

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення**

ДОПУСТИТИ ДО
ЗАХИСТУ

завідувача кафедри

Катерина НЕСТЕРЕНКО

« ____ » _____ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“БАКАЛАВР”**

Тема: «Застосунок оцінки ефективності проекту на основі
методики проектування інтелектуальних систем»

Виконавець: Зазимко Віталій Вікторович

Керівник: к.т.н. Ключев Євген Іванович

Нормоконтролер: Варнавський Вячеслав Володимирович

Київ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки та програмної інженерії

Кафедра інженерії програмного забезпечення

Освітній ступень бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Катерина НЕСТЕРЕНКО

«___» _____ 2024 р

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Зазимка Віталія Вікторовича

1. Тема роботи: “Застосунок оцінки ефективності проекту на основі методики проектування інтелектуальних систем”

затверджена наказом ректора від 28.04.2023р. №597/ст

2. Термін виконання проекту: з 19.01.2024 - 29.02.2024

3. Вихідні дані до роботи: Програмний інтелектуальний продукт для оцінки ефективності проекту на основі методики проектування інтелектуальних систем.

4. Зміст пояснювальної записки:

1. Існуючий стан, проблеми, шлях вирішення.
2. Вимоги до програмного застосунку
3. Структура застосунку
4. Прототип за стосунку

5. План графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Розробка плану роботи	19.01.2024-21.01.2024	
2.	Розробка 1 розділу (на основі матеріалів переддипломної практики) Розробка 2 розділу: вимоги до програмного застосунку	22.01.2024-28.01.2024	
3.	Розробка 3 розділу: структура програмного застосунку	29.01.2024-04.02.2024	
4.	Розробка 4 розділу: прототип програмного застосунку	05.02.2024-11.02.2024	
5.	Проходження нормо контролю.Отримання відгуку керівника.Відсилка Пз для перевірки на плагіат.Друк ПЗ.	12.02.2024-18.02.2024	
6.	Передзахист. Допуск до захисту.Рецензування.Здача документів секретарю ДЕК	19.02.2024-23.02.2024	
7.	ДЕК захист	26.02.2024-29.02.2024	

Дата видачі завдання 04.01.2024.

Керівник: к.т.н. Євген КЛЮЄВ

Завдання прийняв до виконання:

Віталій Зазимко

Дата

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Застосунок оцінки ефективності проекту на основі методики проектування інтелектуальних систем»: 44 сторінки, 7 рисунків, 19 таблиць.

ПРОГРАМНИЙ ПРОДУКТ, ОЦІНКА ЕФЕКТИВНОСТІ ПРОЕКТУ, C#, АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ, НЕЧІТКІ МНОЖИНИ.

Об’єкт дослідження – алгоритми для розрахунку ефективності проекту.

Мета дипломної роботи – аналіз існуючих програмних продуктів для оцінки ефективності проекту, виявлення недоліків, створення власного програмного продукту який не буде мати виявлених недоліків.

Метод дослідження – розробка універсального алгоритму та розробка інтелектуального засобу для розрахунку ефективності проекту.

В процесі роботи, був зроблений аналіз предметної області, проаналізовані існуючі ПЗ, розроблений власний алгоритм на основі існуючих алгоритмів, теорії нечітких множин та формули Шортлифа, розроблений інтелектуальний засіб для розрахунку ефективності проекту, який використовує розроблений алгоритм.

Результати роботи можуть бути використані для розрахунку ефективності будь-якого проекту.

Розробка та дослідження проводилися під управлінням ОС Windows 10. Розробка програми проводилася у середовищі VisualStudio 2022 на мові програмування C#.

ABSTRACT

Explanatory note to the thesis "The application of project efficiency assessment based on the method of designing intelligent systems": 44 pages, 7 figures, 19 tables.

SOFTWARE, PROJECT PERFORMANCE EVALUATION, C#, ALGORITHMS AND DATA STRUCTURES, Fuzzy Sets.

The object of research is algorithms for calculating project efficiency.

The purpose of the thesis is to analyze existing software products to assess the effectiveness of the project, identify flaws, and create your own software product that will not have the identified flaws.

The research method is the development of a universal algorithm and the development of an intelligent tool for calculating the effectiveness of the project.

In the process of work, an analysis of the subject area was made, existing software was analyzed, an own algorithm was developed based on existing algorithms, the theory of fuzzy sets and Shortleaf's formula, an intelligent tool was developed for calculating the efficiency of the project, which uses the developed algorithm.

The results of the work can be used to calculate the effectiveness of any project.

The development and research was carried out under Windows 10 operating system. The program was developed in the VisualStudio 2022 environment in the C# programming language.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1	9
ІСНУЮЧИЙ СТАН, ПРОБЛЕМИ, ШЛЯХ ВИРІШЕННЯ.....	9
1.1 Загальні поняття та причини розрахунку ефективності проекту.....	9
1.2 Огляд існуючих ПЗ для розрахунку ефективності проекту. ...	10
1.2.1 Огляд Project Expert.	10
1.2.2 Огляд COMFAR.	13
1.3 Проблеми оцінки ефективності та шляхи вирішення	16
РОЗДІЛ 2	19
ВИМОГИ ДО ПРОГРАМНОГО ЗАСТОСУНКУ.....	19
2.1 Основні вимоги до програмного застосунку.....	19
РОЗДІЛ 3	25
СТРУКТУРА ЗАСТОСУНКУ.....	25
3.1 Виявлення потреб до інтелектуального засобу	25
3.2 Написання універсального алгоритму для розрахунку ефективності проекту	25
РОЗДІЛ 4.....	40
РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЗАСОБУ НА МОВІ C#.....	40

ВСТУП

В наш час побудова будь-якого бізнесу, послуги, ПЗ починається з ідеї, яка надалі переростає у повноцінний проект. Тому не менш важливо робити оцінку свого проекту різними методами, щоб зрозуміти, чи буде проект «успішний».

Проект – це тимчасове підприємство, спрямоване на створення унікального продукту, послуги чи результату. Також до проекту можна віднести сукупність намічених дій та управлінських рішень, спрямованих на досягнення поставленої мети. Стартапи також відносяться до проектів, оскільки це молоді компанії без досвіду оперативної чи проектної діяльності, які працюють над ідеєю з високою часткою ризику та невизначеності.

Приклади проектів:

- розробка ІТ-продукту для вирішення будь-якої проблеми,
- будівництво будинку,
- створення послуги (яка вирішує будь-яку проблему),
- розробка нового пристрою.

Потрібно також враховувати, що будь-який проект починається з ідеї, а закінчується, коли:

- досягнуто мети проекту;
- стає зрозуміло, що досягти поставленої мети неможливо;
- проект більше не актуальний для замовника чи ринку.

Як бачимо, проекти бувають різні і грамотної оцінки, проект може просто достроково закритися. Оцінка дозволяє зрозуміти реальний статус проекту. Також оцінка допомагає команді визначити вектор розвитку проекту та вчасно його скоригувати.

