

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Аналітична підсистема підтримки прийняття рішень по вибору методології управління ІТ-проектом»»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проектування»

Виконавець роботи: студент групи ІТ.м-91 Псарьов Олександр Вікторович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«__» грудня 2020 р.

Науковий керівник

(підпис)

к.т.н., доц., Парфененко Ю. В.

Голова комісії

(підпис)

Шифрін Д. М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2020 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентів

Псарьов Олександр Вікторович
(прізвище, ім'я, по батькові)

1 Тема проекту Аналітична підсистема підтримки прийняття рішень по вибору методології управління ІТ-проектом

затверджена наказом по університету від « 26 » листопада 2020 р. № 1824-III

2 Термін здачі студентом закінченого проекту « 07 » _____ грудня _____ 2020 р.

3 Вхідні дані до проекту методології управління ІТ-проектами, фактори впливу на успішність ІТ-проекту, критерії вибору методологій.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їй належить розробити) аналіз предметної області, постановка задачі та методи дослідження, проектування аналітичної підсистеми підтримки прийняття рішень по вибору методології управління ІТ-проектом, розробка аналітичної підсистеми підтримки прийняття рішень по вибору методології управління ІТ-проектом.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність, мета дипломного проекту, постановка задачі проекту, дослідження аналогів, проектування аналітичної підсистеми, схема процесів розробки аналітичної підсистеми, побудова дерева рішень, реалізація, висновки.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз та опис (факторів) специфіки управління ІТ-проектами	01.10.20 – 02.10.20	
2	Ідентифікація ідеї проекту	05.10.20 – 05.10.20	
3	Дослідження методів та підходів для розробки	06.10.20 – 07.10.20	
4	Огляд інструментарію	12.10.20 – 13.10.20	
5	Визначення вимог	14.10.20 – 16.10.20	
6	Створення WBS	20.10.20 – 20.10.20	
7	Створення OBS	21.10.20 – 21.10.20	
8	Розробка аналітичної підсистеми	02.11.20 – 27.11.20	
9	Виправлення помилок	26.11.20 – 27.11.20	
10	Апробація та імплементація	02.12.20 – 04.12.20	
11	Здача документації	08.12.20 – 09.12.20	
12	Архівація	10.12.20 – 10.12.20	

Магістрант _____

Псарьов О.В.

Керівник роботи _____

к.т.н., доц. Парфененко Ю.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Аналітична підсистема підтримки прийняття рішень по вибору методології управління ІТ-проектом»».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 30 найменувань, додатків. Загальний обсяг роботи – 79 сторінок, у тому числі 59 сторінок основного тексту, 3 сторінки списку використаних джерел, 16 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці аналітичної підсистеми підтримки прийняття рішень по вибору методології управління ІТ-проектом. В роботі проведено огляд факторів впливу на успішність ІТ-проекту, ролі прийняття рішень в управлінні ІТ-проектами та визначення потреби у розробці системи, досліджено існуючі технології для розробки. У роботі було спроектовано аналітичну підсистему з точки зору розробника та процес використання продукту, побудовано діаграму варіантів використання, створено схему процесів використання аналітичної підсистеми підтримки прийняття рішень по вибору методології управління ІТ-проектом, побудовано дерево рішень на основі єдиних критеріїв вибору методологій управління проектами та реалізовано алгоритм по вибору методології. Результатом проведеної роботи є аналітична підсистема підтримки прийняття рішень по вибору методології управління ІТ-проектом. Практичне значення роботи полягає у розробці аналітичної підсистеми, що дає змогу спростити процес прийняття рішень по вибору оптимальної методології управління ІТ-проектом.

Ключові слова: підсистема, підтримка, прийняття рішення, аналітика, управління, ІТ- проект.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Огляд факторів впливу на успішність ІТ– проекту	9
1.2 Роль прийняття рішень в управлінні ІТ – проектами	12
1.3 Огляд методологій управління ІТ – проектами.....	19
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	27
2.1 Мета та задачі	27
2.2 Методи дослідження.....	28
2.3 Планування робіт	32
2.3.1 Ідентифікація мети ІТ-проекту	32
2.3.2 Планування змісту структури робіт ІТ-проекту	33
2.3.3 Побудова календарного графіку виконання ІТ-проекту.....	34
2.3.4 Планування ризиків ІТ-проекту.....	34
3 ПРОЕКТУВАННЯ АНАЛІТИЧНОЇ ПІДСИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПО ВИБОРУ МЕТОДОЛОГІЇ УПРАВЛІННЯ ІТ- ПРОЕКТОМ.....	37
3.1 Структурно-функціональне моделювання	37
3.2 Створення моделі варіантів використання.....	41
4 РОЗРОБКА АНАЛІТИЧНОЇ ПІДСИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПО ВИБОРУ МЕТОДОЛОГІЇ УПРАВЛІННЯ ІТ-ПРОЕКТОМ.....	42
4.1 Налаштування інструментів для розробки	42
4.2 Розробка веб-орієнтованого аналітичного рішення	45
4.3 Тестування системи	56
ВИСНОВКИ.....	59

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
Додаток А.....	63
Додаток Б.....	67

ВСТУП

У своїй повсякденній роботі, проектні менеджери весь час працюють в рамках прийняття управлінських рішень різного рівня складності. Багато в чому вибір, який ми робимо, і дії, яких ми дотримуємось в результаті важливих рішень, можуть значно вплинути на благополуччя, майбутні перспективи для нас самих, клієнтів та проектних команд. Іншими словами, такий вибір може сформувати саму якість кар'єри і навіть життя всіх, хто покладається на нас. Враховуючи потенційно-критичний характер цих рішень, можна очікувати, що, як правило, необхідно бути обережним при визначенні найкращого шляху : достатньо часу буде витрачено на дослідження та зважування найрізноманітніших альтернатив - і послідовна та перевірена стратегія прийняття рішень буде застосована для визначення найкращого з усіх доступних варіантів. Тільки таким чином можливо зменшити ризик та збільшити ймовірність успіху.

Проте, чи через брак часу, знань або інтересу, мало хто з нас вдається до такої строгості та належної підготовки, приймаючи ключові рішення. Натомість ми можемо бути вражені великою проблемою, яка стоїть перед нами, і піти шляхом найменшого опору. Можна зосередитись на одній категорії даних та ігнорувати іншу, або уникати сторонніх думок, обґрунтовуючи, що додавання нової інформації не обов'язково покращить якість остаточного результату. Можна розподілити всі наявні факти та ідеї, щоб дійти до висновку, який все ще залишає сумніви щодо того, чи достатньо врахований «людський фактор». Нарешті, необхідно просто робити те, що «відчувається як правильний варіант», сподіваючись, що інтуїція синтезує як об'єктивний факт, так і суб'єктивне почуття в кінцевий результат, який задовільнить як інтелект так і емоції.

Мета: Розробка аналітичної підсистеми для вибору методології управління IT-проектом з урахуванням аналізу даних різних показників та

критеріїв підбору, вивченні специфіки ведення проекту та аналізі можливої конвергенції підходів.

Задачі:

- Дослідження методологій управління ІТ-проектами;
- Аналіз факторів впливу на успішність ІТ-проекту;
- Аналіз процесів прийняття рішень по вибору методології управління

ІТ-проектом;

- Огляд та вибір методів розробки;
- Реалізація аналітичної підсистеми;

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд факторів впливу на успішність ІТ– проекту

Наявність талановитого менеджера проекту - це перший крок до реального успіху проекту, але є й інші важливі фактори, які значною мірою впливають на результату проекту. Щоб проект досяг успіху, потрібне ретельне планування, увага до деталей та ефективне спілкування. Завдяки пильному керівництву та потужному закриттю проекту компанія може постійно досягати успіху в проекті [10].

Кваліфіковані люди. Без відповідної команди будь-яка стратегія та план можуть повністю розвалитися. Через це основний персонал проекту, експертні ресурси, постачальники та всі зацікавлені сторони повинні бути частиною командної динаміки. Усі залучені особи повинні мати прихильність до групи, поділяти подібні бачення проектів і прагнути загального успіху [11].

Керівники проектів можуть зіткнутися з серйозними проблемами, якщо в команді присутня неадекватність. Невміле керівництво або несинхронізована команда можуть направити проект назустріч невдачі. Важливо призначити відповідних людей для кожного аспекту проекту та переконатися, що вони добре працюють разом. Крім того, вся команда повинна бути повністю проінформована та залучена для досягнення найбільш успішного результату, а це означає, що спілкування має бути на відповідному рівні [11].

Раціональне планування. Всебічне планування закладає успіх проекту з самого початку. Всі зацікавлені сторони повинні бути на разі під час процесу планування та завжди знати, в якому напрямку буде йти проект. Планування може допомогти команді дотримуватися термінів і залишатися організованими. Хороше планування не тільки дає змогу проектній команді зосереджуватися на поставлених задачах а й інформує зацікавлені сторони про хід проекту. якість, сервіс і швидкість [11].

Раціональне планування має багато переваг. Цей крок у процесі проекту дозволяє створити надійний та реалістичний часовий масштаб. Визначення точного часу для складання кошторисів та чітке документування етапів значно полегшить процеси в ході реалізації проекту. Професійний план детально описує всі потреби в ресурсах. Використовуйте інструмент, який пропонує повну шкалу перетягування, щоб ви могли швидко та ефективно скласти план проекту та встановити точну дату завершення. Використовуйте проекти, які раніше використовувалися як шаблони для своїх майбутніх. Таким чином ви не тільки зможете уникнути невизначеностей, пов'язаних зі створенням нового проекту, чим частіше ви повторно використовуєте шаблонний проект, тим ближче ви будете до точних термінів і кошторисів бюджету [13].

Комунікація. Уважне вивчення деталей та прослуховування зовнішніх джерел інформації є життєво важливим фактором для успіху проекту. Підтримка відкритого спілкування в команді є абсолютно важливим. Працюючи за певним графіком, важливо, щоб команда залишалася добре обізнаною. Якщо проблема виникає в одній частині проекту, це може негативно вплинути і на інші частини. Спілкування - найкращий спосіб запобігти виникненню проблем. Спілкування також повинно бути зосереджене всередині організації. Ведення організаційної історії великих проектів дасть зручний доступ до вдосконаленої політики та бізнес-процесів. Якщо цього не зробити, команда може повторити помилки, які вже сталися [9].

Управління ризиками. Керівники проектів знають, що все рідко йде точно так, як планувалося. Під час процесу планування життєво важливо скласти журнал ризиків із планом дій щодо можливих ризиків, з якими може зіткнутися проект. Переконайтеся, що всі ключові зацікавлені сторони знають ваш журнал ризиків та знають, де вони можуть його знайти. Якщо щось трапиться, команда може швидко вирішити проблему за допомогою плану управління, який уже встановлено. Це додасть команді впевненості у вирішенні проектних ризиків та дасть змогу клієнтам почуватися комфортно [2].

Чітко визначені цілі. Для забезпечення успіху проекту важливо ретельно визначити цілі, згідно потребами клієнта. Пам'ятайте, кроки, які ви робите для того, щоб ваш проект рухався вперед, або обов'язки, які ви встановлюєте перед кожним членом команди - все залежить від виду цілей, які ви маєте. Переконайтеся, що ви оцінюєте всі аспекти свого проекту та використовуєте ідеї для визначення чітких цілей [5].

Закриття проекту. Якщо проект не сильно закритий, він може продовжувати споживати ресурси. Команда проекту повинна бути сильною і погоджуватися із замовником, що всі критичні фактори успіху були дотримані. Підтвердження доставки проекту, тестування та випуску має бути узгоджене та підписане. Опитування щодо задоволеності - це хороші форми документації для реєстрації та подання для подальшого використання корисної інформації [3].

Робота менеджера проекту - забезпечити, щоб у проекті все працювало безперебійно, але навіть наявність чудового менеджера проекту не гарантує успішного результату. Вся команда, яка звертає увагу на ключові фактори - це те, що допоможе привести проект до справжнього успіху. Далі цей успіх призведе до ініціативних, організованих планів проектів та підвищенні якості всіх майбутніх проектів [3].

Інструменти для реалізації успішних проектів. Програмне забезпечення для управління проектами - це один із найкращих способів організувати всі вищезазначені етапи та забезпечити відкриту співпрацю, планування та виконання поставлених задач командою протягом усього проекту. Інструменти допомагають менеджерам проектів організувати кожен крок проекту та оптимізувати командні ресурси. Вони також інтегруються з іншим програмним забезпеченням, дозволяючи командам переглядати як загальні витрати, так і деталі проекту [4].

1.2 Роль прийняття рішень в управлінні ІТ – проектами

Прийняття рішень - це акт вибору серед наявних альтернатив. Існує велика кількість рішень, які приймаються людьми у повсякденному житті. В бізнесі рішення приймаються на кожному кроці [1]. Прийняття рішень також розглядається як одна з важливих функцій управління проектами. Управлінські функції, такі як планування, організація, робота з персоналом, керівництво, координація та контроль здійснюються через рішення. Прийняття рішень можливе за наявності двох або більше варіантів вирішення проблеми. Якщо є лише одна альтернатива, тоді мова не йде про прийняття рішень. Таким чином, можна сказати, що - це підхід до вирішення проблем шляхом вибору конкретного курсу дій серед можливих альтернатив [6].

Прийняття рішень вважається основою для управління бізнесом, оскільки без прийняття правильного рішення в потрібний час, нічого не можна зробити. Важливість прийняття рішень можна описати такими пунктами [21]:

1. **Досягнення цілі.** Прийняття рішень грає велику роль у досягненні організаційних цілей / задач протягом заданого часу та бюджету. Це допомагає знайти найкращу альтернативу, як правильно використовувати ресурси та задовільняє працівників на робочому місці. Як результат, цілі або завдання організації можуть бути досягнуті відповідно до бажаного результату [9].
2. **Мотивація команди.** Прийняття рішень забезпечує загальну основу діяльності та управлінські принципи для оперативного рівня персоналу, надає різні типи зручностей та переваг вчасно. Як результат, працівники мотивовані до своєї роботи / відповідно до організаційних вимог.
3. **Правильне використання ресурсів.** Організація має різні ресурси, такі як людина, гроші, метод, матеріал та інформація. Всі ці ресурси використовуються належним чином без будь-яких витоків та

марнотратства за допомогою прийняття правильного рішення в потрібний час. Як результат, організація може працювати з мінімальними витратами.

