох проєктів.

Таким чином необхідно витратити 0,5 годин роботи розробника та 5 годин роботи бізнес-аналітика на оцінку 1-го ІТ-проєкту, реалістична оцінка якого становить 245,5 години роботи розробника. Також необхідно витратити 1 годин роботи розробника та 9 годин роботи бізнес-аналітика на оцінку 2-го ІТ-проєкту, реалістична оцінка якого дорівнює 560 години роботи розробника.

В результаті аналізу методики формування оцінки кількості годин розробника витрачених на розробку ІТ-проектів з використанням розробленої бази даних робимо висновки, що точність оцінки підвищилась на 7%. Витрати години роботи розробника зменшились приблизно в 5 разів. Витрати години роботи бізнес-аналітика при цьому майже не змінились. Витрати години роботи розробника вході заповнення даними розробленої бази даних будуть прагнути до нуля.

ВИСНОВКИ

Під час виконання дипломного проекту було розроблено методику розрахунку попередньої вартості розробки ІТ-проектів. Було проведено опис визначення оцінки розробки ІТ-проектів, показники якої наступні:

- рівень і якість деталізації вимог;
- фаховий досвід роботи учасників ІТ-проекту;
- ризики веденням проекту.

Сформовано етапи формування оцінки:

- аналіз вимог до проекту;
- оцінювання витрат часу на розробку та розміру продукту;
- оцінювання ризиків, що можуть виникати під час ведення ІТ-проекту;
- аналіз графіка розробки ІТ-проекту;
- формування вартості ІТ-проекту.

Проведено аналіз вимог до проекту, способів визначення вимог та їх класифікація. Визначити вимоги можливо за допомогою інтерв'ю, організації фокус-групи, проведення семінарів, спостереження, опитувальних листів, аналізу документів, системних та користувальницьких інтерфейсів. Проведено аналіз характеристик визначення якості вимог.

Зроблено дослідження методик формування оцінок витрат часу розробників ІТ-проектів, а саме:

- експертна оцінка;
- оцінка за аналогією;
- зверху вниз;
- знизу вгору;
- функціональних точок (FunctionPoint);
- триточкова оцінка або PERT;
- покер оцінка (Pokerestimate);

- методологія Монте-Карло;
- використання програмних засобів COCOMO, SLIM.

Зроблено класифікацію цих методик. Проведено порівняльний аналіз методик формування оцінок витрат часу розробників ІТ-проектів, визначено недоліки та переваги.

Далі проаналізовано методики виявлення ризиків розробки ІТ-проектів, які впливають на вартість ІТ-проектів.

Враховуючи всі переваги та недоліки існуючих методик, було розроблено методику розрахунку попередньої вартості розробки ІТ-проектів та представлена у вигляді алгоритму.

Проведено аналіз розробленої методики на реальних проектах ІТ-компанії «Freshcode». Отримано результати розрахунку попередньої оцінки кількості годин, які планувалися на розробку, та фактична кількість годин, яка витрачена на розробку. Відхилення показників попередніх та фактичних оцінок кількості годин витрачених на розробку не перевищує 15 % за реалістичні оцінки. При цьому необхідно витратити 4,75 годин розробника та 8,8 годин бізнес аналітика для отримання оцінки проекту тривалістю 500 годин реалістичної оцінки роботи розробника. Тестування даної методики дозволили впровадити її в ІТ-компанії «Freshcode».

Дослідження методики дозволило накопичити дані по витратам часу типових задач, присутніх майже у всіх проектах, і оптимізувати методику шляхом розробки комп'ютерної системи, що містить дані про оцінку типових задач.

Дослідження методики з використанням комп'ютерної системи показало, що для розрахунку оцінки проекту тривалістю 500 годин реалістичної оцінки потрібно 9,1 години роботи бізнес-аналітика та 0,95 години роботи розробника. Похибка результату оцінки становитиме 8%.

Розроблена комп'ютерна система дозволила скоротити час, що витрачається розробником на формування оцінки приблизно в п'ять разів і підвищила точність оцінки на 7%, в той час як час роботи бізнес-аналітика залишився практично незмінним.

Під час виконання дипломного проєкту розроблено методику розрахунку попередньої вартості розробки ІТ-проєктів з використанням бази даних, яка впроваджена та використовується в ІТ-компанії «Freshcode».