Оцінку проекту можна розділити на оцінку ідеї проекту та оцінку самого проекту. Дані блоки, у свою чергу, складаються з процесів, пов'язаних з оцінкою бюджету, термінів, якості та інших компонентів залежно від рівня

складності проекту. Оцінка ідеї проекту відбувається на етапі, коли формується бізнес-план та створюється концепт продукту. Вона дозволяє керівнику обґрунтувати рішення про запуск проекту та його необхідність для бізнесу. У оцінці ідеї зазвичай беруть участь аналітики, команда маркетингу, менеджер майбутнього проекту. За її результатами приймають рішення про ініціацію проекту, підписують статут проекту та набирають команду. Оцінка самого проекту може відбуватися на всіх етапах, починаючи від планування до етапу завершення. Її завдання – скоригувати перебіг проекту. Після оцінки проекту зазвичай вносять зміни у документацію, може змінитися склад команди чи перелік фіч продукту, або вирішують закрити проект. Якщо команда вирішила продовжувати, то після цього оцінюють потреби у додаткових ресурсах.

РОЗДІЛ 1

ІСНУЮЧИЙ СТАН, ПРОБЛЕМИ, ШЛЯХ ВИРІШЕННЯ

1.1 Загальні поняття та причини розрахунку ефективності проекту.

Поняття ефективності використовуються для оцінки різних процесів та проектів. Оскільки організації, що ведуть проекти, часто стикаються з викликами балансу між часом, вартістю та якістю, вони зацікавлені в максимально ефективному виконанні завдань. Проте, існує різноманітність поглядів на ефективність і результативність серед професіоналів і вчених, що ускладнює їх застосування у проектній діяльності.

Чудові ідеї можуть бути контекст-залежними. Що підходить для одного місця або проекту, може бути не ефективним для іншого. Те саме стосується й проектів. Цілі проекту можуть здатися привабливими, але, якщо вони не узгоджуються зі стратегіями та цілями організації, вони можуть виявитися неприйнятними для проекту.

Одним із плюсів ефективності є її спроможність привести до успіху. Проте, як і в усьому, є й негативні сторони. Навіть якщо проект відповідає стратегії та цілям організації, але ці стратегії не відповідають сучасним реаліям, це може призвести до невдачі проекту.

Цілі проекту повинні бути відповідними стратегії організації, проте це не повинно обмежувати інновації чи розвиток. Менеджери проектів повинні вміти відстоювати актуальність стратегії організації, проте бути відкритими до змін, якщо з'являються нові можливості.

Ефективність полягає в оптимізації ресурсів для досягнення максимального результату. Якщо цілі проекту не враховують обмеження

часу, вартості та зусиль, результат може не відповідати очікуванням клієнта. Тому важливо, щоб цілі були конкретними та відповідали потребам проекту.

1.2 Огляд існуючих ПЗ для розрахунку ефективності проекту.

Для спрощення та прискорення розрахунку ефективності проектів було створено значну кількість різноманітного програмного забезпечення. На ринку найпопулярнішими є кілька програм, які ґрунтуються на традиційних методах оцінки проектів. Розробники цих пакетів регулярно випускають нові, більш гнучкі версії, що щороку збільшує кількість користувачів. Також існують менш відомі програмні продукти, які спираються на електронні таблиці і часто розробляються консалтинговими фірмами.

Розглянемо 2 найпопулярніші: **ProjectExpert** та **COMFAR**

1.2.1 Огляд Project Expert.

Project Expert – це програмне забезпечення, яке призначене для допомоги в оцінці та управлінні проектами. Воно може містити набір інструментів для планування, відстеження виконання завдань, аналізу ресурсів, оцінки вартості та інші функції, спрямовані на підвищення ефективності та успішності проектів.

Аналітична система Project Expert — це програмне забезпечення, спеціалізована на аналізі даних та використанні їх для прийняття рішень в галузі управління проектами. Ця система може забезпечувати інструменти для збору, обробки та візуалізації даних проектів, а також може надавати аналітичні звіти та прогностичні моделі для підтримки керівництва в управлінні проектами та прийнятті стратегічних рішень.

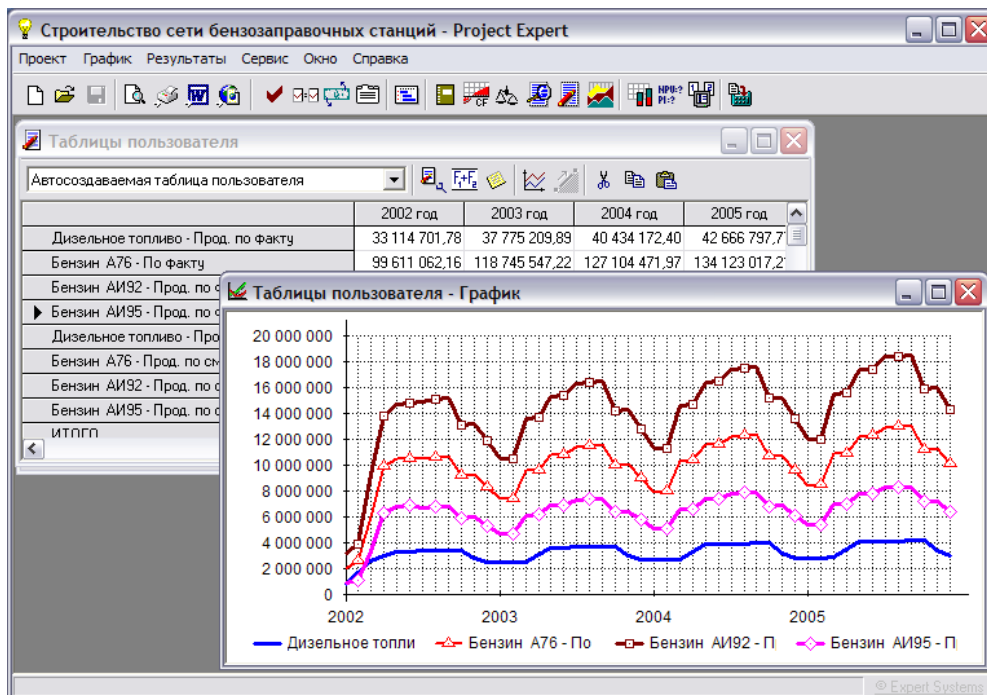


Рисунок 1 – интерфейс Project Expert

Необхідний для розробки та вибору оптимального стратегічного плану розвитку підприємства, аналізу фінансової частини бізнес-плану та оцінки інвестиційних проектів, Project Expert також дозволяє вносити своєчасні корективи до проекту та оновлювати дані. Важливо враховувати, як тимчасові зрушення у календарному плані та динаміка витрат вплинуть на ефективність проекту. Порівняння планових та фактичних результатів проекту має важливе значення для прийняття майбутніх інвестиційних рішень.

Програма надає можливість моделювати діяльність різних галузей та різних масштаб - від малих венчурних компаній до холдингових структур. Вона широко використовується для фінансового моделювання та розробки бізнес-планів у сферах виробництва, послуг у банківському секторі, телекомунікаціях, будівництві, нафтовидобуванні та нафтопереробці, транспорті, хімічній промисловості, переробній та легкій промисловості, машинобудуванні, аерокосмічній галузі та енергетиці.

Оцінка інвестиційних проектів Project Expert.

Після введення усіх даних щодо проекту, таких як звіт про прибутки та збитки, баланс, звіт про рух грошових коштів і т. д., у Project Expert з'являється можливість автоматично розрахувати заплановані показники та ефективність проекту. Окрім оцінки ефективності інвестиційних проектів на основі комплексних показників, програма також надає:

- аналіз чутливості,
- аналіз беззбитковості та
- сценарний аналіз.