4. **Вибір найкращої альтернативи.** Як ми знаємо, проблема має кілька варіантів вирішення. Прийняття рішень дає змогу обрати найкращу альтернативу серед різних, аналізуючи їх по черзі, використовуючи різні фінансові, статистичні та бухгалтерські інструменти / методи.
5. **Оцінка управлінської діяльності.** Якість / успіх менеджера багато в чому залежить від кількості правильних рішень, які він / вона може прийняти для організаційного успіху. Тому прийняття рішень дуже важливе для оцінки результатів діяльності вищого рівня управління.
6. **Незамінний компонент / елемент.** Прийняття рішень є необхідним елементом / компонентом для організаційного успіху, оскільки без прийняття правильного рішення в потрібний час в подальшому нічого не можливо буде виконати відповідно до плану.
7. **Важлива функція для менеджерів.** Рішення повинні прийматись у всіх управлінських функціях, таких як планування, організація, мотивація, управління та контроль, а також у всіх функціональних сферах, таких як виробництво, маркетинг, фінанси.

На рис. 1.1 зображені етапи процесу прийняття рішень:



Рисунок 1.1 – Етапи процесу прийняття рішень

Після вибору найкращої альтернативи, менеджер повинен перетворити рішення на дію. Для цього він повинен спілкуватися зі своїми підлеглими та

управляти різними додатковими ресурсами для реалізації організаційного рішення [20].

Особа, яка приймає рішення, повинна розробити різні можливі альтернативи для вибору найбільш відповідного. При розробці альтернативного курсу дій він може використовувати власні знання, навички, досвід та технічну підтримку зі сторони експертів [23].

Початковий етап процесу прийняття рішення полягає у визначенні точної проблеми. Причина проблем може бути як внутрішня так і зовнішня. Особи, які приймають рішення, повинні правильно визначити проблеми, перш ніж приймати рішення. Тому він / вона може використовувати власні знання, навички, досвід та збирати інформацію із внутрішніх та зовнішніх джерел [23].

Після ідентифікації проблеми особа, яка приймає рішення, повинна систематично та науково проаналізувати проблему з точки зору витрат, часу, законності, організаційних ресурсів та впливу проблеми. Аналізуючи проблему, є можливість використовувати різні фінансові, бухгалтерські та статистичні інструменти чи методи [26].

Проаналізувавши різні альтернативи, особа, що приймає рішення, повинна вибрати найкращу альтернативу. Для цього особа може використовувати свої знання, вміння та досвід а також поспілкуватись з іншими зацікавленими сторонами [25].

Останнім етапом є отримання відповідей або відгуків від інших зацікавлених сторін організації. Якщо відповідь позитивна, тоді процес прийняття рішення успішно завершено. Якщо відповідь негативна, тоді необхідно перейти до першого кроку, для вибору нового організаційного рішення [24].

Після розробки різних варіантів, особа, що приймає рішення, повинна оцінити всі альтернативи по черзі. На цьому кроці він / вона повинен спробувати знайти відповіді на наступні запитання:

- Чи можлива альтернатива з точки зору витрат, часу, законності та інших організаційних ресурсів?

- Чи задовільна альтернатива для вирішення організаційних проблем?
- Чи відповідають функції альтернатив цілям бізнесу?

Нижче наведені моделі прийняття рішень, які найбільш часто використовуються в проектному менеджменті:

Дерево рішень.

Існує велика кількість моделей прийняття рішень, які можна ефективно використовувати як у професійному, так і в особистому житті. Ці інструменти високо цінуються у сфері управління проектами, оскільки мають величезну здатність підтримувати рішення, прийняті керівником проекту. Хороший менеджер проектів може зрозуміти потребу цих ефективних інструментів, які можна використовувати протягом усього життєвого циклу управління проектами. Як менеджер проекту, кожне ваше рішення може викликати як нові загрози, так і нові можливості. Дана методика дозволяє краще підготуватися до кожної події та зробити найбільш обґрунтований вибір для кожного етапу проекту [24]. Слід виділити такі переваги:

- **Ясність:** Дерева рішень надзвичайно легко зрозуміти. При правильній структурі кожен вибір та потенційний результат логічно перетікає один до іншого.
- **Ефективність:** Оскільки дерева рішень подають інформацію таким прямим способом, їх можна швидко проаналізувати та використовувати для прийняття важливих рішень.
- **Адаптивність:** Дерева рішень можна легко адаптувати, щоб врахувати нові ідеї або можливості. Це означає, що дерево може рости поряд із вашими проектами.
- **Сумісність:** Така техніка аналізу може використовуватись в тандемі з іншими методологіями управління проектами, що дає повну гнучкість під час управління проектами.

SWOT-аналіз.

Це дуже поширена модель прийняття рішень, яка спеціально використовується під час техніко-економічного обґрунтування проекту. Даний метод підводить проблему до висновку, опираючись на аналіз сильних та слабких сторін, а також різних загроз зі сторони навколишнього середовища [27]. На рис. 1.2 зображена загальна схема SWOT-аналізу:

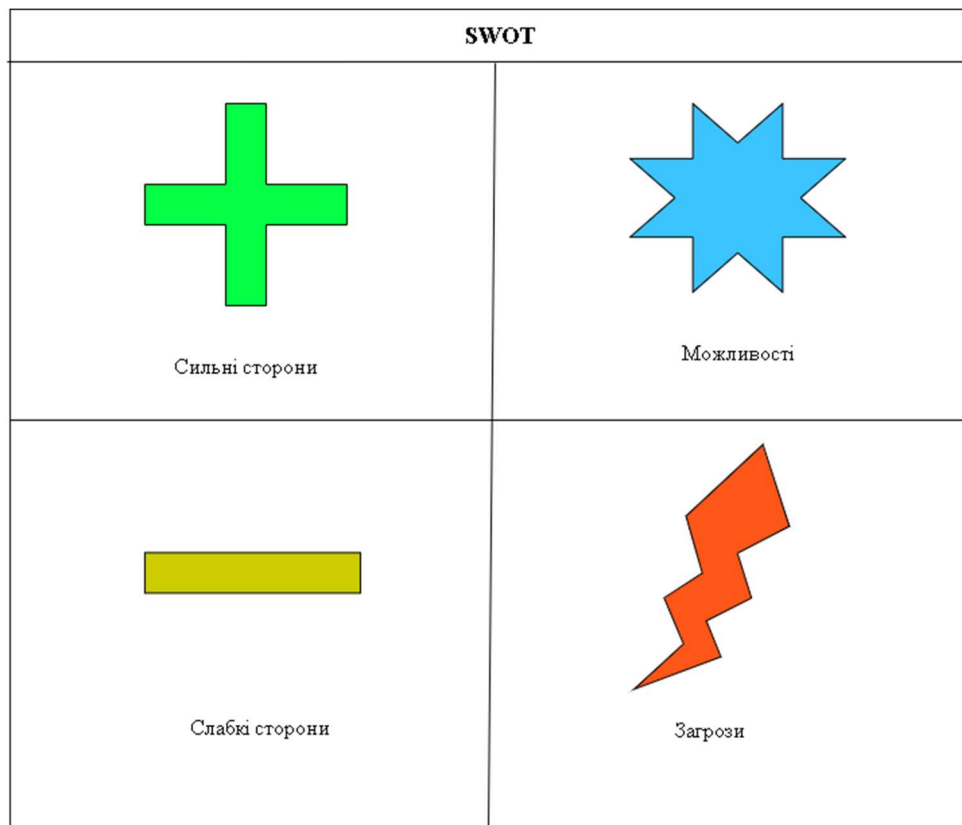


Рисунок 1.2 – Схема SWOT-аналізу

Принцип Парето.

Це також називається правилом 80-20, коли ви розставляєте пріоритети своїм проблемам, а потім знаходите рішення. Для розуміння концепції це правило 80-20 можна описати як приклад проблем в організації, створених людьми. Можна сказати, що 80% проблем створюють 20% людей у будь-якій організації [27].

Симуляція Монте-Карло.

Симуляція Монте-Карло - це модель, яка, по суті, фокусується на численних моделюваннях за випадковою вибіркою, що дає приблизні

результати. Це просто говорить нам, що навіть моделі можуть бути випадковими і не дуже відповідати реальності. Ця імітаційна модель є цікавою завдяки випадковій вибірці та використанню ймовірності та статистики для визначення результату. В управлінні проектами метод імітаційного моделювання використовується для кількісного аналізу ризику, де ви зможете визначити кількісний вплив ризику на мету проекту [9].

Система підтримки прийняття рішень. Система підтримки прийняття рішень - це інформаційна система, яка допомагає бізнесу у прийнятті рішень, які вимагають рішучості та послідовності дій. Інформаційна система допомагає керівництву організації на середньому та високому рівні шляхом аналізу величезних обсягів неструктурованих даних та накопичення інформації, яка може допомогти у вирішенні проблеми та у прийнятті рішень [7].

Система підтримки прийняття рішень розроблює детальні інформаційні звіти шляхом збору та аналізу даних. Отже, вона відрізняється від звичайної операційної програми, метою якої є збір даних, а не їх аналіз [7].

В організації використовується відділами планування - наприклад, операційним відділом, який збирає дані та створює звіт, який може використовуватися менеджерами для прийняття рішень. В основному використовується при прогнозуванні продажів, для інвентаризації та даних, пов'язаних з операціями, а також для надання інформації клієнтам у зрозумілій для них формі [7].

На рис. 1.3 зображені компоненти системи підтримки прийняття рішень:

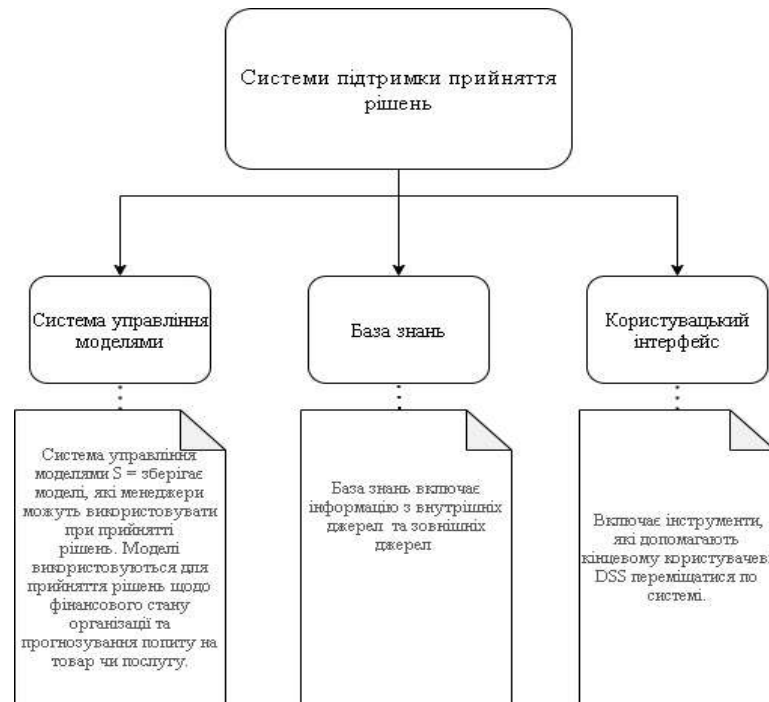


Рисунок 1.3 – Компоненти системи

Переваги:

- Збільшує швидкість та ефективність діяльності з прийняття рішень.
- Автоматизація управлінських процесів, це означає, що більша частина часу менеджера може бути витрачена на прийняття рішень.
- Покращення міжособистісного спілкування всередині організації.

Недоліки:

- Висока вартість розробки та впровадження.
- Перевантаження інформації, інформаційна система схильна враховувати всі аспекти проблеми. Це створює дилему для кінцевих користувачів.
- Впровадження може викликати страх і негативну реакцію у працівників нижчого рівня. Це пов'язано з тим, що багатьом з них не комфортно користуватися новими технологіями і вони бояться втратити роботу через техніку.

1.3 Огляд методологій управління ІТ – проектами

У сучасному світі управління проектами, перспективні менеджери та лідери не дотримуються єдиної методології - вони набувають знань у багатьох з них, також вони вчаться поєднувати різні практики, щоб задовольнити те, що вимагає проект. Розглядаючи типи методології проектів, є можливість побачити відмінності в механізмах, які використовують різні методології; як вони дають визначення способу роботи [13].

Нижче наведено список методологій управління проектами, що описує найпопулярніші підходи до проектного менеджменту.

Waterfall – традиційний підхід до управління проектами. У ньому завдання та етапи виконуються лінійно, послідовно, і кожен етап проекту повинен завершитися перш ніж перейти до наступного [12].

Прогрес тече в одному напрямку, як справжній водоспад. Однак, як справжній водоспад, це може швидко стати небезпечним. Оскільки все заплановано з самого початку, є багато місця для помилок, якщо очікування не збігаються з реальністю. І вже не буде можливості повернутися до попереднього етапу, коли він буде завершений [12].