ПЕРЕЛІК ПОСИЛАНЬ

1. Омана планування [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Омана_планування (дата звернення: 09.09.2023). — Назва з екрана.
2. Darryl K. Forsyth, Christopher D. B. Burt. Allocating time to future tasks: The effect of task segmentation on planning fallacy bias. *Memory & Cognition*. June 2008 [Електронний ресурс]. — Режим доступу: <https://link.springer.com/content/pdf/10.3758/MC.36.4.791.pdf> (дата звернення: 09.09.2023).
3. Roger Buehler, Dale Griffin, and Michael Ross. Exploring the Planning Fallacy. *Journal of Personality and Social Psychology*. 1994. — [Електронний ресурс]. — Режим доступу: http://homepages.se.edu/cvonbergen/files/2013/01/Exploring-the-Planning-Fallacy_Why-People-Underestimate-Their-Task-Completion-Times.pdf (дата звернення: 09.09.2023). — Назва з екрана.
4. Оценка по аналогии [Електронний ресурс]. — Режим доступу: https://project.dovidnyk.info/index.php/home/skolkostoitprogrammnyjproekt/305-ocenka_po_analogii (дата звернення: 09.09.2023). — Назва з екрана.
5. Близнюкова І.О., Данченко О.Б., Тесленко П.О. Аналіз сучасних визначень ІТ-проектів [Електронний ресурс]. — Режим доступу: https://library.krok.edu.ua/media/library/category/materiali-konferentsij/danchenco_0015.pdf (дата звернення: 09.09.2023). — Назва з екрана.
6. Вимоги до програмного забезпечення [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Вимоги_до_програмного_забезпечення (дата звернення: 15.09.2023). — Назва з екрана.
7. Трикутник проекту. [Електронний ресурс]. — Режим доступу: <https://support.microsoft.com/uk-ua/office/трикутник-проекту-8c892e06-d761-4d40-8e1f-17b33fdcf810> (дата звернення: 15.09.2023). — Назва з екрана.

8. Грицюк Ю. І., Далявський В. С. Формалізація процесу управління ризиками розроблення програмного забезпечення. Науковий вісник НЛТУ України. 2018. № 28(11). С. 135-154. <https://doi.org/10.15421/40281124>.

9. Матриця оцінки ризику [Електронний ресурс]. — Режим доступу: <https://studfile.net/preview/5152289/page:2/> (дата звернення: 15.09.2023). — Назва з екрана.

10. Alistair Cockburn People and Methodologies in Software Development [Електронний ресурс]. — Режим доступу: https://alistair.cockburn.us/wp-content/uploads/2018/02/Peopleand_methodologiesinsoftwaredevelopment.pdf (дата звернення: 15.09.2023). — Назва з екрана.

11. Методики Оценивания [Електронний ресурс]. — Режим доступу: <https://tenstep.com.ua/basicI/2.2.1Estimating.htm> (дата звернення: 15.09.2023). — Назва з екрана.

12. Оцінка зверху вниз [Електронний ресурс]. — Режим доступу: <https://uk.laviesurlecaillou.com/1798-top-down-estimating> (дата звернення: 15.09.2023). — Назва з екрана.

13. Оцінка вартості проекту [Електронний ресурс]. — Режим доступу: https://pidru4niki.com/87722/menedzhment/otsinka_vartosti_proektu (дата звернення: 15.09.2023). — Назва з екрана.

14. Функціональні точки (FunctionPoints - FP). [Електронний ресурс]. — Режим доступу: <https://www.maxzosim.com/funktsionalni-tochki/> (дата звернення: 15.09.2023). — Назва з екрана.

15. Триточкова оцінка (PERT) [Електронний ресурс]. — Режим доступу: <https://www.maxzosim.com/tritochkova-otsinka/> (дата звернення: 15.09.2023). — Назва з екрана.

16. Покер планирование, инструкция по применению (PlanningPoker). [Електронний ресурс]. — Режим доступу: <https://grow.org.ua/2016/06/17/planning-poker/> (дата звернення: 15.09.2023). — Назва з екрана.

17. What is The Monte Carlo Simulation? [Електронний ресурс]. — Режим доступу: https://aws.amazon.com/what-is/monte-carlo-simulation/?nc1=h_ls (дата звернення: 15.09.2023). — Назва з екрана.