Аналіз чутливості

Одним з завдань проектного аналізу є встановлення, наскільки фінансова модель підприємства чутлива до змін різних параметрів. Чим ширший діапазон параметрів, при яких показники ефективності залишаються в прийнятних межах, тим більша міцність проекту, і тим краще він захищений від впливу коливань різних факторів на результати його реалізації.

Дослідження чутливості показників ефективності до зміни ставки дисконтування дозволяє визначити стійкість проекту до коливань фінансового ринку та можливих змін економічних умов діяльності.

Аналіз беззбитковості

Система досліджує взаємозв'язки між витратами та доходами при різних обсягах виробництва. Project Expert вирішує завдання визначення точки беззбитковості для кожного типу продукції у будь-якому періоді та обчислює обсяг продажів, який компенсує всі витрати на виробництво та збут.

Під час проведення аналізу беззбитковості в Project Expert використовується метод, що базується на внеску покриття. Крім кількісних і вартісних значень точки беззбитковості, система визначає запас фінансової міцності та оцінює вплив обраної структури витрат на прибуток,

розраховується операційний важіль. Це дозволяє визначити межу безпеки та рівень підприємницького ризику як для окремих періодів, так і протягом кількох періодів у числовому і графічному форматах.

Сценарний аналіз – порівняння варіантів розвитку

У пакеті Professional, крім проведення аналізу проекту, доступний додатковий модуль, який дозволяє вести багатоваріантний сценарний аналіз та відповідати на запитання "Якщо?".

Під час створення проекту, завданням аналітика є пошук найкращого варіанта розвитку. У Project Expert передбачено можливість швидкого створення кількох варіантів розвитку підприємства для їх порівняння та вибору найкращого. Для створення нового варіанта достатньо задати у відсотках величину відхилень вихідних даних проекту, наприклад, обсягів продажу різних продуктів, їх ціни, ставок податків тощо. У Project Expert можна створити будь-яку кількість варіантів розвитку та провести їх порівняльний аналіз не лише за фінансовими показниками та показниками ефективності інвестицій, але й по всьому спектру даних фінансових звітів, таблиць користувача, деталізації та аналізу варіантів. Результати аналізу можуть бути представлені у вигляді графіків.

1.2.2 Огляд COMFAR.

COMFAR (Computer Model for Feasibility Analysis and Reporting) - це програмне забезпечення, розроблене ЮНІДО (ЮНІДО) для проведення аналізу та оцінки економічної доцільності промислових проектів. Воно дозволяє моделювати фінансові аспекти проектів, такі як витрати, доходи, прибутки, а також проводити чутливість та ризик-аналіз. COMFAR використовується для оцінки фінансової доцільності та прийняття рішень щодо реалізації різних промислових та інфраструктурних проектів.

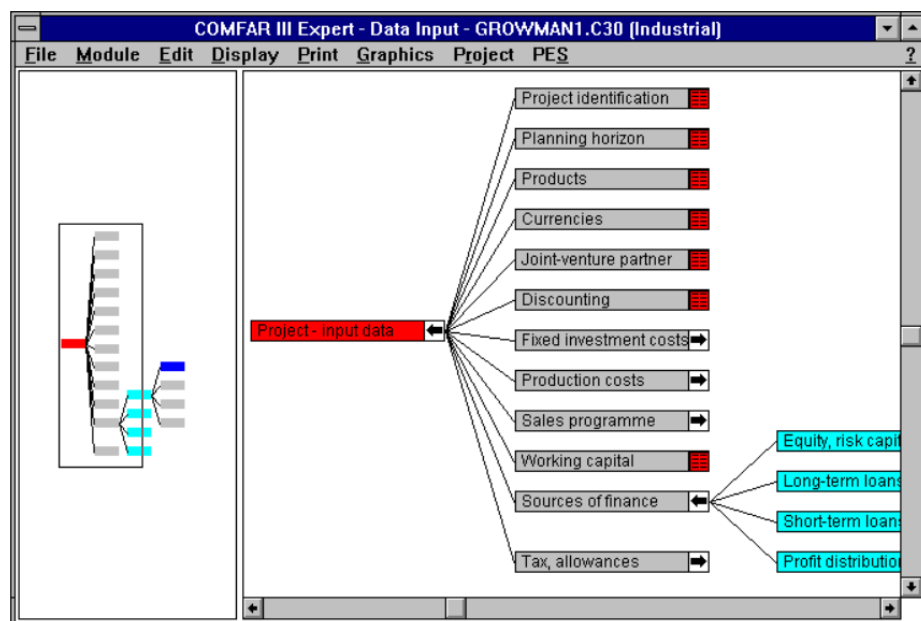


Рисунок 2 – інтерфейс COMFAR

COMFAR (Computer Model for Feasibility Analysis and Reporting) призначений для проведення аналізу та оцінки економічної доцільності промислових проектів. Це програмне забезпечення дозволяє моделювати фінансові аспекти проектів, такі як витрати, доходи, прибутки, а також проводити чутливість та ризик-аналіз. COMFAR використовується для оцінки фінансової доцільності та прийняття рішень щодо реалізації різних промислових та інфраструктурних проектів.

Основні можливості COMFAR включають:

- Аналіз фінансової доцільності проектів: COMFAR дозволяє проводити детальний аналіз фінансових аспектів проектів, включаючи витрати, доходи, прибутки та інші фінансові показники.
- Моделювання витрат та доходів: Програма дозволяє створювати моделі витрат та доходів для проектів у різних сферах діяльності.
- Чутливість та ризик-аналіз: COMFAR допомагає проводити аналіз чутливості проектів до зміни ключових параметрів та оцінювати ризики, пов'язані з реалізацією проекту.

- Генерація звітів: Програма забезпечує можливість генерації звітів та аналітичних документів з результатами аналізу проектів.
- Підтримка прийняття рішень: COMFAR надає інструменти для прийняття обґрунтованих рішень щодо реалізації та фінансування промислових проектів.
- Моделювання різних сценаріїв: Програма дозволяє моделювати різні сценарії розвитку проекту та оцінювати їх вплив на фінансові показники.
- Підтримка різних галузей: COMFAR може бути використаний для аналізу проектів у різних галузях, включаючи промисловість, інфраструктуру та інші сфери діяльності.

Можливості COMFAR дозволяють робити наступне:

Аналіз фінансової доцільності проектів: COMFAR дозволяє оцінити фінансову придатність проектів шляхом аналізу їхніх витрат, доходів та прибутків.

Моделювання фінансових потоків: Ви можете створювати моделі фінансових потоків для проектів, щоб краще зрозуміти їхню економічну динаміку.

Чутливість до змін: COMFAR дозволяє вам визначити, наскільки чутливий ваш проект до зміни різних параметрів та умов.

Аналіз ризиків: Ви можете оцінити ризики, пов'язані з реалізацією проекту, і визначити шляхи їх мінімізації.

Генерація звітів: COMFAR надає засоби для створення звітів і аналітичних документів з результатами аналізу проектів.

Підтримка прийняття рішень: Ви можете використовувати дані, отримані за допомогою COMFAR, для прийняття обґрунтованих рішень щодо фінансування та реалізації проектів.

Моделювання різних сценаріїв: Програма дозволяє вам моделювати різні сценарії розвитку проекту та оцінювати їх вплив на фінансові результати.