На рис. 1.4 зображені загальна схема каскадної моделі:

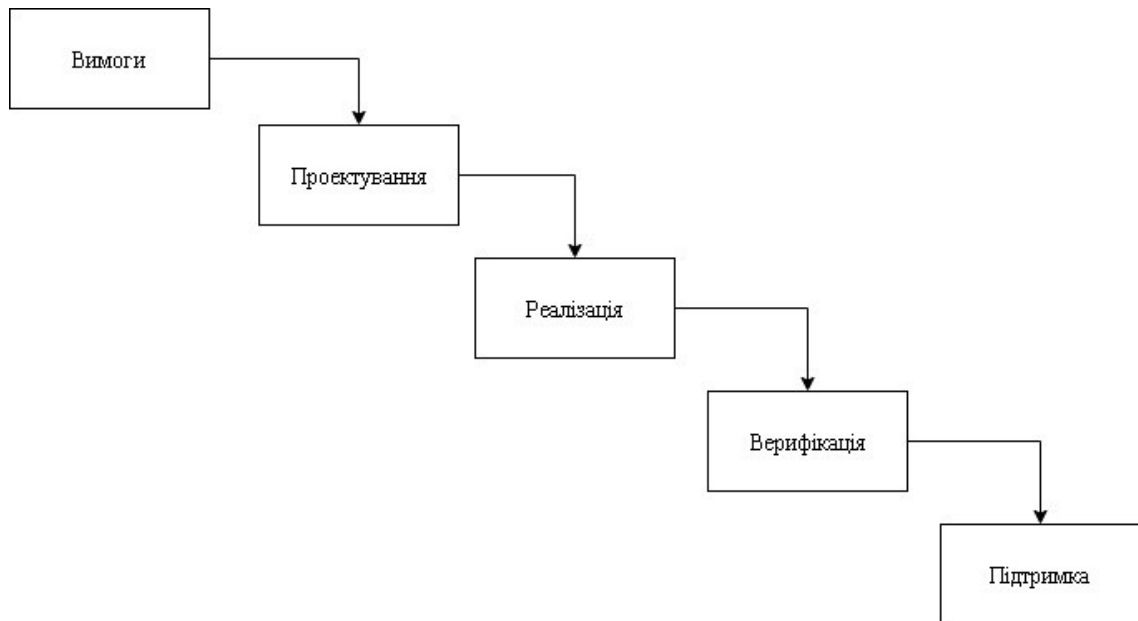


Рисунок 1.4 – Загальна схема каскадної моделі

Дану методологію доцільно використовувати коли:

- Кінцева мета вашого проекту чітко визначена - і не буде змінюватися.
- Зацікавлені сторони точно знають, чого хочуть.
- Ваш проект є послідовним та передбачуваним.
- Ви працюєте в регульованій галузі, яка потребує великого відстеження проектів або документації.

Ця методологія управління проектами може бути не для вас, якщо:

- Ваш проект може змінитися.
- Перед початком роботи у вас немає повного уявлення про всі вимоги.
- Вам потрібно постійно проводити тестування або пристосовуватися до зворотного зв'язку під час процесу.

Agile методологія. Концепція гнучкого управління проектами продовжила створювати кілька конкретних підфреймів та методологій, таких як scrum, kanban та lean [12].

Ключовими принципами гнучких методологій управління проектами є:

- Спільна робота.
- Швидкість.
- Відкритість до буд-яких змін.

Гнучкі методології управління проектами, як правило, передбачають короткі етапи роботи з частим тестуваннями та адаптацією протягом усього періоду.

Дану методологію доцільно використовувати, коли ваш проект відкритий до змін, є бачення кінцевого результату, стейкхолдери хочуть приймати участь на кожному етапі проекту [22].

Ця методологія управління проектами не доцільно використовувати коли:

- Потрібна велика кількість документації.
- Потрібен передбачуваний результат.
- Проект не може дозволити вносити корективи.
- Відсутність мотивованих людей.
- Наявні чіткі терміни або кінцеві результати, які вам потрібні, щоб не відставати.

Scrum – форма гнучкого управління проектами. Можна сприймати це скоріше як структуру, ніж як методологію управління проектами.

З Scrum робота ділиться на короткі цикли, відомі як "спринти", які зазвичай тривають близько 1-2 тижнів. Невеликі команди очолює Скрам-Майстер протягом усього спринту, після чого команда переглядає та аналізує свої результати у «ретроспективі спринту» та вносять необхідні зміни перед початком наступного спринту [17].

Спробуйте цю методологію управління проектами, якщо:

- Ви прагнете постійного вдосконалення.

Kanban – це популярний фреймворк, який використовується для реалізації гнучкої розробки програмного забезпечення. Базується на обміні інформацією в режимі реального часу та повної прозорості роботи. Робочі предмети представлені візуально на дошці канбану, що дозволяє членам команди моніторити стан кожної роботи в будь-який час[15].

Робота всіх команд обертається навколо дошки kanban, інструменту, який використовується для візуалізації роботи та оптимізації потоку робіт всередині команди. Хоча фізичні дошки популярні серед деяких команд, віртуальні дошки

є найважливішою функцією будь-якого гнучкого інструменту розробки програмного забезпечення для їх моніторингу, спрощення співпраці та доступності з різних місць [15].

Базова дошка канбан складається з трьох робочих процесів: «Виконати», «Виконується» та «Готово». Однак, залежно від розміру, структури та цілей команди, робочий процес може бути складений відповідно до унікального процесу будь-якої конкретної команди [15].

Методологія kanban спирається на повну прозорість роботи та передачу інформації про потенціал у режимі реального часу, тому дошка повинна розглядатися як єдине джерело інформації, для успішної роботи команди [16].

Можна виділити такі переваги:

- Гнучкість планування.
- Візуальні метрики. Однією з основних цінностей є сильна увага до постійного підвищення ефективності та результативності команди на кожній ітерації роботи. Діаграми забезпечують візуальний механізм для команд, які забезпечують їх подальше вдосконалення. Коли команда може бачити дані, легше виявити вузькі місця в процесі (і видалити їх). Два звіти, які використовують команди kanban, є контрольні діаграми та сукупні діаграми потоків.
- Скорочені цикли часу. Час циклу є ключовим показником для команд kanban. Час циклу – це кількість часу, за який одиниця роботи проходить робочий процес команди - від моменту початку роботи до моменту її відправлення. Оптимізуючи час циклу, команда може впевнено прогнозувати майбутню роботу.

Канбан і Скрам поділяють одні й ті самі поняття, але мають дуже різні підходи.

Порівняльна характеристика Scrum та Kanban (табл.1.1).

Таблиця 1.1 – Порівняльна характеристика методологій.

	Scrum	Kanban
Каденція	Спринти регулярної фіксованої довжини (тобто 2 тижні)	Безперервний потік
Методика релізу	В кінці кожного спринту, якщо це схвалено власником продукту	Безперервна доставка або на розсуд команди
Ролі	Власник продукту, скрам-майстер, команда розробників	Немає існуючих ролей
Основні показники	Швидкість	Час циклу
Філософія змін	Команди повинні прагнути не змінювати прогноз спринту під час спринту	Зміни можуть відбутися в будь-який час

Prince2. Основні особливості PRINCE2 зосереджені на виправданні бізнесу, визначенні структури організації для команди управління проектами та використанні підходу, заснованого на продуктах. Акцент робиться на розподілі проекту на керовані та контрольовані етапи з гнучкістю. PRINCE2 забезпечує великий контроль над ресурсами проекту та дає змогу більш ефективно управляти бізнес- ризиками та ризиками проекту [8].

Методологія використовується для відповіді та структурування базових питань з управління проектами, таких як:

- Що ти намагаєшся робити?
- Коли ти почнеш?
- Що потрібно для виконання?
- Тобі необхідна допомога?
- Скільки часу необхідно?
- Яка вартість?

Організовуючи проект у логічні етапи, PRINCE2 вимагає структури, яка має організований і контрольований план. Контроль досягається низкою процесів. Ці процеси охоплюють усі дії, що складають проект, від початку до кінця [8].

Ролі в методології чітко визначені. Керівник проекту – це той, хто відповідає за організацію та контроль за проектом та його процесами. Він обирає команду, яка буде виконувати проект, і контролюватиме їх роботу, щоб переконатися, що вона виконується вчасно та належним чином, згідно з планом проекту[14].

У проекті також бере участь замовник, користувач або постачальник. Це особа чи організація, яка платить за проект. Вони використовуватимуть продукт проекту або будуть якимось впливати на його результат. Іноді замовник є користувачем, тоді як постачальник - це сторона, яка пропонує досвід для успішного завершення проекту. Вони, як і команда проекту, потребують організації, щоб переконатися, що проект відповідає вимогам [18].

Нарешті, існує рада проекту, яка складається із замовника та керівників, особа, яка представляє сторону користувача, та особа, яка представляє постачальників або спеціалістів, залучених у проект. Керівник проекту буде регулярно звітувати перед радою проекту, повідомляючи їм про прогрес та будь-які проблеми. Рада проекту, в свою чергу, приймає рішення про те, як буде розвиватися проект, або вирішуватиме проблемні питання [19].

Процес PRINCE2 складається з семи етапів:

- Запуск проекту
- Ініціалізація проекту
- Керування
- Етап контролю
- Управління доставкою продукту
- Управління межами етапу
- Закриття

Спіральна методологія. Спіральна модель – це метод життєвого циклу розробки систем, що використовується для управління ризиками, що поєднує в собі ітеративну модель процесу розробки з елементами моделі водоспаду . Спіральна модель використовується інженерами-програмістами та часто застосовується у великих, дорогих та складних проектах [12].

Спіральна модель дозволяє поступово випускати та вдосконалювати продукт через кожен етап спіралі, а також дає можливість створювати прототипи на кожній фазі. Найважливішою особливістю моделі є її здатність управляти невідомими ризиками після початку проекту; створення прототипу робить це можливим [12].

Фази спіральної моделі. Дивлячись на схему спіральної моделі, радіус спіралі відображає вартість проекту, а кутова ступінь - прогрес, досягнутий у поточній фазі. Кожен етап починається з мети проектування і закінчується, коли розробник або клієнт перевіряє прогрес [12].

Кожну фазу можна розділити на чотири квадранти: виявлення та розуміння вимог, проведення аналізу ризиків, побудова прототипу та оцінка продуктивності програмного забезпечення [12].

Етапи починаються у квадранті, присвяченому виявленню та розумінню вимог. Повинна бути визначена загальна мета етапу, а всі цілі повинні бути розроблені та проаналізовані.

Далі слід провести аналіз ризиків для всіх можливих рішень, щоб знайти будь-які несправності або уразливі місця - наприклад, перевитрати бюджету або областей програмного забезпечення, які можуть бути відкритими для кібератак. Потім кожен ризик слід вирішити, використовуючи найбільш ефективну стратегію.

У наступному квадранті прототип вже розроблений, пройшов тестування. Цей етап включає: архітектурне проектування, проектування модулів, фізичне проектування продукту та остаточний дизайн. Він бере пропозицію, створену в перших двох квадрантах, і перетворює її на програмне забезпечення, яке можна використовувати [20].

Нарешті, у четвертому квадранті оцінюються результати тестування останньої версії.

Прогресивна природа моделі дозволяє розробникам розбивати великий проект на менші частини та вирішувати по одній функції за раз, гарантуючи, що нічого не пропущено. Крім того, оскільки побудова прототипу виконується поступово, оцінка вартості всього проекту іноді може бути простішою.

Переваги спіральної моделі:

- Гнучкість - Зміни, внесені до вимог після початку розробки, можуть бути легко прийняті та включені.
- Управління ризиками - спіральна модель передбачає аналіз та обробку ризиків на кожному етапі, покращуючи безпеку та шанси уникнути атак. Повторний процес розробки також полегшує управління ризиками.
- Задоволеність клієнтів - спіральна модель полегшує зворотній зв'язок із клієнтами.

Обмеження спіральної моделі включають:

- Висока вартість.
- Складність.
- Складність управління часовими рамками.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Досягнення мети роботи полягає у розробленні аналітичної підсистеми для вибору методології управління ІТ-проектом з урахуванням аналізу даних різних показників та критеріїв підбору, вивченні специфіки ведення проекту та аналізі можливої конвергенції підходів.

В роботі розв'язувались такі питання:

- дослідження факторів впливу на успішність ІТ-проекту;
- дослідження методологій управління ІТ-проектів;
- дослідження методів реалізації ІТ-проектів;
- дослідження моделей управління проектами;
- дослідження процесів підтримки прийняття рішень;

Об'єкт дослідження: процеси вибору методології управління проектами.

Предмет дослідження: процеси підтримки прийняття рішень по вибору методології управління ІТ-проектом.

Продуктом даного проекту є аналітична підсистема для проектних менеджерів, який дає змогу підібрати найбільш відповідну методологію, зменшити ризики для успішного ведення ІТ-проекту.

Розроблена аналітична підсистема дозволить вдосконалити виробничий процес, зменшити час на прийняття рішень при виборі методології управління ІТ-проектом, більш раціонально розподілити ресурси, скоротити час прийняття рішень, підвищити якість та швидкість розробки в ході проекту того чи іншого типу.

2.2 Методи дослідження

Перед початком розробки, був проведений аналіз та обрано методи і технології, які будуть використовуватися у роботі.

Вибір найкращої методології для використання у вашому бізнесі не є простим завданням.

Методологія управління проектами – це набір прийомів та практик, які використовуються організацією та проектними командами для ефективного управління проектом та збільшення шансів на досягнення запланованих цілей. Але існує велика кількість методологій на вибір, тому потрібна ретельна стратегія, для того щоб обрати найкращу. Перш ніж зупинитися на певній методології, слід врахувати дві важливі речі [12].:

- Декілька внутрішніх та зовнішніх факторів можуть вплинути на обрану вами методологію.
- Процес оцінки кожної методології повинен бути передбачений, щоб переконатися, що вона найкраще відповідає потребам.

Багато факторів можуть вплинути на ваш успіх та успіх проекту. Перш ніж приймати рішення, переконайтеся, що ви відповіли на такі запитання:

- Як побудована ваша компанія?
- Які цілі вашої компанії?
- Які сильні, слабкі сторони, можливості, можливі ризики у вашому бізнесі?
- Хто ваші потенційні клієнти?
- Як кожна методологія ризикує успіхом проектів?
- Наскільки складний проект?
- Який розмір і вартість проекту?
- Чи підтримує внутрішня інфраструктура, технології та процеси ту чи іншу методологію?

При оцінці методології реалізації проекту слід враховувати різні фактори. Але основними є:

- Стратегічні цілі клієнта;
- Ділові цінності;
- Зацікавлені сторони;
- Обсяг і складність проекту;
- Бізнес-ризиками;

Оцінка методології проекту повинна обов'язково включати організаційні можливості та готовність до впровадження певної моделі реалізації. Оскільки кожна методологія призначена для вирішення конкретних цілей, таких як якість, ефективність та зменшення витрат, дуже важливо підібрати методологію яка б відповідала цілям вашого проекту. Наприклад, вибір методології Lean - не найкращим, якщо якість є головним пріоритетом; Agile був би кращим варіантом, оскільки він призначений для досягнення вищого рівня якості [12].