ДОДАТОК А

Результати дослідження методики

Таблиця А.1 - Декомпозиція та результати проведення оцінки

Поставлені задачі	Кількість годин			Коментарії та припущення
	Оптиміс-тична	Реаліс-тична	Песиміс-тична	
1	2	3	4	5
Планування архітектури	16	20	24	
Розробка бази даних	10	16	24	
Розробка сутності проекту	16	20	24	
Регістрація та авторизація користувача	8	12,5	16	
Розробка інтерфейсу користувача	6.1	9.1	11.1	
Сторінка менеджменту користувачів				
Розробка інтерфейсу користувача	10	15	20	
Інтеграція GDPR	5	12	20	Припущення: система повинна відповідати всім вимогам GDPR.
Створення нового користувача	5	7,5	10	
Пошук користувача	5	7,5	10	Припущення: користувач має можливість пошуку користувача по його імені.
Редагування даних користувача	5	7,5	10	
Редагування паролю	10	15	20	Припущення: користувачу при зміні паролю буде приходити лист з підтвердженням на його пошту.
Видалення користувача	1	2	2,5	
Розробка привілеїв користувача	1	1	1,5	
Сторінка усіх проєктів				
Розробка Back-Endпроєкту	10	15	20	
Список усіх проєктів	5	7,5	10	Припущення: список проєктів однієї організації, та організації мають доступ тільки для своїх проєкт. Сортування списку здійснюється по даті створення.
Менеджмент організацій	16	24	32	
Менеджмент змін у редагуванні проєкту	8	12	16	Припущення: тільки один користувач має можливість редагувати проєкт у єдиний момент часу.
Головна сторінка				
Розробка інтерфейсу користувача	10	15	20	

Продовження таблиці А.1

1	2	3	4	5
Сторінка редагування проєкту				
Розробка інтерфейсу користувача	30	35	40	
Лінія хронології часу	45	47,5	50	Припущення: користувач має можливість змінити розташування порядку зображення, відео або події на лінії хронології часу
Draganddrop	30	35	40	Припущення: користувач має можливість перетягнути і відпустити зображення, відео або подію на лінію хронології часу
Меню з відео	10	15	20	
Меню з зображеннями	10	15	20	
Меню з подіями	5	7,5	10	
Подія "MultipleChoice"				
Попередній перегляд	20	25	30	
Режим редагування	20	25	30	
Режим змінення конфігурацій	10	15	20	
Відповідь в певний час	2,5	3,5	5	
Відображення відповіді	2,5	3,5	5	
Вибір стилю кольору	10	15	20	
Подія "Rating"				
Попередній перегляд	5	7,5	10	
Режим редагування	10	15	20	
Режим змінення конфігурацій	1	2	2,5	
Подія "Infoscreen"				
Попередній перегляд	5	7,5	10	
Режим редагування	10	15	20	
Режим змінення конфігурацій	10	15	20	
Подія "Yes/No"				
Попередній перегляд	5	7,5	10	
Режим редагування	5	7,5	10	
Режим змінення конфігурацій	2,5	3,5	5	
Подія "Summary"				
Попередній перегляд	5	7,5	10	
Режим редагування	20	25	30	
Режим змінення конфігурацій	5	7,5	10	

Продовження таблиці А.1

Попередній перегляд всього проєкту	50	60	70	Припущення: користувач має можливість увімкнути відтворення проєкту та взаємодіяти з ним
Сторінка публікацій проєкту				
Редагування заголовку	5	7,5	10	
Можливість інтеграції	20	25	30	
Іконки взаємодії з проєктом	5	7,5	10	
Можливість опублікувати чи скрити проєкт	5	7,5	10	Припущення: користувач має можливість змінити статус проєкту. Проєкт має статус "публічний" та "неопублікований". Коли користувач установлює статус "публічний", тоді проєкт повинен бути доступний до зовнішніх користувачів і недоступний при статусі "неопублікований"
Сторінка результатів та статистики				
Розробка логіки статистики	30	40	50	
Розробка інтерфейсу користувача	10	15	20	
Експорт результатів до Excel	20	25	30	
Звичайний графік результату	5	12,5	20	
Рейтинговий результат	5	12,5	20	
Середній результат	5	7,5	10	Припущення: середній результат не буде враховувати результати події "Yes/No"
Підготовка до релізу системи	16	24	32	