Підтримка різних галузей: COMFAR може бути використаний для аналізу проектів у різних сферах діяльності, включаючи промисловість, інфраструктуру, торгівлю та інші. COMFAR III Expert дає користувачеві можливість створювати графічне уявлення коефіцієнтів та структури фінансових потоків, витрат та виручки.

1.3 Проблеми оцінки ефективності та шляхи вирішення

Оцінка ефективності програмних продуктів може стикатися з рядом проблем, серед яких наступні:

Визначення критеріїв ефективності: Визначення чітких та об'єктивних критеріїв, за якими буде оцінюватися ефективність програмного продукту, може бути складним завданням. Часто критерії можуть бути суб'єктивними або неоднозначними.

Вимірювання результатів: Вимірювання реальних результатів від використання програмного продукту може бути важким через складність відстеження впливу програми на бізнесові показники.

Недостатня об'єктивність оцінки: В оцінці ефективності програмного продукту можуть виникати проблеми з об'єктивністю, особливо коли вона здійснюється тими, хто мають особистий інтерес у певному результаті.

Зміна умов: Умови використання програмного продукту можуть змінюватися з часом, що може впливати на його ефективність. Наприклад, технологічні зміни, конкуренція на ринку або зміна потреб користувачів можуть впливати на ефективність програмного продукту.

Суб'єктивність оцінки якості: Оцінка якості програмного продукту може бути суб'єктивною і залежати від індивідуальних переваг та вражень користувачів.

Співвідношення ціни та якості: Оцінка ефективності часто пов'язана з питанням співвідношення між вартістю програмного продукту та його якістю. Деякі продукти можуть бути відносно дорогими, але при цьому не відповідати очікуванням користувачів.

Недоліки методологій оцінки: Деякі методології оцінки ефективності програмних продуктів можуть мати свої обмеження або недоліки, що може впливати на точність отриманих результатів.

Врахування цих проблем може допомогти здійснити більш об'єктивну та інформативну оцінку ефективності програмного продукту.

Для вирішення проблем оцінки ефективності програмних продуктів можна використовувати наступні шляхи:

Чітке визначення критеріїв ефективності: Потрібно чітко визначити критерії ефективності на етапі планування проекту. Критерії повинні бути конкретними, вимірюваними та реалістичними.

Використання об'єктивних методів вимірювання: Важливо використовувати об'єктивні методи вимірювання результатів, такі як метрики якості програмного забезпечення, аналіз даних користувачів та оцінка відповідності вимогам.

Створення стандартизованих процедур: Розробка стандартизованих процедур оцінки ефективності може допомогти уніфікувати процес та забезпечити більш об'єктивні результати.

Залучення експертів: Важливо залучати експертів з різних галузей та досвіду для оцінки програмного продукту з різних точок зору.

Використання автоматизованих засобів: Використання спеціалізованих програмних засобів для вимірювання та аналізу ефективності може забезпечити більш точні та об'єктивні результати.

Постійний моніторинг та аналіз: Важливо здійснювати постійний моніторинг та аналіз результатів впровадження програмного продукту для вчасного виявлення проблем та коригування стратегій.

Збільшення відкритості та зворотного зв'язку: Забезпечення відкритості та зворотного зв'язку з користувачами може допомогти зрозуміти їхні потреби та вподобання та врахувати їх під час оцінки ефективності.

Контроль якості: Впровадження системи контролю якості програмного продукту може допомогти забезпечити високу якість продукту та підвищити його ефективність.

Ці шляхи дозволяють покращити процес оцінки ефективності програмних продуктів та забезпечити більш об'єктивні та інформативні результати.

РОЗДІЛ 2

ВИМОГИ ДО ПРОГРАМНОГО ЗАСТОСУНКУ

2.1 Основні вимоги до програмного застосунку

Програмний застосунок для оцінки ефективності проекту на основі методики проектування інтелектуальних систем повинен відповідати наступним вимогам:

Модульність: Система повинна бути побудована у вигляді модулів, які можна легко масштабувати та модифікувати, забезпечуючи гнучкість та зручність у використанні.

Інтеграція з базами даних: Здатність до інтеграції з різноманітними джерелами даних для збору та аналізу інформації про проект.

Аналітичні засоби: Наявність інструментів аналізу даних, включаючи статистичні методи, машинне навчання та інші аналітичні техніки.

Візуалізація результатів: Можливість візуалізації результатів аналізу у вигляді діаграм, графіків, теплових карт і т. д., що допомагає зрозуміти та інтерпретувати дані.

Інтерактивний інтерфейс: Зручний та інтуїтивно зрозумілий інтерфейс користувача, який дозволяє легко взаємодіяти з системою та вводити необхідні дані.

Підтримка експертних знань: Можливість враховувати експертні знання та досвід у процесі оцінки ефективності проекту.

Автоматизація процесів: Автоматизоване збирання, обробка та аналіз даних для забезпечення швидкості та точності процесу.

Захист даних: Забезпечення конфіденційності та цілісності даних, а також відповідність вимогам щодо захисту персональної інформації.

Документування результатів: Можливість створення звітів та документації з отриманими результатами оцінки ефективності проекту.

Підтримка різних методик: Здатність працювати з різними методиками та моделями оцінки ефективності проектів, зокрема на основі проектування інтелектуальних систем.

Оцінка проекту – визначення його значимості, цінності, виявлення користі та вигоди від його реалізації в порівнянні з альтернативами.

Існує велика кількість методів оцінки проектів.



Рисунок 3 – Методи оцінки проекту

Державні установи використовують різні методи оцінки, а приватні компанії розробляють власні підходи до оцінки, спираючись на світовий та особистий досвід. З цього випливає, що методи оцінки проектів підійдуть як проекту, так стартапа, оскільки стартап по суті є технологічним проектом.

Дослідження показали, що в роботах, які присвячено оцінці ефективності проектів, використовуються методи оцінки, які можна умовно класифікувати, як якісні, кількісні і комбіновані (Рисунок 3)

Розглянемо по одному методу з кожної категорії:

Методи експертних оцінок – це спосіб прогнозування та оцінки майбутніх результатів дій на основі прогнозів фахівців.



Рисунок 4 – Методи експертних оцінок

При застосуванні методу експертних оцінок проводиться опитування спеціальної групи експертів (5–7 осіб) з метою визначення певних змінних величин, необхідних для оцінки досліджуваного питання.

Необхідною умовою ефективного застосування методів експертної оцінки є достатня обізнаність експерта з досліджуваною проблемою, високий рівень ерудиції, здатність його давати чіткі вичерпні відповіді, до того ж експромтом.

Крім того, експерт не повинен бути зацікавленим в тому чи іншому варіанті вирішення поставленої перед ним проблеми. Експерти підбираються за ознакою їх формального професійного статусу – посади, наукового ступеня, стажу роботи та ін. Такий підбір сприяє тому, що в число експертів потрапляють високопрофесійні, з великим практичним досвідом у даній галузі спеціалісти.

Завдяки застосуванню експертних оцінок отримують два види інформації, на підставі якої вирішуються два види завдань різної значимості і на різних рівнях управління:

1. Інформація про поодинокі причинно-наслідкові зв'язки в конкретних умовах місця і часу. Здебільшого цю інформацію одержують в результаті опитування керівників виробничих підрозділів підприємства (бригадири, керівник відділення, начальник цеху) та робітників. Вона призначена для пошуку напрямів підвищення ефективності виробництва і реалізації продукції шляхом встановлення причин непродуктивного використання ресурсів та формування дієвих заходів щодо їх усунення.