Внутрішні критерії оцінки методології (табл.2.1).

Таблиця 2.1 – Критерії оцінки методологій.

Організація	Рієнь зрілості, рівень гнучкості, доступні ресурси, продуктивність, ризики
Проект	Обсяг та складність, вимоги, терміни, оцікування стейкхолдерів, характер проекту
Люди	Розташування команди, досвід та навички, навчання
Процес	Внутрішні та зовнішні
Технологія	Доступне програмне забезпечення та інструменти

Крім того, слід враховувати характер методології. Деякі з них призначені для динамічної та швидкої доставки, а інші вимагають більшої стабільності та суворої структури [30].

На рис. 2.1 зображено приклад розподілу методологій за деякими факторами:

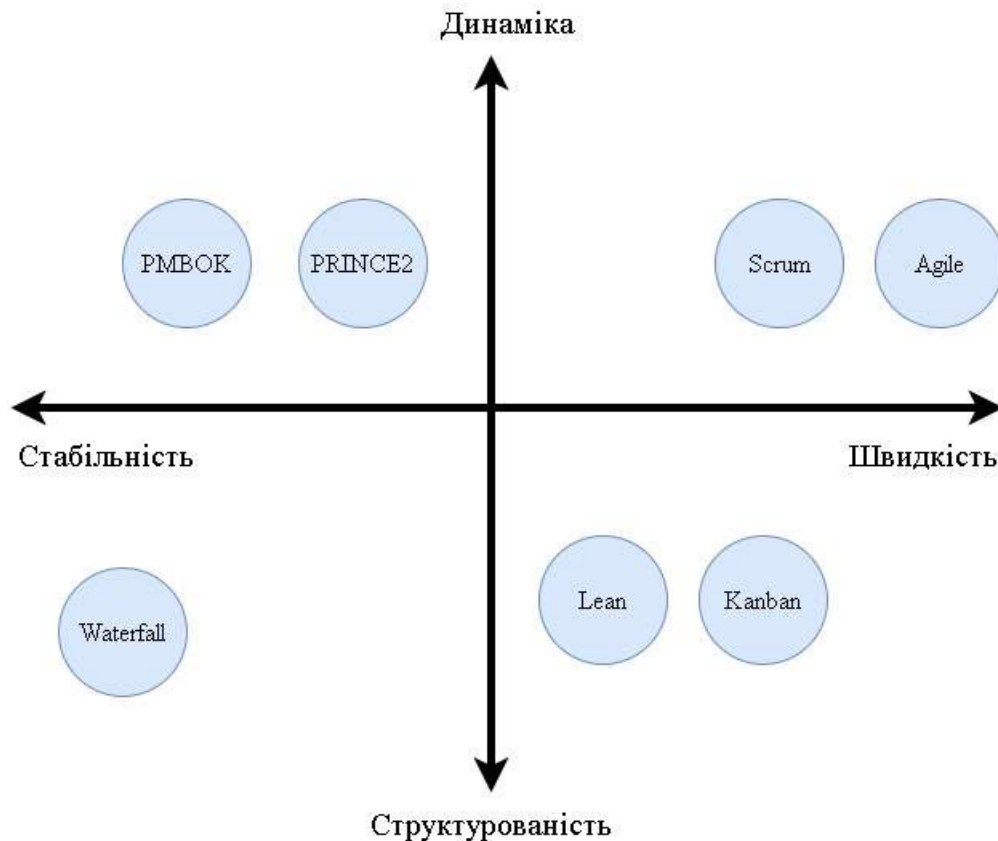


Рисунок 2.1 – Розподіл методологій за факторами Динаміка, Швидкість, Якість, Структурованість

Звичайно, існує ряд інших широко розповсюджених методологій управління проектами, які забезпечують ефективну реалізацію проектів. І, на жаль, універсальної, яка підходить для кожного проекту, не існує. Тож ви повинні адаптувати принципи, процеси та стандарти, які допоможуть забезпечити безперервну реалізацію проектів [14].

Вибір відповідної методології повинен ґрунтуватися на вашій команді, особливостях програмного забезпечення та організаційних можливостях. Після попереднього відбору декількох, наступним кроком є подальший аналіз, для ідентифікації найкращого збігу [14].

Фактори провалу проектів. Щороку організації у всьому світі стикаються з астрономічними показниками відмов, часто витрачаючи мільйони доларів через невдалий проект. Ті самі підприємства страждають від причин провалу проектів, залучають дорогих консультантів для оцінки та відновлення

провальних проєктів і часто відмовляються від того, що спочатку здавалося добре спланованими та добре організованими проєктами [19].

Не існує єдиного методу чи організаційної структури, яку можна використовувати для управління проєктами, щоб досягти успіху. Різні організації по-різному обробляють функціональні проєкти. Деякі з них мають фрагментовані та децентралізовані групи з декількома назвами, що вказують на те, що вони є проєктами, тоді як інші можуть мати велику кількість спеціалістів з управління проєктами в централізованій організації підтримки [29].

Незалежно від фізичного розташування функцій, існує загальний набір пов'язаних організаційних потреб, коли вони належним чином делегуються відповідним групам, які можуть бути використані для збереження або управління проєктами. Окрім чітко визначеного статуту, наступний список узагальнює деякі основні функції управління проєктами, необхідні для досягнення успіху в проєктах [18]:

- Набір та підтримка адекватних технічних та нетехнічних навичок роботи з ресурсами.
- Управління розподілом дефіцитних ресурсів.
- Ефективна комунікація.
- Правильний вибір та використання інструментів і технологій.

Існує безліч причин, які призводять до провалу ІТ-проєкту, але слід виділити з них 9 основних:

1. Невідповідність між проєктом та стратегічними пріоритетами організації.
2. Відсутність попередньо узгоджених мір щодо успіху проєкту.
3. Неєфективна взаємодія з зацікавленими сторонами.
4. Погані технічні навички в області управління проєктами.
5. Нестандартний підхід до управління проєктами та управління ризиками.
6. Відсутність можливості диференціювання етапів розробки та реалізації проєкту.
7. Оцінка була зосереджена на ціні, а не на довгостроковому співвідношенні ціни та якості, та досягненні цілей бізнесу.

8. Відсутність комунікації з вищими рівнями управління в організації.
9. Погана інтеграція проектної команди між клієнтами.

З дев'яти вищезазначених причин несправності лише одна є результатом недостатнього технічного знання. Решта стосується більш фундаментальних питань організації. Деякі з них є культурними або структурними, але всі служать як показники низької зрілості управління проектами на організаційному рівні. У багатьох випадках організації, які зазнають невдачі, мають висококомпетентних менеджерів проєктів, здібності яких значно перевершують можливості організацій, в яких вони працюють, з точки зору зрілості управління проектами [23].

Даний сценарій неминуче завдає шкоди організації і це означає, що вони не можуть реалізувати цілісні вигоди, які могли б отримати від управління проектами. Наприклад, якщо управління витратами або процеси збору даних неможливо стандартизувати, то як можна точно виміряти ефективність проєкту? Надалі такі ситуації зазвичай означають, що рівень відмов у проєктах зростає зі збільшенням їх складності та це робить неможливим виміряти ефективність управління проектами у зростаючій організації [28].

2.3 Планування робіт

2.3.1 Ідентифікація мети ІТ-проєкту

Мета проєкту полягає у розробленні аналітичної підсистеми для вибору методології управління ІТ-проєктом з урахуванням аналізу даних різних показників та критеріїв підбору, вивченні специфіки ведення проєкту та аналізі можливої конвергенції підходів.

2.3.2 Планування змісту структури робіт IT-проекту

Структура розподілу робіт (WBS) - це інструмент, який інтегрує базові показники обсягу, вартості та графіку, забезпечуючи узгодженість планів проектів, є одним з найважливіших документів управління проектами. На верхньому рівні ієрархії фіксується продукт проекту – аналітична підсистема для проектних менеджерів, який дає змогу підібрати найбільш відповідну методологію, зменшити ризики для успішного ведення IT-проекту. Наступний рівень відповідає за основні етапи проекту. Наступним кроком є розбиття цих етапів до елементарних робіт, що дасть змогу обчислити витрати та терміни виконання.

Продуктом у даному проекті є аналітична підсистема підтримки прийняття рішень по вибору методології управління IT-проектом. Далі виділяємо основні етапи виконання проекту: Ініціювання, Планування, Реалізація та Завершення. Ініціювання розбиваємо на аналіз та опис (факторів) специфіки управління IT-проектами, ідентифікація ідеї проекту. Найголовнішим етапом у проекті є розробка аналітичної підсистеми. Даний етап розбиваємо на такі пункти: налаштування php скрипту, створення форми критеріїв, реалізація алгоритму вибору методології та формування дизайну. Формування дизайну в свою чергу декомпозується на такі пункти як формування UI/UX дизайну та налаштування сітки Bootstrap.

Остаточний результат діаграми WBS зображено у Додатку А(рис.А1).

Після розробки календарного графіку, було побудовано OBS структуру.

Організаційна структура управління – це впорядкована сукупність взаємопов'язаних елементів, які забезпечують їх функціонування і розвиток як єдиного цілого.

Управляти організаційною структурою - означає оптимально розподілити цілі і завдання між працівниками організації.

У проекті розроблення аналітичної підсистеми були задіяні наступні виконавці:

- викладач Гайдабрус Б.В.– керівник проекту.
- студент Псарьов О.В. – розробник

Діаграма OBS для даного проекту представлена у Додатку А(рис.А2).

2.3.3 Побудова календарного графіку виконання ІТ-проекту

Для повного розуміння термінів виконання, залежностей та наявних ресурсів, було побудовано календарний графік проекту у вигляді діаграми Ганта, яку наведено у Додатку А(рис.А3).

Діаграма Ганта - це графічне зображення графіку проекту. Це тип гістограми, який показує дати початку та закінчення декількох елементів проекту, які включають ресурси, етапи, завдання та залежності.

2.3.4 Планування ризиків ІТ-проекту

Після визначення робіт проекту та побудови календарного графіку, необхідно ідентифікувати можливі ризики. Аналіз ризиків передбачає вивчення того, як можуть змінитися результати та цілі проекту внаслідок впливу ризикової події. Після виявлення ризиків їх необхідно проаналізувати, щоб визначити якісний та кількісний вплив ризику на проект, для того, щоб можна було застосувати відповідні заходи для їх усунення. Оцінка ризиків описана в табл. 2.2.

Таблиця 2.2 – Шкала оцінювання ймовірності виникнення та впливу ризику

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Таблиця 2.3 – Матриця ймовірності виникнення та впливу ризиків

Ймовірність виникнення	3	RS_1, RS_4, RS_5, RS_6, RS_9	RS_3, RS_7	
	2	RS_2	RS_8, RS_10	
	1			
		1	2	3
	Вплив ризику			

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики

Таблиця 2.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	1R2	RS_2
2	Виправдані	3R5	RS_1, RS_4, RS_5, RS_6, RS_8, RS_9, RS_10
3	Недопустимі	6R9	RS_3, RS_7

Шкала оцінювання визначених ризиків представлена в табл. 2.5

Таблиця 2.5 – Шкала оцінювання визначених ризиків

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	1R2	RS_2
2	Виправдані	3R5	RS_1, RS_4, RS_5, RS_6, RS_8, RS_9, RS_10
3	Недопустимі	6R9	RS_3, RS_7

Опис ризиків, визначення рангу, впливу ризику та стратегії реагування наведено в табл. 2.6.

Таблиця 2.6 – Оцінка ймовірності виникнення ризику, величини витрат, індекс ризику та стратегія реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	Тип стратегії реагування
RS_1	Відкритий	Відсутність взаємопорозуміння між розробником та стейкхолдером	Низька	Середній	4	Попередження
RS_2	Відкритий	Можливість появи альтернативи	Низька	Середній	2	Прийняття
RS_3	Відкритий	Завдання на розробку поставлено не чітко	Низька	Високий	6	Попередження
RS_4	Відкритий	Недостатні навички розробника	Низька	Середній	3	Попередження
RS_5	Відкритий	Некоректний розподіл часу	Низька	Середній	3	Попередження
RS_6	Відкритий	Постійні зміни ТЗ	Низька	Середній	3	Попередження
RS_7	Відкритий	Помилки розробки	Середня	Високий	9	Попередження
RS_8	Відкритий	Нечітка робота ПЗ	Низька	Високий	5	Попередження
RS_9	Відкритий	Відсутність бекапу	Низька	Середній	3	Попередження
RS_10	Відкритий	Зміни вимог в ході розробки	Середня	Середній	4	Попередження

3 ПРОЕКТУВАННЯ АНАЛІТИЧНОЇ ПІДСИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПО ВИБОРУ МЕТОДОЛОГІЇ УПРАВЛІННЯ ІТ-ПРОЕКТОМ

3.1 Структурно-функціональне моделювання

Перед початком розробки необхідно змоделювати процес проведення робіт, детально зобразити усі процеси, етапи їх виконання та можливі варіанти розвитку подій. Для цього необхідно побудувати діаграму IDEF0 та діаграму варіантів використання.

Першим кроком є побудова контекстної діаграми IDEF0(рис.3.1), яка декомпозується на 4 блоки(рис.3.2). На вході маємо такий пункт, як потреба в розробці аналітичної підсистеми. Елементами управління виступають методології управління ІТ - проектами та методичні вказівки. Механізми управління: розробник, програмне забезпечення, апаратне забезпечення, керівник проекту.

Аналіз предметної області включає в себе визначення пакету робіт, необхідний для досягнення певних цілей.

Етап створення підсистеми у вигляді веб-орієнтованого аналітичного рішення декомпозується на 3 пункти(рис.3.3) та включає в себе реєстрацію на вебхостингу, створення php скрипту та завантаження його на веб хостинг та підключення сітки bootstrap.

Створення php скрипту та завантаження його на веб хостинг декомпозується на такі пункти як: створення форми з питаннями, створення алгоритму вибору методології по заданим питанням, налаштування виводу результату роботи алгоритму(рис.3.4).