Таблиця А.2 - Результати роботи над проєктом

Поставлені задачі	Оптимізація	Реалізація	Песимізація	Зафіксовано витрачений час	Розробник А	Розробник Б	Розробник В	Розробник Г
1	2	3	4	5	6	7	8	9
Планування архітектури	16	20	24	16	16			
Розробка бази даних	10	16	24	18	18			
Розробка сутності проєкту	16	20	24	20	20			
Реєстрація та авторизація користувача	8	12,5	16	13			13	
Розробка інтерфейсу користувача	7,5	10	12,5	8,5		8,5		
Сторінка менеджменту користувачів								
Розробка інтерфейсу користувача	10	15	20	15		15		

Продовження таблиці А.2

Інтеграція GDPR	5	12	20	10	10			
Створювання нового користувача	5	7,5	10	9	1		8	
Пошук користувача	5	7,5	10	7			7	
Редагування даних користувача	5	7,5	10	6			6	
Редагування паролю	10	15	20	20	4		16	
Видалення користувача	1	2	2,5	1,5			1,5	
Розробка привілеїв користувача	1	1	1,5	1			1	
Сторінка усіх проєктів								
Розробка Back-Endпроєкту	10	15	20	16			16	
Список усіх проєктів	5	7,5	10	10		10		
Менеджмент організацій	16	24	32	34		10	24	
Менеджмент змін у редагуванні проєкту	8	12	16	11,5	3,5		8	
Головна сторінка								
Розробка інтерфейсу користувача	10	15	20	17,5		17,5		
Сторінка редагування проєкту								
Розробка інтерфейсу користувача	30	35	40	36		36		
Лінія хронології часу	45	47,5	50	48	4	8		36
Draganddrop	30	35	40	35,5		5,5		30
Меню з відео	10	15	20	16,5		2,5	14	
Меню з зображеннями	10	15	20	18,5		2,5	16	
Меню з подіями	5	7,5	10	6		2	4	
Подія "MultipleChoice"								
Попередній перегляд	20	25	30	26	6			20
Режим редагування	20	25	30	24	6			18
Режим змінення конфігурацій	10	15	20	20	6,5	0,5		6
Відповідь в певний час	2,5	3,5	5	3				3
Відображення відповіді	2,5	3,5	5	3				3
Вибір стилю кольору	10	15	20	14		6		8
Подія "Rating"								
Попередній перегляд	5	7,5	10	6				6
Режим редагування	10	15	20	16	3			13
Режим змінення конфігурацій	1	2	2,5	2				2
Подія "Infoscreen"								
Попередній перегляд	5	7,5	10	6				6

Продовження таблиці А.2

Режим редагування	10	15	20	14	2			12
Режим змінення конфігурацій	10	15	20	18	2			16
Подія "Yes/No"								
Попередній перегляд	5	7,5	10	4				4
Режим редагування	5	7,5	10	8				8
Режим змінення конфігурацій	2,5	3,5	5	4				4
Подія "Summary"								
Попередній перегляд	5	7,5	10	6				6
Режим редагування	20	25	30	28	4			24
Режим змінення конфігурацій	5	7,5	10	10		4		6
Попередній перегляд всього проєкту	24	28	32	30				30
Сторінка публікації проєкту								
Редагування заголовку	5	7,5	10	6		3	3	
Можливість інтеграції	20	25	30	20	4		16	
Іконки взаємодії з проєктом	5	7,5	10	4		4		
Можливість опублікувати чи скрити проєкт	5	7,5	10	10				10
Сторінка результатів та статистики								
Розробка логіки статистики	30	40	50	44	10		34	
Розробка інтерфейсу користувача	10	15	20	15		15		
Експорт результатів до Excel	20	25	30	28	10		18	
Звичайний графік результату	5	12,5	20	12	3	2	7	
Рейтинговий результат	5	12,5	20	14	3	4	7	
Середній результат	5	7,5	10	6,5	1,5		5	
Менеджмент проєкту	16	24	32	28				
Загальне значення годин	577	794	1014	816	157,5	156	224,5	271