2. Інформація про типові взаємозв'язки досліджуваних економічних явищ і процесів. Таку інформацію здатні надати тільки експерти високого класу, професіонали, глибоко обізнані з сутністю та закономірностями прояву вказаних явищ за різних умов господарювання.

Імітаційне моделювання — це метод, що дозволяє будувати моделі процесів, що описують, як ці процеси проходили б насправді.

Таку модель можна «програти» в часі як для одного випробування, так і заданої їх кількості. При цьому результати визначатимуться випадковим характером процесів. За цими даними можна отримати достатньо стійку статистику.

Імітаційне моделювання — це метод дослідження, заснований на тому, що система, яка вивчається, замінюється імітатором і з ним проводяться експерименти з метою отримання інформації про цю систему.

Серед методів імітаційного моделювання найбільш розповсюдженим є метод Монте-Карло. Даний метод базується на застосуванні імітаційних моделей, які дозволяють створити множину сценаріїв, що погоджуються з заданими обмеженнями на вихідні змінні.

Метод Монте-Карло найбільш повно відображає всю гаму невизначеностей, з якою може бути пов'язаний реальний проект. Через початково задані обмеження цей метод дозволяє врахувати усю інформацію, яка є в розпорядженні аналітика проекту. Перевагою даного методу є й

можливість отримання «інтервальних» (а не «точкових») значень аналізованих показників.

На практиці даний метод може бути застосований тільки з використанням комп'ютерних програм, які дозволяють описувати прогнозні моделі та генерувати велику кількість випадкових сценаріїв.

Методи багатовимірного статистичного аналізу базуються на методах статистики (теорія ймовірностей, математична статистика, загальна теорія статистики) та методах вищої математики (аналітична геометрія, матрична алгебра, багатомірний математичний аналіз) і використовують методи ймовірнісного аналізу даних і методи логіко-геометричного напрямку (Рисунок 5).



Рис. 5 – Багатовимірний статистичний аналіз

Отже, методи багатовимірного статистичного аналізу дозволяють розв'язувати різноманітні задачі дослідження багатовимірних процесів: групування, стиснення даних, моделювання складних систем, оцінювання параметрів багатовимірних сукупностей та інші. Це дозволяє

використовувати отримані результати для прийняття ефективних управлінських рішень.

РОЗДІЛ 3

СТРУКТУРА ЗАСТОСУНКУ

3.1 Виявлення потреб до інтелектуального засобу

Розглянуті програмні засоби, зокрема "ProjectExpert", хоч і відображають певні аспекти ефективності проекту, проте мають свої обмеження у формі обмеженого числа метрик, які ґрунтуються виключно на фінансових показниках. Це може призвести до неповного та обмеженого розуміння ефективності проекту, оскільки інші важливі аспекти, такі як технічна складність, соціальні наслідки, екологічні впливи та інші, не враховуються.

Враховуючи це, важливо розробити програмне забезпечення, яке було б універсальним і здатним аналізувати широкий спектр аспектів проекту. Таке програмне забезпечення має включати не лише фінансові метрики, але й інші ключові параметри, які відображають різноманітні аспекти проекту. Наприклад, воно може охоплювати технічні показники, соціальні виміри, ризики, чинники сталого розвитку та інші фактори, які впливають на успішність проекту.

Таке програмне забезпечення повинно бути розроблене з урахуванням потреб різних сфер діяльності та типів проектів, забезпечуючи гнучкість та налаштованість для відображення специфіки кожного конкретного проекту. В результаті, користувачі зможуть отримати більш повний та об'єктивний аналіз ефективності своїх проектів та зробити відповідні зміни для покращення їх результативності..

3.2 Написання універсального алгоритму для розрахунку ефективності проекту

У цьому підрозділі, ми розглянемо створення універсальної методології або узагальненого алгоритму для оцінки будь-якого проекту, будь то ІТ проект, або проект будівництва будівлі та інші.

Слід зазначити, що ця методологія поділяється на два типи, коли розглядаємо:

- **Детерміновану** – або точну систему, де ми точно знаємо, що брехня, а що істина. Значення, що використовуються: 0 - не правда, 1 - правда.
- **Не детерміновану** – чи ймовірнісну систему, де ми заздалегідь не можемо знати, що є істина, а що ні. Тому для оцінки тих чи інших характеристик використовуємо числа проміжку від 0 до 1.

Із цього випливає, що нам потрібно буде використати Експертний Метод. Цей метод зводиться до збору, обробки та аналізу думок та оцінок кількох експертів. Також потрібно буде використовувати формулу Шортліфа, а саме знаходження коефіцієнта впевненості для досліджуваної системи. Цей коефіцієнт вказує на ступінь впевненості експерта у правильності висновку або оцінки системи. Використання формули Шортліфа дозволяє кількісно визначити ступінь впевненості експерта у правильності висновку системи.

Апарат теорії нечітких множин в контексті вашого дослідження може включати в себе визначення нечітких правил для оцінки різних аспектів ефективності проекту на основі методики проектування інтелектуальних систем. Можливість врахування нечіткості та невизначеності у визначенні оцінок може поліпшити точність результатів та робити їх більш придатними для реальних умов. До цього варто додати апарат теорії нечітких множин.

З одержаною інформацією можемо скласти узагальнений алгоритм.

Узагальнений алгоритм складається з наступних кроків:

1. Опис рівнів досліджуваної системи.
2. Визначення, чи є система детермінованою чи ймовірнісною.

3. Опис всіх можливих характеристик системи у вигляді множин та елементів цих множин.

4. Складання правил, якими керуватимуться дані множини та їх елементи.

5. Складання таблиць істинності для множин і правил.

6. Складання графа всім рівнів.

7. Якщо система не детермінована, тоді слід задіяти Експертний метод для того, щоб точно знати значення характеристик на нижньому рівні. Якщо система детермінована, тоді слід запровадити значення для характеристик.

8. Опис бази знань, всіх значень, які повинні туди входити. Здебільшого туди входить значення: Міра довіри – величина, яка визначає рівень довіри користувача даної системи.

9. Знаходження загального значення кожної групи характеристик.

10. Використання формули ШОРТЛІФА на **КОЖНОМУ РІВНІ**, саме знаходження коефіцієнта впевненості кожної групи характеристик.

11. З отриманими коефіцієнтами впевненості визначити яка саме група характеристик максимально описує цю систему.

12. Використання коефіцієнта впевненості на верхньому рівні з групи, яка максимально описує систему, як результат обчислень.

Для того, щоб точно зрозуміти, про що йдеться, розглянемо приклад не детермінованої системи, а саме: підрахуємо значення фактора універсальності. За прикладом буде видно, як ми можемо використовувати дану методологію для інших не детермінованих систем.