Перевірка функціонування системи. Керівник проекту перевіряє розроблену аналітичну підсистему на відповідність вимогам.

Заключним етапом є створення звіту, розроблюється документація та опис робіт, яка використовувалася для досягнення поставленої цілі.

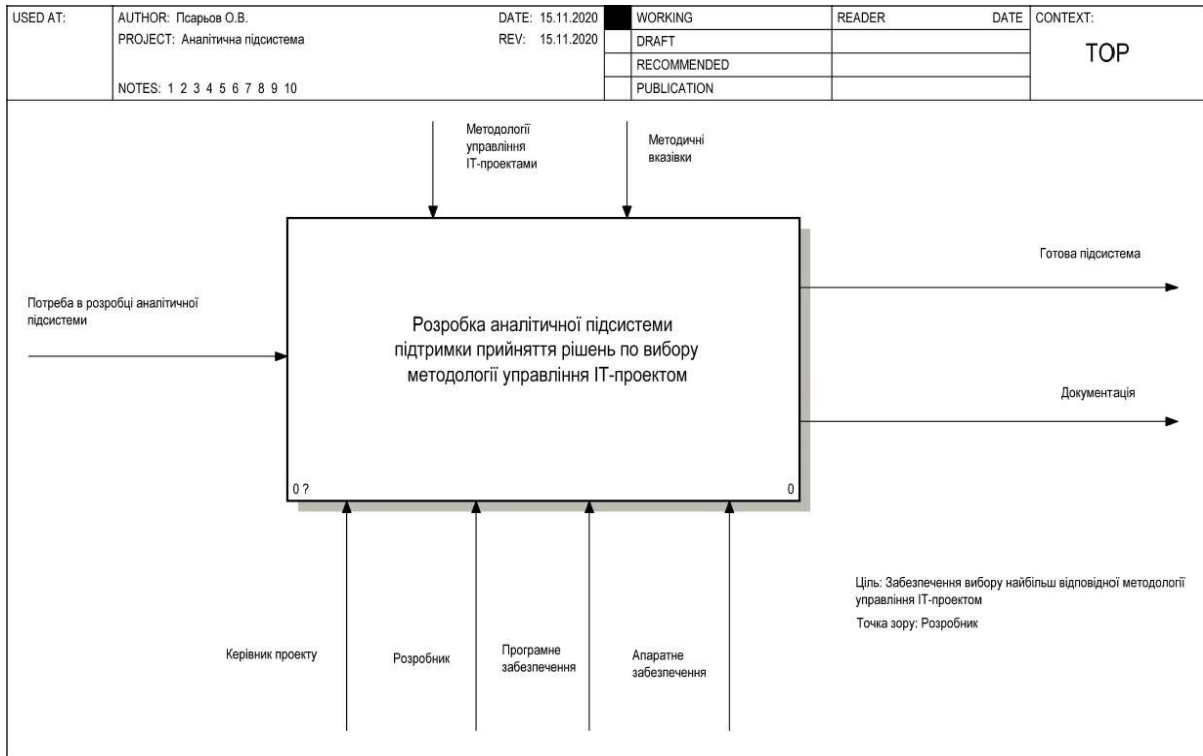


Рисунок 3.1 – Діаграма IDEF0

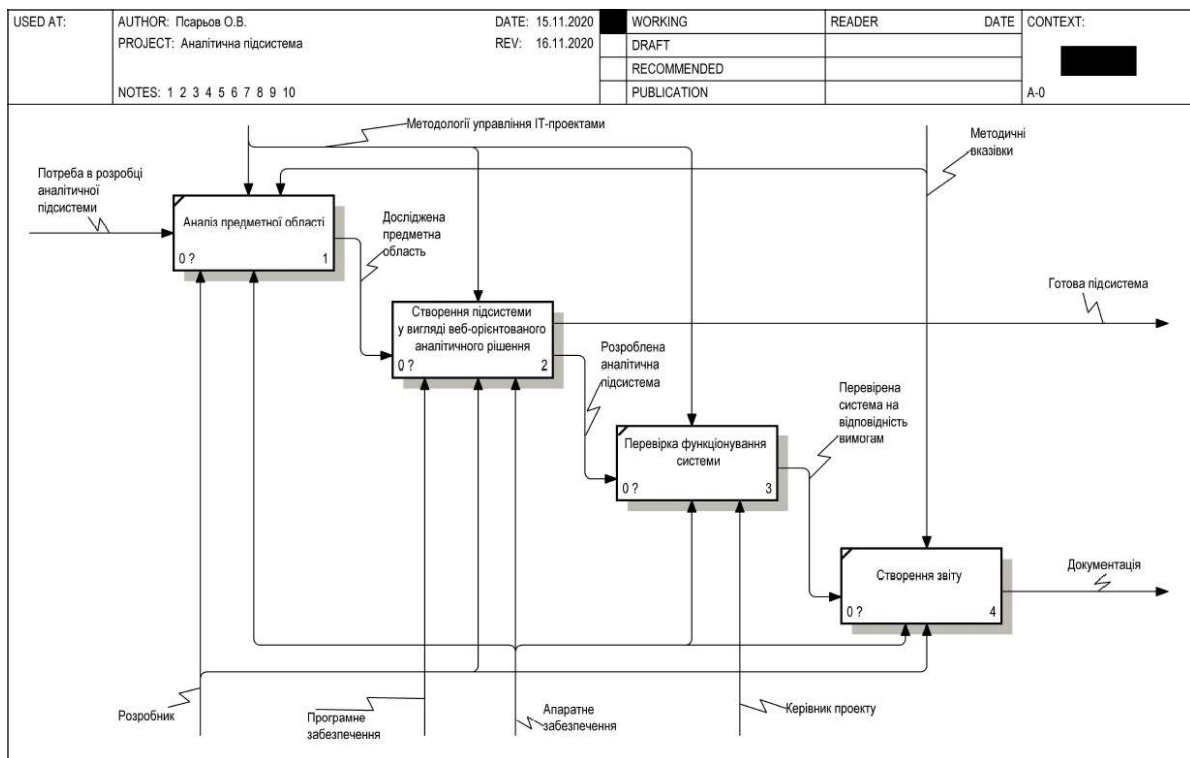


Рисунок 3.2 – Декомпозиція контекстної діаграми

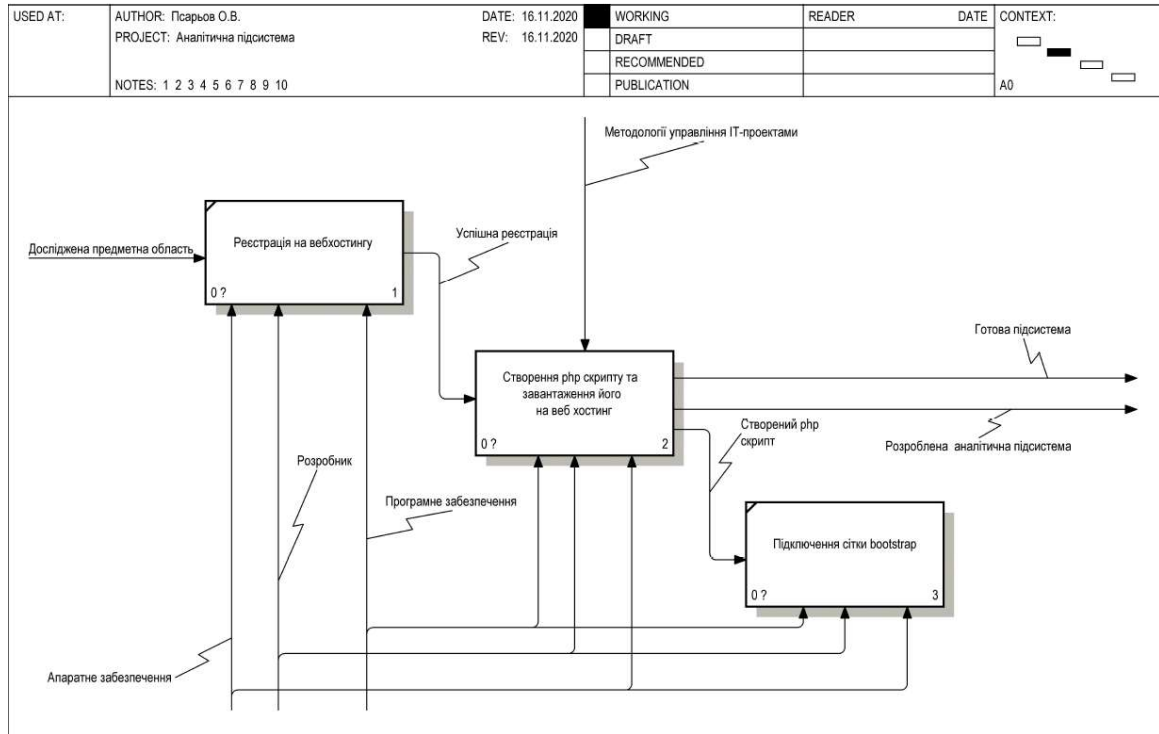


Рисунок 3.3 – Декомпозиція процесу Створення підсистеми у вигляді веб-орієнтованого аналітичного рішення

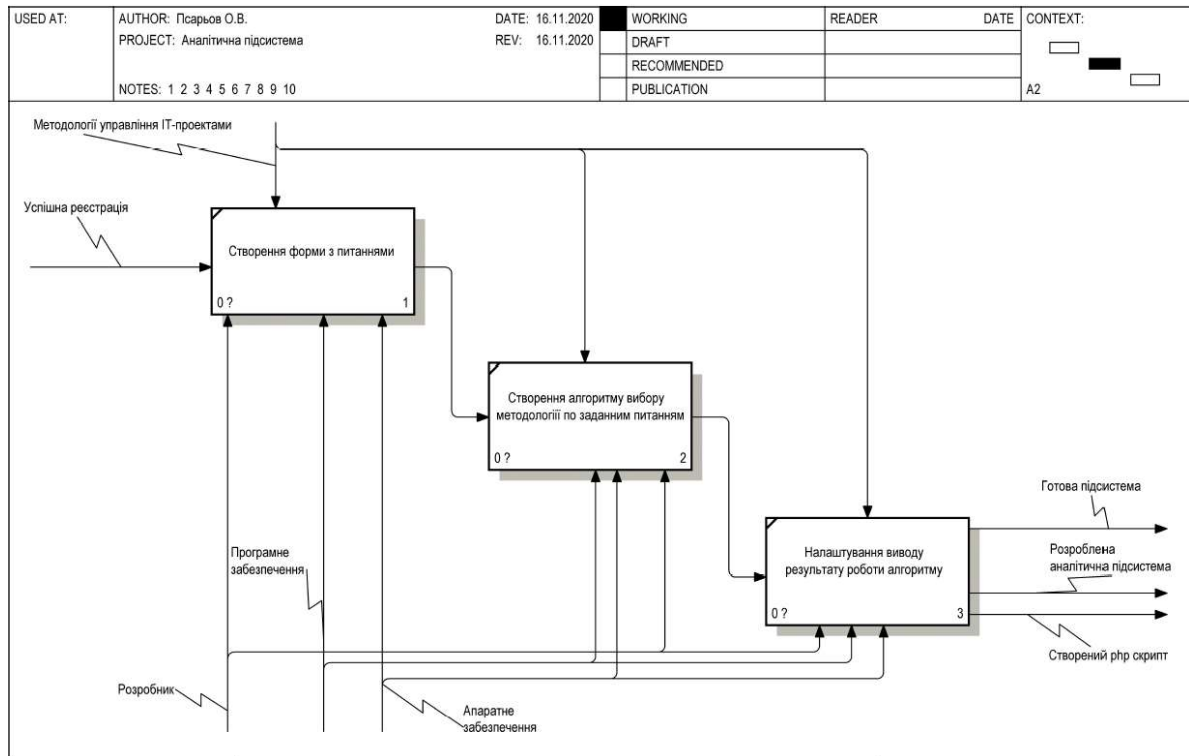


Рисунок 3.4 – Декомпозиція процесу Створення ррр скрипту та завантаження його на веб хостинг

Опис блоків та стрілок наведено в таблицях 3.1-3.2.

Таблиця 3.1 – Опис блоків

Name (Activities)	Definition
Аналіз предметної області	Огляд факторів впливу на успішність ІТ- проекту, роль прийняття рішень в управлінні ІТ – проектами, огляд методологій управління ІТ – проектами
Створення звіту	Звіт з виконання робіт (результати)
Створення підсистеми у вигляді веб-орієнтованого аналітичного рішення	Реєстрація на вебхостингу, створення php скрипту та завантаження його на веб хостинг, підключення сітки bootstrap.

Таблиця 3.2 – Опис стрілок

Name (Arrows)	Definition
Апаратне забезпечення	Необхідність використання ПК для розробки аналітичної підсистеми
Програмне забезпечення	Використання програмного забезпечення та технологій для розробки
Документація	Опис системи та написання звіту про виконану роботу
Досліджена предметна область	Розглянуті фактори впливу на успішність ІТ- проекту, ролі прийняття рішень в управлінні ІТ – проектами, методології управління ІТ – проектами
Керівник проекту	Перевіряє систему на відповідність вимогам
Розробник	Виконує розробку аналітичної підсистеми, проводить аналіз предметної області та створює звіт з виконаної роботи

3.2 Створення моделі варіантів використання

Для відображення взаємодії між акторами та взаємодії користувача з системою було побудовано діаграму варіантів використання(рис.3.5).