Варто уточнити, що всі характеристики для об'єкта, що досліджується, беруться з ГОСТу. Нижче представлені дані характеристики:

Таблица 9

«Оценочные элементы фактора «универсальность»

Код элемента	Наименование	Метод оценки	Оценка
Г0101	Оценка числа потенциальных пользователей	Экспертный	0—1
Г0102	Оценка числа функций ПС	То же	0—1
Г0103	Насколько набор функций удовлетворяет требованиям пользователя	»	0—1
Г0104	Насколько возможности программ охватывают область решаемых пользователем задач	»	0—1
Г0105	Возможность настройки формата выходных данных для конкретных пользователей	»	0—1
Г0201	Наличие схемы иерархии модулей программы	»	0—1
Г0202	Оценка независимости модулей	»	0—1
Г0203	Оценка числа уникальных элементов/реквизитов	»	0—1
Г0204	Используется ли в текущем вызове модуля информация, полученная в предыдущем вызове	»	0—1
Г0205	Оценка организации точек входа и выхода модуля	»	0—1
Г0206	Наличие описания атрибутов модуля	»	0—1
Г0301	Оценка программ по числу переходов и точек ветвления	»	0—1

Рисунок 6 – характеристики для фактора Универсальность

Також представлена схема всіх елементів досліджуваної системи.



Рисунок 7 – схема для фактора Универсальность

Діятимемо строго за алгоритмом для того, щоб побачити, що цей алгоритм працює.

Крок 1. Опис рівнів.

Як бачимо з Рисунок 7, в нашій системі є 4 рівні:

- Рівень Фактору
- Рівень Критерію
- Рівень Метрики
- Рівень оціночних елементів

Крок 2. Визначення, чи є система імовірнісною чи детермінованою.

У цій системі використовуються значення проміжку від 0 до 1. Тому ця система буде ймовірнісною. Так як не можна точно оцінити ту чи іншу характеристику. (Додати інформації)

Крок 3. Опис множин та елементів цих множин.

Як бачимо, у нашій системі 3 метрики. До кожної метрики є своя група оціночних елементів. Тому кожна метрика буде представляти безліч, а кожен оцінний елемент з Рисунокб буде елементом для певної множини.

1. Множина «широта охопту функцій»:

$H = \{ h_1, h_2, h_3, h_4, h_5 \}$ – де h_1, h_2, h_3, h_4, h_5 – елементи множини H

Опис елементів множини H

h_1	Оцінка кількості потенційних користувачів
h_2	Оцінка числа функцій ПЗ
h_3	Наскільки набір функцій удів. вимоги користувача
h_4	Наскільки можл. ПЗ охоплюють область реш. Задач
h_5	Можливість налаштування формату вихід. даних

2. Множина «простота архітектури проекту»:

$P = \{ p_1, p_2, p_3, p_4, p_5, p_6 \}$ – де $p_1, p_2, p_3, p_4, p_5, p_6$ – елементи множини P

Опис елементів множини P

p_1	Наявність схеми ієрархії модулів
p_2	Оцінка незалежності модулів
p_3	Оцінка числа унік. елементів
p_4	Чи використовується в модулі інф. з попер. Модуля
p_5	Оцінка організації точок входу та виходу
p_6	Наявність опису атрибутів модуля

3. Множина «складність архітектури проекту»:

$S = \{ s_1 \}$ – де s_1 – елементи множини S

Опис елементів множини S

s_1	Оцінка програм за кількістю переходів та точок розгалуження
-------	---

Крок 4. Опис правил.

Після того, як множини та їх елементи складені, слід підготувати правила для цих множин. Також потрібно відразу скласти правила для вищих рівнів.

Правила рівняючі елементів:

Правила для множини H :

$$R_1 = \exists(H), h_1 \in H, h_1 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_2 = \exists(H), h_2 \in H, h_2 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_3 = \exists(H), h_3 \in H, h_3 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_4 = \exists(H), h_4 \in H, h_4 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_5 = \exists(H), h_5 \in H, h_5 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 1$$

Правила для множини P :

$$R_6 = \exists(P), p_1 \in P, p_1 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_7 = \exists(P), p_2 \in P, p_2 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_8 = \exists(P), p_3 \in P, p_3 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 1$$

$$R_9 = \exists(P), p_4 \in P, p_4 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_{10} = \exists(P), p_5 \in P, p_5 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

$$R_{11} = \exists(P), p_6 \in P, p_6 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 0$$

Правила для множини S:

$$R_{12} = \exists(S), s_1 \in S, s_1 \in \exists(\{0.95, 0.96 \dots 1\}) \rightarrow 1$$

Правила переходу до рівня метрик:

$$R_{13} = ((\neg R_1 \cap \neg R_2 \cap \neg R_3 \cap \neg R_4 \cap R_5) \rightarrow 1) \rightarrow \exists(H)$$

$$R_{14} = ((\neg R_6 \cap \neg R_7 \cap R_8 \cap \neg R_9 \cap \neg R_{10} \cap \neg R_{11}) \rightarrow 1) \rightarrow \exists(P)$$

$$R_{15} = ((R_{12}) \rightarrow 1) \rightarrow \exists(S)$$

$$R_{16} = ((R_{13} \cap R_{14} \cap R_{15})) \rightarrow 1$$

Правила рівня метрик:

$$R_{17} = ((R_{16} \cup R_{13}) \rightarrow 1)$$

$$R_{18} = ((R_{16} \cup R_{14}) \rightarrow 1)$$

$$R_{19} = ((R_{16} \cup R_{15}) \rightarrow 1)$$

Правила переходу на рівень критеріїв:

$$R_{20} = (R_{17} \cup R_{18} \cup R_{19}) \rightarrow 1)$$

Правила рівня критеріїв:

$$R_{21} = (R_{20}) \rightarrow 1)$$

Правила рівня факторів:

$$R_{22} = (R_{21}) \rightarrow 1)$$

Крок 5. Опис таблиць істинності.

Використовуючи раніше описані правила та множини, можемо скласти таблиці істинності:

Таблиця істинності для множини H :

H	h_1	h_2	h_3	h_4	h_5	\rightarrow
R_1	1	0	0	0	0	0
R_2	0	1	0	0	0	0
R_3	0	0	1	0	0	0
R_4	0	0	0	1	0	0
R_5	0	0	0	0	1	1

Таблиця істинності для множини P :

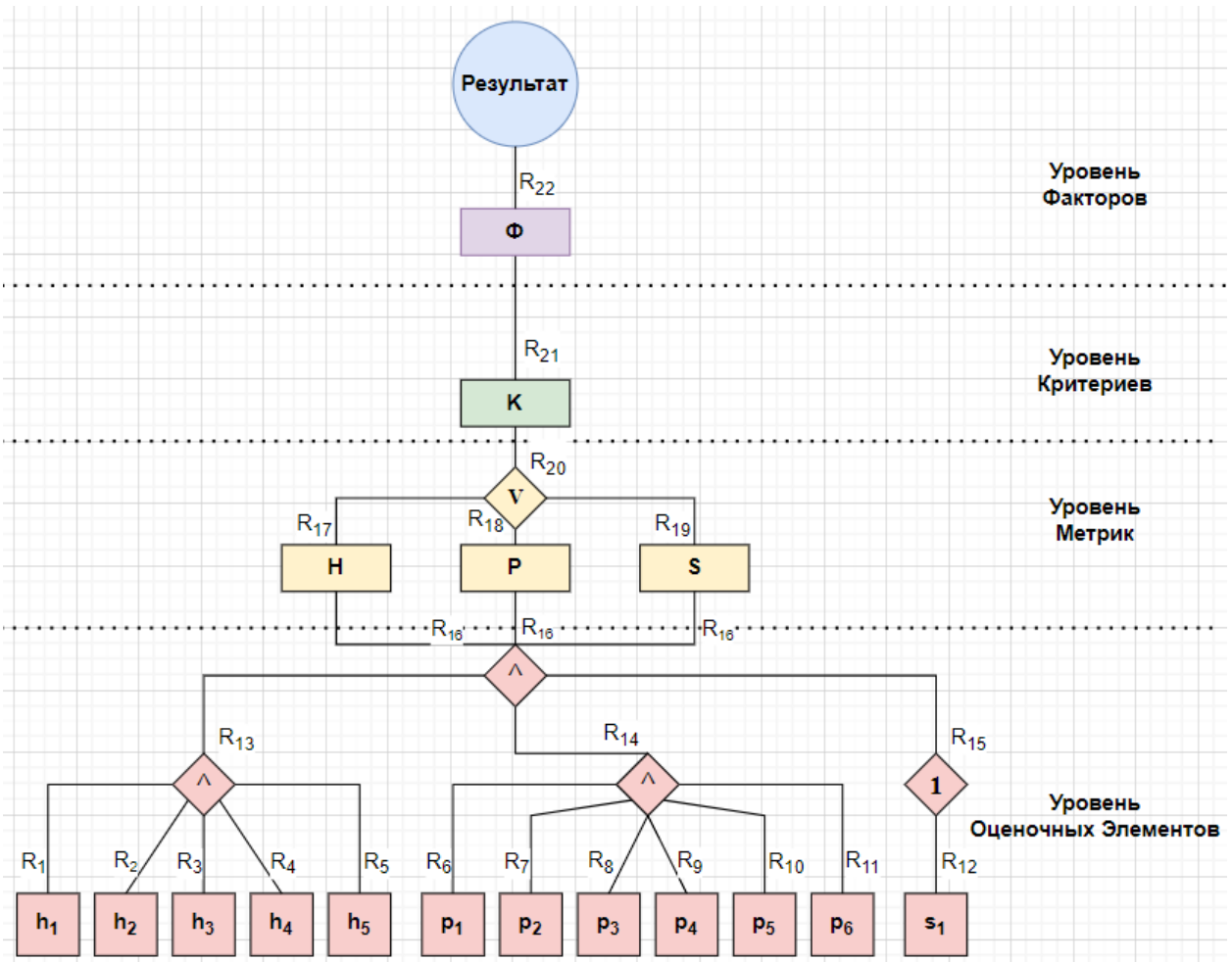
P	p_1	p_2	p_3	p_4	p_5	p_6	\rightarrow
R_6	1	0	0	0	0	0	0
R_7	0	1	0	0	0	0	0
R_8	0	0	1	0	0	0	1
R_9	0	0	0	1	0	0	0
R_{10}	0	0	0	0	1	0	0
R_{11}	0	0	0	0	0	1	0

Таблиця істинності для множини S :

S	s_1	\rightarrow
R_{12}	1	1

Крок 6. Упорядкування графа

Тепер уже маючи всі можливі дані, а саме: множини та їх елементи, правила, таблиці істинності, можемо скласти граф, який буде графічним поданням нашої системи:



Де –

Н – безліч, або метрика «широта охоплення функцій»

$h_1 - h_5$ – елементи множини Н, або оціночні елементи метрики «широта охоплення функцій»

Р – безліч, або метрика «простота архітектури проекту»

$p_1 - p_6$ – елементи множини Р, або оціночні елементи метрики «простота архітектури проекту»

S – безліч, або метрика «складність архітектури проекту»

s_1 – елемент множини S , або оціночні елементи метрики «складність архітектури проекту»

K – Значення на рівні критерію Гнучкість

Φ – Значення на рівні фактора Універсальність

Крок 7. Оцінки експертів

Оскільки система у нас імовірна чи не детермінована, нам слід звернутися до Експертного методу. За основу будуть взяті оцінки 4 експертів. Варто зазначити, що група експертів оцінюватиме лише оціночні елементи.

Елементи для метрики «широта охоплення функцій»

h_1	Оцінка кількості потенційних користувачів
h_2	Оцінка числа функцій ПЗ
h_3	Наскільки набір функцій удів. вимоги користувача
h_4	Наскільки можл. ПЗ охоплюють область реш. Задач
h_5	Можливість налаштування формату вихід. даних

Оцінки для метрики «широта охоплення функцій»

Елемент множини \ Експерт	№1	№2	№3	№4	Сер. Ариф. Для оцінок експертів	Сер. Ариф. для групи
h_1	0,3	0,2	0,11	0,16	0,1925	0,1215
h_2	0	0,1	0,1	0,15	0,0875	
h_3	0	0,12	0,13	0,15	0,1	
h_4	0,11	0,1	0,12	0,09	0,105	
h_5	0,14	0,13	0,1	0,12	0,1225	

Елементи для метрики «простота архітектури проекту»

p_1	Наявність схеми ієрархії модулів
p_2	Оцінка незалежності модулів
p_3	Оцінка числа унік. елементів
p_4	Чи використовується в модулі інф. з попер. Модуля
p_5	Оцінка організації точок входу та виходу
p_6	Наявність опису атрибутів модуля

Оцінки для метрики «простота архітектури проекту»

Елемент множини \Експерт	№1	№2	№3	№4	Сер. Ариф. Для оцінок експертів	Сер. Ариф. для групи
p_1	1	0,85	0,9	1	0,9375	0,95
p_2	0,9	1	0,95	1	0,9625	
p_3	0,99	0,99	0,99	0,99	0,99	
p_4	0,85	0,9	0,86	1	0,9025	
p_5	1	0,96	1	0,84	0,95	
p_6	1	1	0,95	0,88	0,9575	

Елементи для метрики «складність архітектури проекту»

s_1	Оцінка програм за кількістю переходів та точок розгалуження
-------	---

Оцінки для метрики «складність архітектури проекту»

Елемент множини \Експерт	№1	№2	№3	№4	Сер. Ариф. Для оцінок експертів	Сер. Ариф. для групи
s_1	0,2	0,21	0,23	0,21	0,2125	0,2125

На цьому кроці слід описати всі можливі значення для всіх рівнів:

Рівень оціночних елементів:

Min_1	0.4
Max_1	0.8

де:

Min_1 –Мінімальне значення,

Max_1 –Максимальне значення

Крок 9. Знаходження значення кожної групи характеристик.

У прикладі групи характеристик представлені як метрик. Тому, використовуючи оцінки Експертів, можемо знайти значення групи оціночних елементів, а саме метрик.

Оцінки для множини Н, або метрики «широта охоплення функцій»

Елемент множини\Експерт	Сер. Ариф. Для оцінок експертів	Сер. Ариф. для групи
h_1	0,1925	0,1215
h_2	0,0875	
h_3	0,1	
h_4	0,105	
h_5	0,1225	

Оцінки для множини P , або метрики «простота архітектури проекту»

Елемент множини \ Эксперт	Сер. Ариф. Для оцінок експертів	Сер. Ариф. для групи
p_1	0,9375	0,95
p_2	0,9625	
p_3	0,99	
p_4	0,9025	
p_5	0,95	
p_6	0,9575	

Оцінки для множини S , або метрики «складність архітектури проекту»

Елемент множини \ Эксперт	Сер. Ариф. Для оцінок експертів	Сер. Ариф. для групи
s_1	0,2125	0,2125

Крок 10. Знаходження коефіцієнта впевненості.