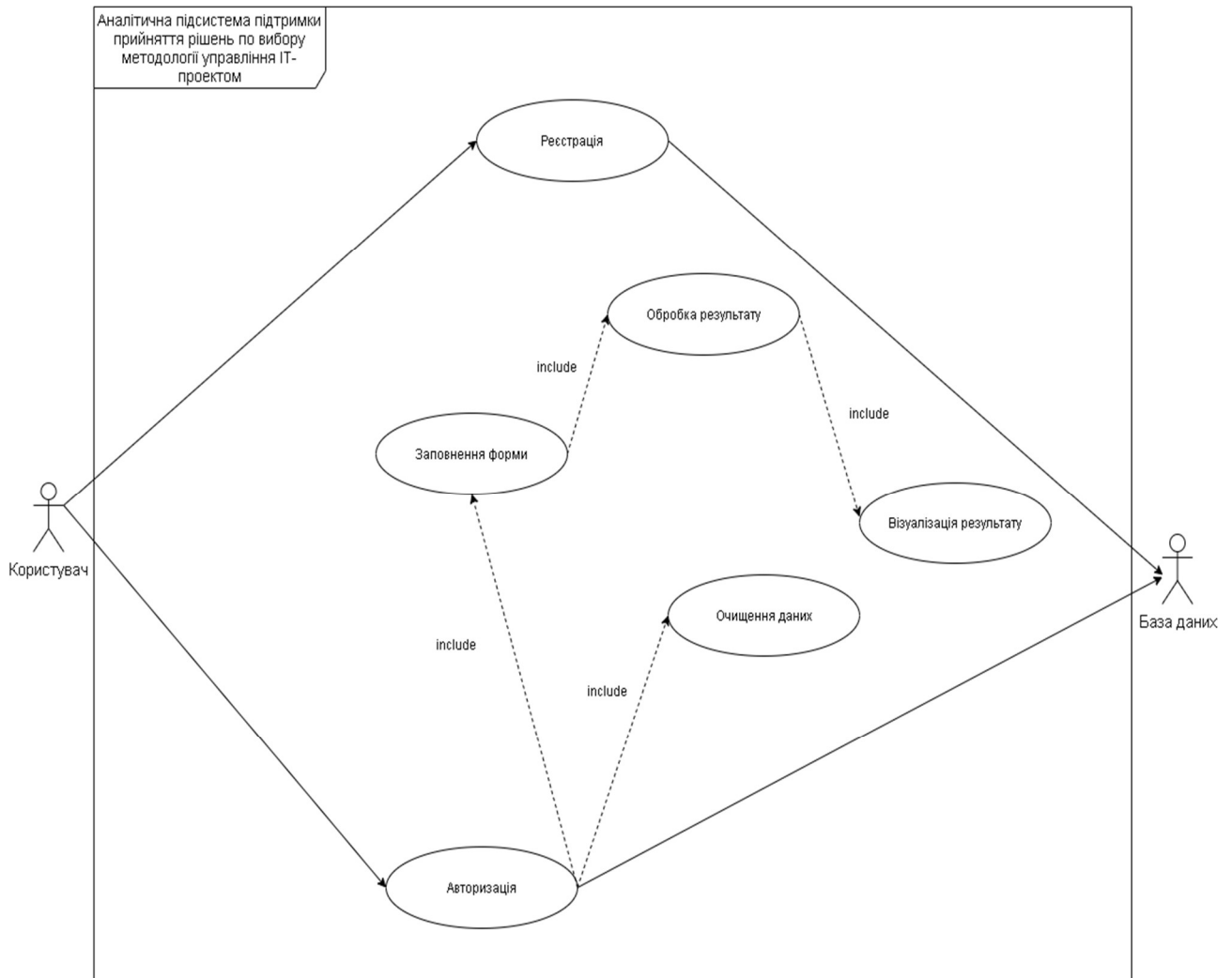


Рисунок 3.5 – Діаграма варіантів використання

4 РОЗРОБКА АНАЛІТИЧНОЇ ПІДСИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПО ВИБОРУ МЕТОДОЛОГІЇ УПРАВЛІННЯ ІТ-ПРОЕКТОМ

4.1 Налаштування інструментів для розробки

Перед початком роботи, необхідно пройти процес Реєстрації на веб-хостингу(рис. 4.1).

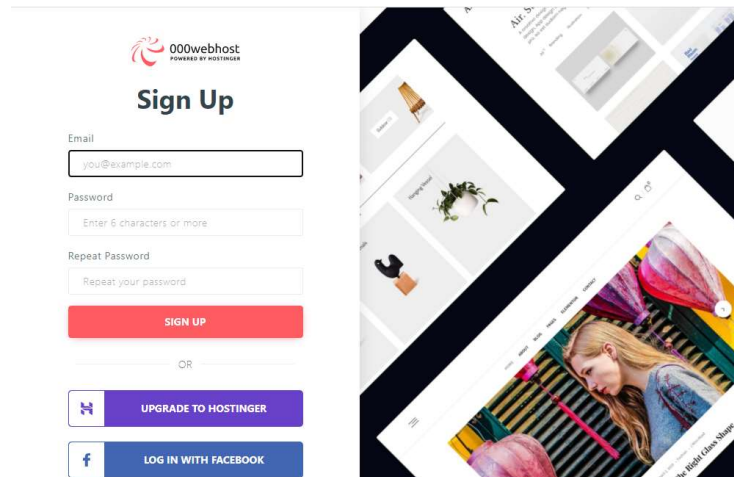


Рисунок 4.1 – Форма реєстрації на веб-хостингу

Наступним кроком є створення шаблону для аналітичної підсистеми. Для цього потрібно на головній сторінці натиснути кнопку «Создать сайт». Далі в списку сайтів з'явиться створений нами шаблон(рис.4.2).

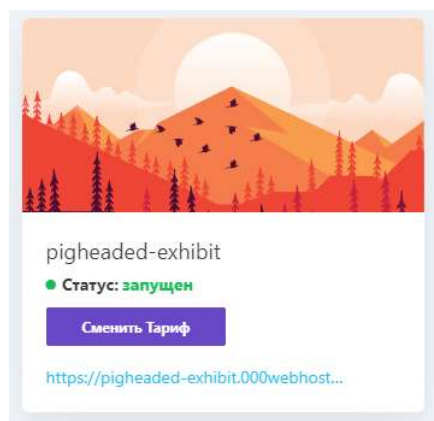


Рисунок 4.2 – Відображення створеного шаблону

Після успішного створення шаблону, переходимо до етапу створення Баз Даних, тому що є необхідність зберігати дані реєстрації користувачів. Створення БД за допомогою інструментарію Веб-хостингу(рис.4.3).

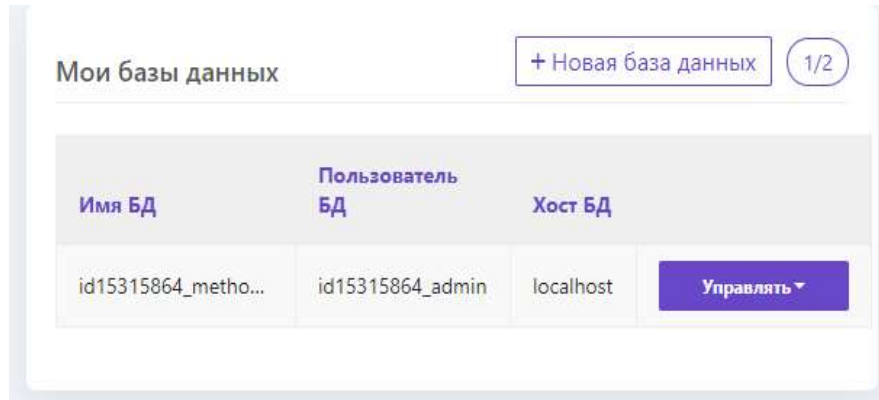


Рисунок 4.3 – Створення БД

Для того щоб зберігати інформацію про зареєстрованих користувачів необхідно в БД створити таблицю з відповідними полями.

На рис. 4.4 Наведено етап створення таблиці users.

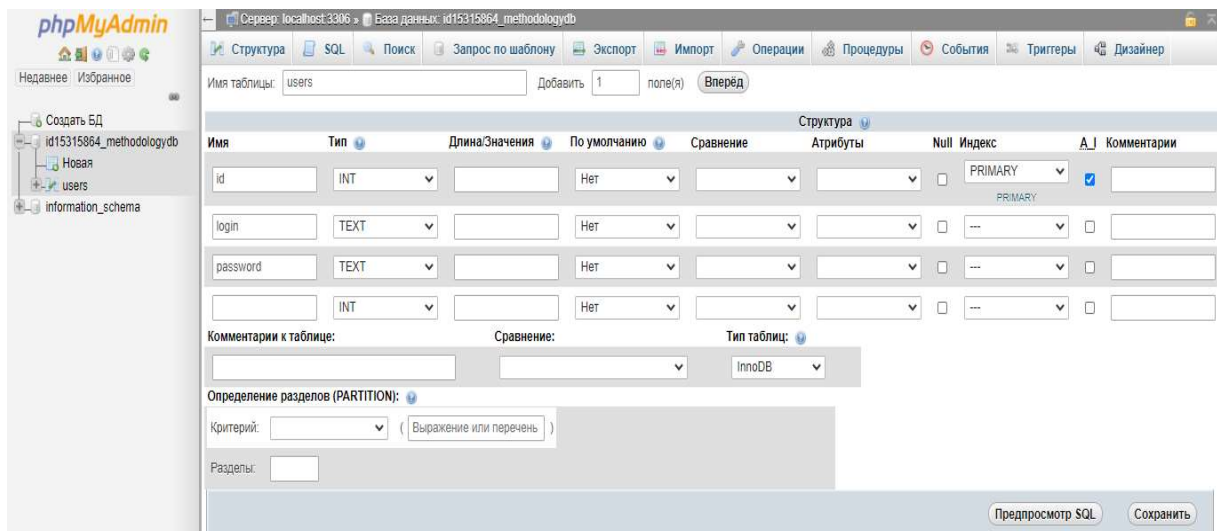


Рисунок 4.4 – Додавання таблиці users до БД

Структура таблиці users повинна містити такі поля:

- id – особистий ідентифікатор;
- login - це слово, яке буде використовуватися для входу на сайт;

- password – стовбчик зберігання паролю для входу.

Таблиця users та її структура(рис.4.5).

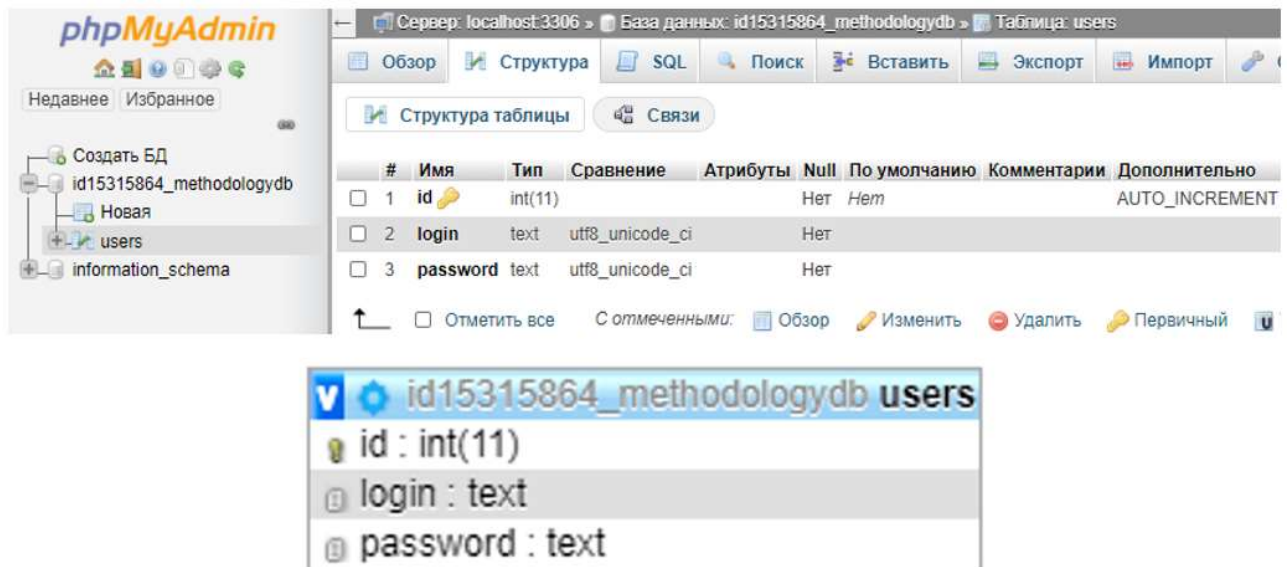


Рисунок 4.5 – Структура таблиці users

Після успішної реєстрації виконується запис відповідних даних до таблиці users(рис.4.6).

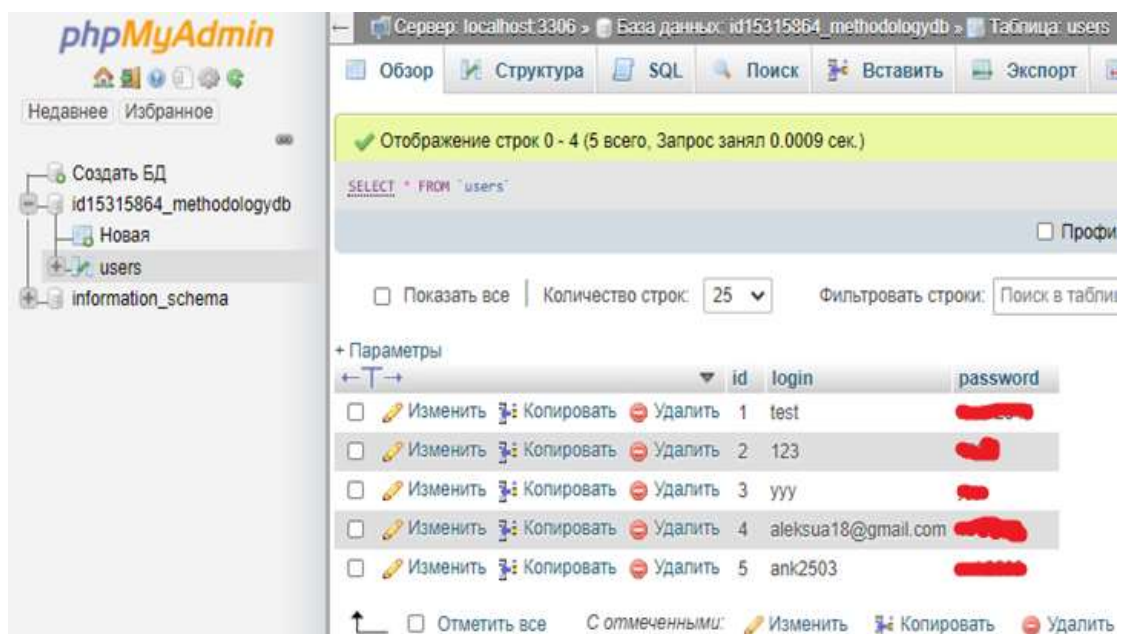


Рисунок 4.6 – Таблиця users з даними

Наявність файлового менеджера дає можливість завантажувати та швидко редагувати необхідні файли(рис.4.7).

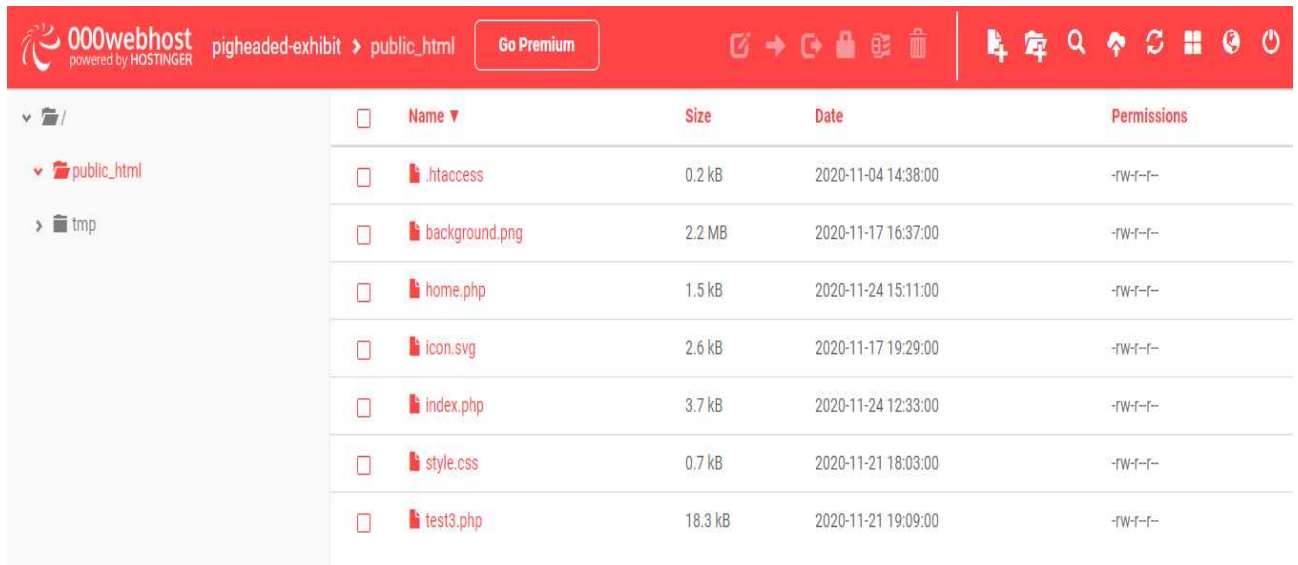


Рисунок 4.7 – Використання файлового менеджера

4.2 Розробка веб-орієнтованого аналітичного рішення

Перед початком роботи необхідно побудувати схему процесів розробки аналітичної підсистеми(рис.4.8), для того, щоб визначити необхідний пакет робіт, більш детально зрозуміти, з чим саме будемо працювати, та що представлятиме собою кінцевий продукт. На вході маємо такі дані: список методологій управління IT-проектами, досвід з попередніх проектів, характер кожної методології управління IT-проектном. Після аналізу цих даних переходимо до формування єдиних критеріїв вибору методології управління проектами, далі, після сформованих критеріїв переходимо до побудови дерева рішень, наступним кроком є реалізація алгоритму по вибору методології на основі сформованого дерева. На виході маємо готовий продукт, який дасть змогу проектним менеджерам спростити процес прийняття рішень по вибору методології.

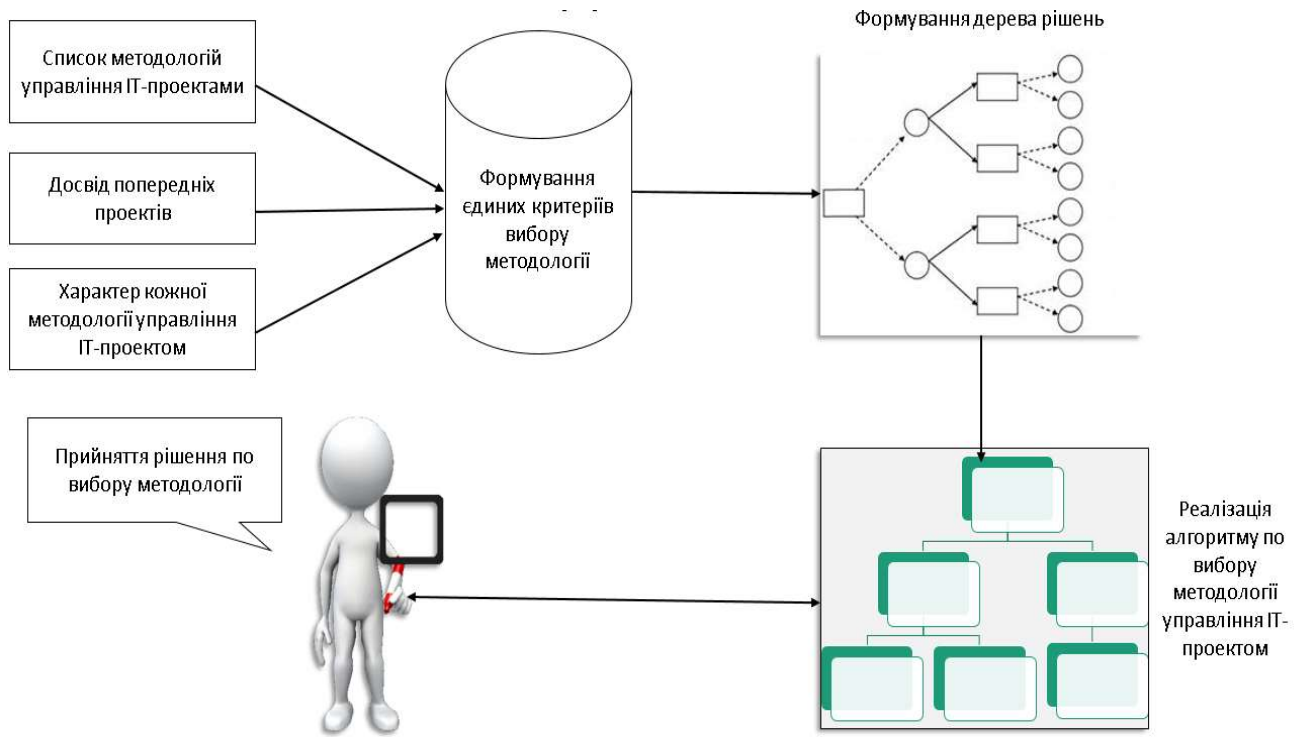


Рисунок 4.8 – Схема процесів розробки аналітичної підсистеми

Дерево рішень продемонстровано на рис. 4.9 – 4.10.

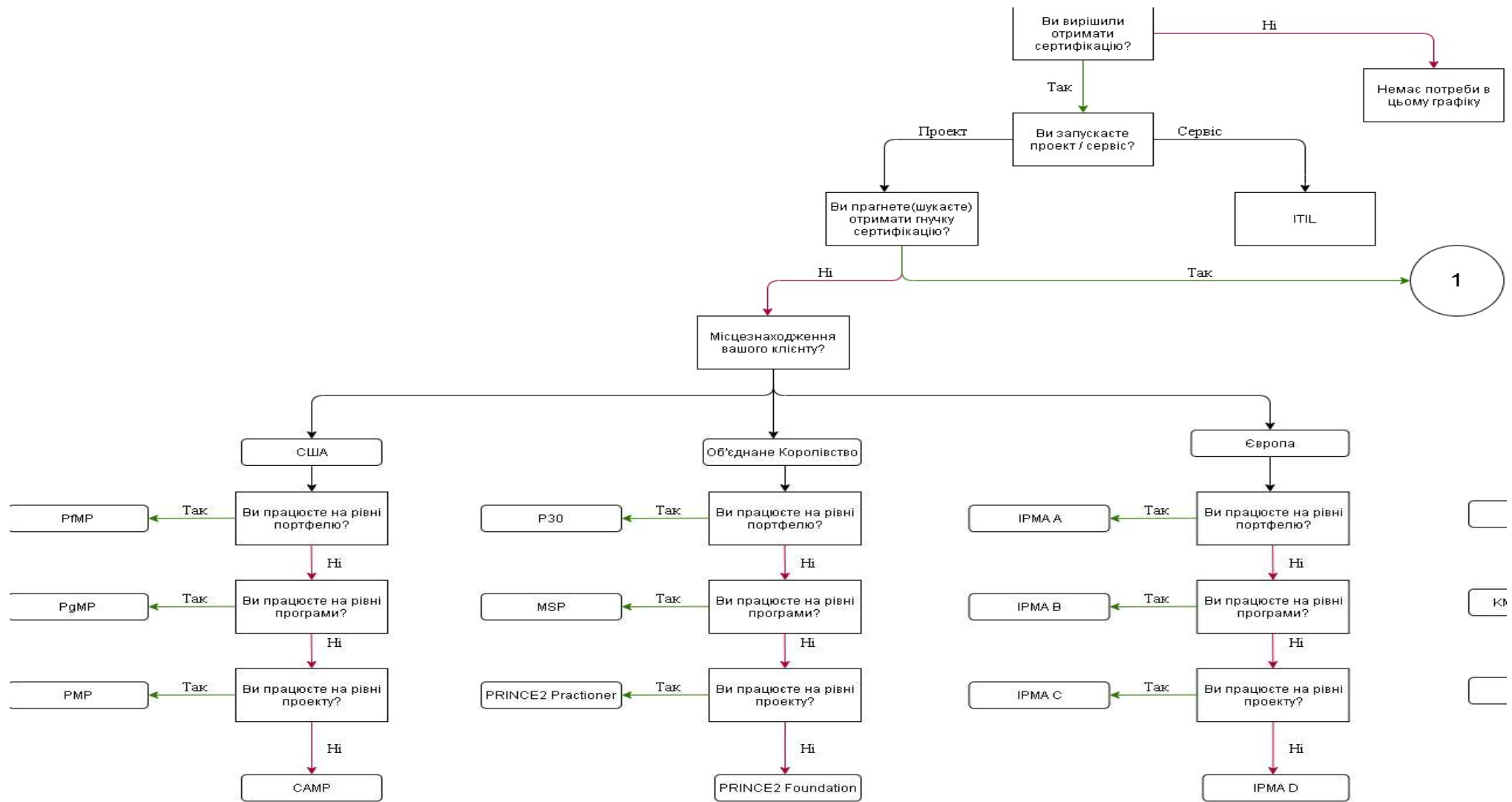


Рисунок 4.9 – Дерево рішень по вибору методології управління ІТ-проектом

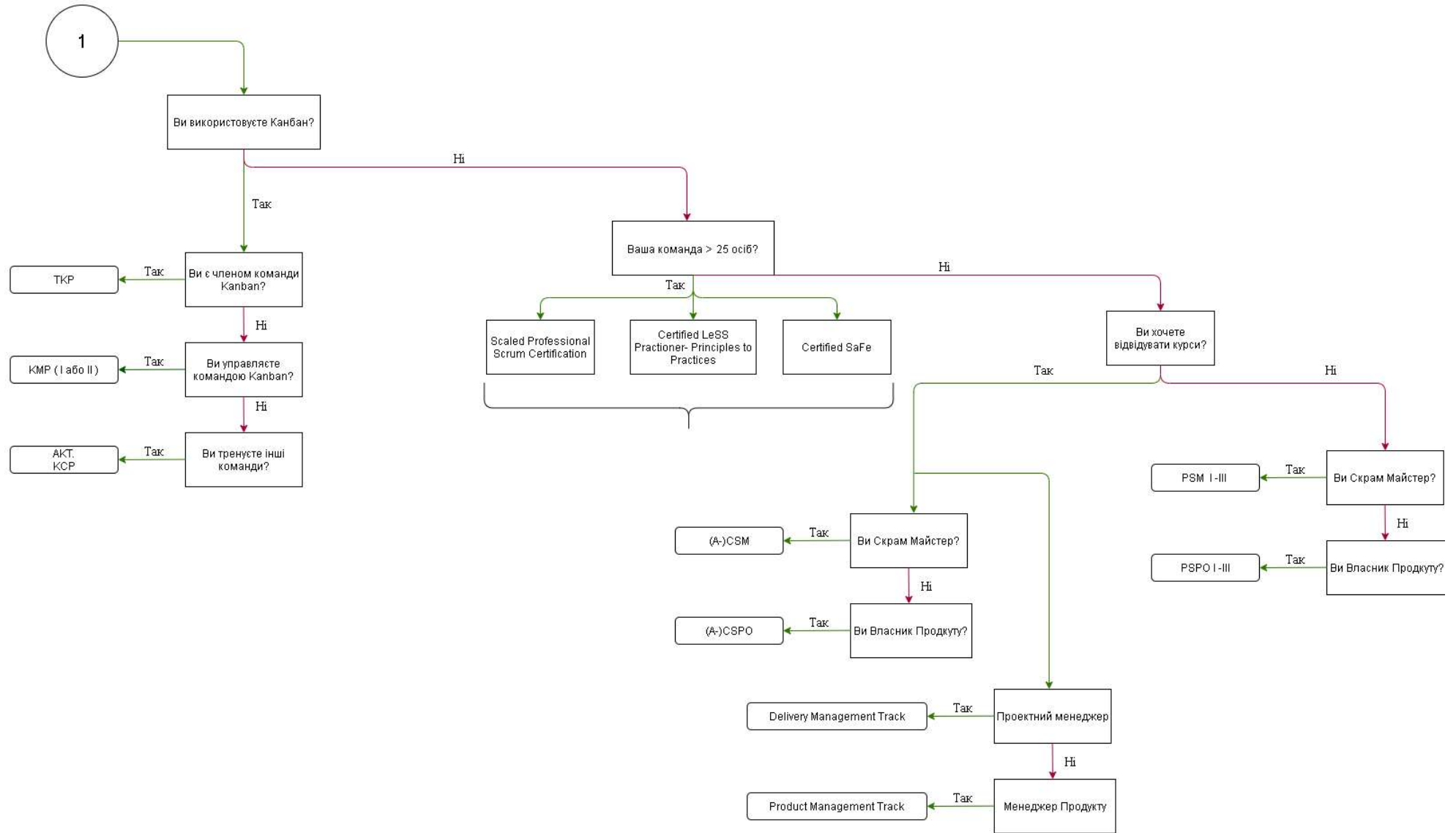


Рисунок 4.10 – Дерево рішень по вибору методології управління IT-проектом (продовження)

При реєстрації також відбувається підключення до БД та надсилання SQL запиту, але не на зчитування таблиці users, а на запис. Також виконується перевірка унікальності логіна. Двоє користувачів не можуть мати однаковий логін.