На цьому кроці ми передаємо отримані значення формулу Шортлифа:

Для **рівня Оціночних Елементів**, а саме нижнього рівня:

$$KU = MD - MND \rightarrow MD - M_1$$

Знаходимо серед. ариф. для всіх груп оціночних елементів:

H	0,1215
P	0,95
S	0,2125

$$(0,1215 + 0,95 + 0,2125) / 3 = M_1 = 0.4321$$

Перевіряємо, чи знаходиться значення M_1 у допустимому діапазоні:
 Мінімальне та максимальне допустиме значення для рівня оціночних елементів знаходиться у Базі Знань:

M_1	0.4321
Min_1	0.4
Max_1	0.8

Значення знаходиться в допустимому діапазоні, тому беремо його і додаємо у формулу Шортлифа, як міру недовіри.

Міра довіри для нашого випадку, на рівні оціночних елементів дорівнюватиме середовищ. ариф. для кожної окремої групи.

$$KB_H = 0.1215 - 0.4321 = -0.3065$$

$$KB_P = 0.95 - 0.4321 = 0.522$$

$$KB_S = 0.225 - 0.4321 = 0.2155$$

Звідси беремо Максимальний KB для рівня оціночних елементів та передаємо його на рівень Метрик:

$$KB_P = 0.522$$

Для **рівня Метрик**, а саме другого рівня:

$$KB_M = MD - MND \rightarrow KU_P - \text{Сред. Ариф. } (M_1)$$

Перевіряємо, чи входить KB_P у допустимий діапазон:

KB_P	0.522
Min_1	0.4
Max_1	0.8

Так як KB входить у діапазон, ми можемо його підставити міру довіри для знаходження KB для рівня метрик:

$$KB_M = 0.522 - 0.4321 = 0.0899$$

$$KB_M = 0.0899$$

Крок 13. Знаходження результату:

З попереднього кроку знаємо, що значення $KB_M = 0.0899$

Але оскільки розгалужень більше немає, ми можемо передати це значення на рівень вище. І використати його як результат.

Тому наш Фактор Універсальності ПЗ дорівнюватиме 0.0899

РОЗДІЛ 4

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЗАСОБУ НА МОВІ C#

```
using System;
using System.IO;

class Program
{
    static void Main(string[] args)
    {
        double totalCost = GetTotalCostFromUser();
        double expectedBenefit = GetExpectedBenefitFromUser();

        if (!ValidateData(totalCost, expectedBenefit))
        {
            Console.WriteLine("Invalid input. Please enter valid data.");
            return;
        }

        double efficiencyCoefficient = CalculateEfficiencyCoefficient(totalCost, expectedBenefit);
        DisplayEfficiency(efficiencyCoefficient);

        double netPresentValue = CalculateNPV(totalCost, expectedBenefit);
        double roi = CalculateROI(totalCost, expectedBenefit);
        double irr = CalculateIRR(totalCost, expectedBenefit);

        DisplayAdditionalMetrics(netPresentValue, roi, irr);

        double convertedTotalCost = ConvertToCurrency(totalCost);
```



```

double convertedExpectedBenefit = ConvertToCurrency(expectedBenefit);

double convertedNPV = ConvertToCurrency(netPresentValue);

double convertedROI = ConvertToCurrency(roi);

double convertedIRR = ConvertToCurrency(irr);

    DisplayCurrencyConversion(convertedTotalCost, convertedExpectedBenefit,
convertedNPV, convertedROI, convertedIRR);

    SaveDataToFile(totalCost, expectedBenefit, netPresentValue, roi, irr);

    LoadDataFromFile();

    Console.ReadLine(); // Зупиняємо виконання програми, щоб користувач міг
переглянути результат
}

static double GetTotalCostFromUser()
{
    Console.WriteLine("Enter the total cost of the project:");
    return double.Parse(Console.ReadLine());
}

static double GetExpectedBenefitFromUser()
{
    Console.WriteLine("Enter the expected benefit of the project:");
    return double.Parse(Console.ReadLine());
}

static bool ValidateData(double totalCost, double expectedBenefit)
{
    return totalCost > 0 && expectedBenefit > 0;
}

```

```
}
```

```
static double CalculateEfficiencyCoefficient(double totalCost, double expectedBenefit)
```

```
{
```

```
    return expectedBenefit / totalCost;
```

```
}
```

```
static void DisplayEfficiency(double efficiencyCoefficient)
```

```
{
```

```
    Console.WriteLine("Efficiency coefficient of the project: " + efficiencyCoefficient);
```

```
    if (efficiencyCoefficient > 1)
```

```
    {
```

```
        Console.WriteLine("The project is efficient.");
```

```
    }
```

```
    else if (efficiencyCoefficient == 1)
```

```
    {
```

```
        Console.WriteLine("The project is worth considering.");
```

```
    }
```

```
    else
```

```
    {
```

```
        Console.WriteLine("The project needs to be reviewed or rejected.");
```

```
    }
```

```
}
```

```
static double CalculateNPV(double totalCost, double expectedBenefit)
```

```
{
```

```
    // Реалізуйте розрахунок чистої присутньої вартості тут
```

```
    return 0.0;
```

```
}
```

```

static double CalculateROI(double totalCost, double expectedBenefit)
{
    // Реалізуйте розрахунок прибутку від інвестицій тут
    return 0.0;
}

static double CalculateIRR(double totalCost, double expectedBenefit)
{
    // Реалізуйте розрахунок внутрішньої норми прибутку тут
    return 0.0;
}

static void DisplayAdditionalMetrics(double npv, double roi, double irr)
{
    Console.WriteLine("Additional metrics:");
    Console.WriteLine("Net Present Value (NPV): " + npv);
    Console.WriteLine("Return on Investment (ROI): " + roi);
    Console.WriteLine("Internal Rate of Return (IRR): " + irr);
}

static double ConvertToCurrency(double value)
{
    // Реалізуйте конвертацію валюти тут
    return value * 27.5; // Приклад
}

static void DisplayCurrencyConversion(double totalCost, double expectedBenefit, double npv,
double roi, double irr)
{

```

```

Console.WriteLine("Currency conversion:");
Console.WriteLine("Total Cost (Converted): " + totalCost);
Console.WriteLine("Expected Benefit (Converted): " + expectedBenefit);
Console.WriteLine("Net Present Value (Converted): " + npv);
Console.WriteLine("Return on Investment (Converted): " + roi);
Console.WriteLine("Internal Rate of Return (Converted): " + irr);
}

```

```

static void SaveDataToFile(double totalCost, double expectedBenefit, double npv, double roi,
double irr)

```

```

{
    string filePath = "project_data.txt";
    using (StreamWriter writer = new StreamWriter(filePath))
    {
        writer.WriteLine($"Total Cost: {totalCost}");
        writer.WriteLine($"Expected Benefit: {expectedBenefit}");
        writer.WriteLine($"Net Present Value: {npv}");
        writer.WriteLine($"Return on Investment: {roi}");
        writer.WriteLine($"Internal Rate of Return: {irr}");
    }
    Console.WriteLine("Data saved to file: " + filePath);
}

```

```

static void LoadDataFromFile()

```

```

{
    string filePath = "project_data.txt";
    if (File.Exists(filePath))
    {
        Console.WriteLine("Loading data from file...");
        using (StreamReader reader = new StreamReader(filePath))

```

```
{
    string line;
    while ((line = reader.ReadLine()) != null)
    {
        Console.WriteLine(line);
    }
}
else
{
    Console.WriteLine("No data file found.");
}
}
```