Скрипт реєстрації користувачів(рис.4.13).

```

53 <?php
54 if(isset($_POST["btn2"])){
55 if(!empty($_POST['login1']) && !empty($_POST['password1'])) {
56 $connect=mysqli_connect('localhost', 'id15315864_admin', 'Qwerty12345@', 'id15315864_methodologydb');
57 $login1=mysqli_real_escape_string($connect,$_POST['login1']);
58 $password1=mysqli_real_escape_string($connect,$_POST['password1']);
59 $query=mysqli_query($connect,"SELECT * FROM `users` WHERE login='{$_login1}'");
60 $numr=mysqli_num_rows($query);
61 if($numr==0)
62 {
63 $sql_q="INSERT INTO `users`
64 (login,password)
65 VALUES('{$_login1}','{$_password1}')";
66 $res=mysqli_query($connect,$sql_q);
67 if($res){
68 echo "Аккаунт успішно створений";
69 }
70 else {
71 echo "Не вдалося додати інформацію";
72 }
73 }
74 else {
75 echo '<p class="formstyle">Цей нік зайнятий. Спробуйте інший!</p>';
76 }

```

Рисунок 4.13 – Скрипт реєстрації користувачів

Для створення php-скрипту першим етапом, для зручності ініціалізуємо змінні які зберігатимуть відповідні питання які визначені у дереві рішень (рис.4.14).

/public_html/test3.php

```

1  <?php
2
3      error_reporting(0);
4
5      $question1 = "Ви вирішили отримати сертифікацію?";
6      $question2 = "Ви запускаєте проект / сервіс?";
7      $question3 = "Ви прагнете(шукаєте) отримати гнучку сертифікацію?";
8      $question4 = "Місцезнаходження вашого клієнту?";
9      $question5 = "Ви використовуєте Канбан?";
10     $question6 = "Ви працюєте на рівні портфелю?";
11     $question7 = "Ви працюєте на рівні програми?";
12
13     $question8 = "Ви працюєте на рівні проекту?";
14     $question9 = "Ви є членом команди Канбан?";
15     $question10 = "Ви управляєте командою Канбан?";
16     $question11 = "Ви тренуєте інші команди?";
17     $question12 = "Ваша команда > 25 осіб?";
18     $question13 = "Ви хочете відвідувати курси?";
19     $question14 = "Ви Скрам Майстер?";
20     $question15 = "Ви Власник Продукту?";
21     $question16 = "Ви Проектний менеджер?";
22     $question17 = "Ви Менеджер Продукту?";
23

```

Рисунок 4.14 – Ініціалізація змінних

Наступним етапом буде створення змінної `check` яка зберігатиме значення 0 або 1 за такої умови:

- 0 якщо немає жодної відповіді на питання;
- 1 якщо є хоча б одна відповідь на питання.

Відповідно це необхідно для того щоб при використанні `switch` відображались певні данні.

У нашому випадку `switch` працює у двох режимах, а саме:

- коли 0 – відображається увесь список питань;
- коли 1 – виводиться відповідна відповідь по обраним питанням.

На рис. 4.15 зображено фрагмент коду ініціалізації умов параметрів змінної `check` та вивід даних за умов якщо `check` дорівнює 0.

/public_html/test3.php

```

24 ▾   if($_POST['q1'] == null){
25     $check = 0;
26 ▾   } else {
27     $check = 1;
28   }
29
30
31
32 ▾   switch($check){
33
34     case 0:
35
36     echo '<form action="home.php" method="POST"><div class="centerstyle"><br><br><br><br>';
37
38     echo '<h3 class="display-3">'.$question1.'</h3>';
39     echo '<p><label><input name="q1" type="radio" value="yes">&nbsp;Так</label>&nbsp;&nbsp;&nbsp;';
40     echo '<label><input name="q1" type="radio" value="no">&nbsp;Ні</label></p><hr>';
41
42     echo '<h3 class="display-3">'.$question2.'</h3>';
43     echo '<p><label><input name="q2" type="radio" value="Проект">&nbsp;Проект</label>&nbsp;&nbsp;&nbsp;';
44     echo '<label><input name="q2" type="radio" value="Сервіс">&nbsp;Сервіс</label></p><hr>';

```

Рисунок 4.15 – Ініціалізація змінної check та вивід даних якщо check дорівнює нулю

Якщо виконується умова за якої check дорівнює 1 тоді оператор switch виконує фрагмент коду за яким виконується алгоритм, за яким виводиться результат оптимальної методології.

На рис. 4.16 наведений фрагмент коду за яким check дорівнює 1.

/public_html/test3.php

```

113
114     case 1:
115
116     if($_POST['q1'] == "yes"){
117         if($_POST['q2'] == "Проект"){
118             if($_POST['q3'] == "yes"){
119                 if($_POST['q5'] == "yes"){
120                     if($_POST['q9'] == "yes"){
121                         echo '<div class="text-center" style="display: flex; justify-content: center; align
                            -items: center; height: 100vh"><h1 class="bg-primary text-white">TKP</h1></div>';
122
123                     }elseif($_POST['q9'] == "no"){
124                         if($_POST['q10'] == "yes"){
125                             echo '<div class="text-center" style="display: flex; justify-content: center;
                            align-items: center; height: 100vh"><h1 class="bg-primary text-white">KMP ( I
                            або II )</h1></div>';
126
127                         }elseif($_POST['q10'] == "no"){
128                             if($_POST['q11'] == "yes"){
129                                 echo '<div class="text-center" style="display: flex; justify-content: center;
                            align-items: center; height: 100vh"><h1 class="bg-primary text-white">АКТ
                            .KCP</h1></div>';
130
131                             }elseif($_POST['q11'] == "no"){
132                                 echo '<div class="text-center" style="display: flex; justify-content: center;

```

Рисунок 4.16 – Виконання алгоритму за умови якщо check дорівнює 1.

Алгоритм вибору методології був налаштований за допомогою вкладених операторів if та else. В залежності вибору відповіді того чи іншого питання виконується проходження шляху до потрібної методології по дереву рішень.

Оператор if є вкладеним, якщо він вкладений, тобто знаходиться всередині іншого оператора if або else. На практиці програмування вкладені умовні оператори використовуються досить часто. У вкладеному умовному операторі фраза else завжди асоціюється з найближчим if в тому ж блоці, якщо цей if не асоційований з іншого фразою else.

Стандарт C89 допускає 15 рівнів вкладеності умовних операторів, C99 - 127 рівнів. В даний час більшість компіляторів допускають значно більшу кількість рівнів вкладеності. Однак на практиці необхідність в глибині вкладеності, більшою, ніж кілька рівнів, виникає досить рідко, так як збільшення глибини укладення швидко заплутує програму і робить її нечитаною.

На рис. 4.17 схематично зображено блок – схему алгоритму вибору методології з використанням вкладених операторів if та else.

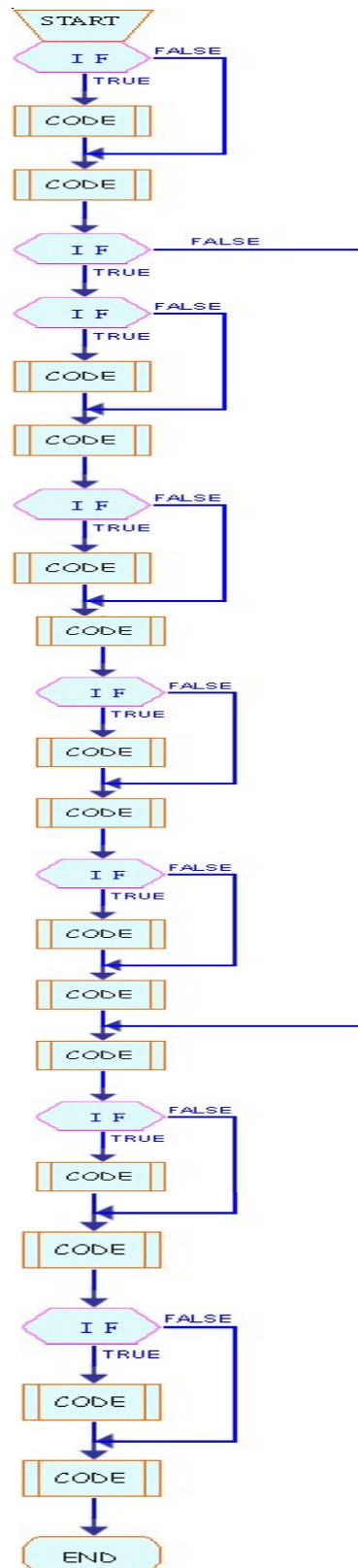


Рисунок 4.17 – Блок - схема алгоритму

Відповідно блок-схемі було налаштовано вкладені if та else. Частина коду з реалізації алгоритму(рис.4.18).

/public_html/test3.php

```

137     }
138     }elseif($_POST['q5'] == "no"){
139     if($_POST['q12'] == "yes"){
140         echo '<div class="text-center" style="display: flex; justify-content: center; align
            -items: center; height: 100vh"><h1 class="bg-primary text-white">Scaled
            Professional Scrum Certification<br><br>Certified LeSS Practioner- Principles to
            Practices<br><br>Certified SaFe</h1></div>';
141
142     }elseif($_POST['q12'] == "no"){
143     if($_POST['q13'] == "yes"){
144     if($_POST['q14'] == "yes"){
145         echo '<div class="text-center" style="display: flex; justify-content: center;
            align-items: center; height: 100vh"><h1 class="bg-primary text-white">(A
            -)CSM</h1></div>';
146
147     }elseif($_POST['q14'] == "no"){
148     if($_POST['q15'] == "yes"){
149         echo '<div class="text-center" style="display: flex; justify-content:
            center; align-items: center; height: 100vh"><h1 class="bg-primary
            text-white">(A-)CSPO</h1></div>';
150
151     }elseif($_POST['q15'] == "no"){
152         //-----
153         echo '<div class="text-center" style="display: flex; justify-content:
            center; align-items: center; height: 100vh"><h1 class="bg-primary

```

Рисунок 4.18 – Фрагмент коду алгоритму

Повний лістинг програмного коду представлено у додатку (Б).

Наступним кроком, для коректного відображення всіх елементів на екрані та налаштування дизайну був bootstrap та додатковий файл стилів CSS(рис.4.19).

Підключення bootstrap та додаткового файлу стилів style.css необхідно всередині HTML тега <head>.

/public_html/index.php

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8" />
5     <title>Підсистема аналітичного рішення</title>
6     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity
        ="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">
7     <link rel="stylesheet" href="style.css">

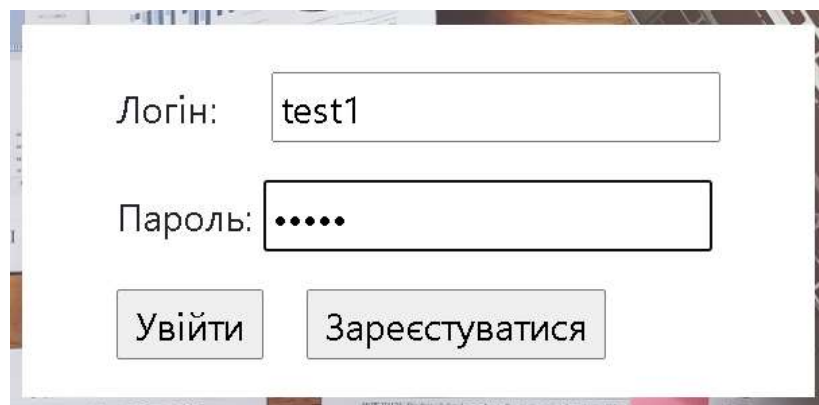
```

Рисунок 4.19 – Підключення bootstrap та style.css

4.3 Тестування системи

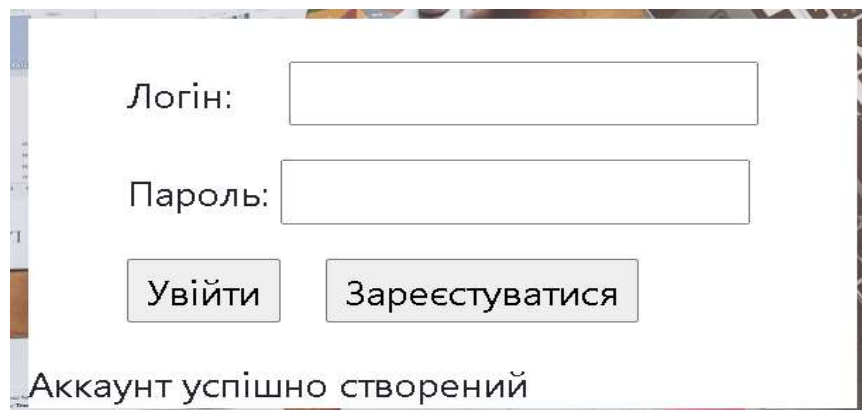
Після реалізації аналітичної підсистеми підтримки прийняття рішень, необхідно протестувати роботу програмного продукту.

Перед тим, як зайти до системи, перед нами з'являється досить проста форма реєстрації(рис.4.20). Після заповнення полів з логіном та паролем потрібно натиснути на кнопку «Зареєструватися», після натискання з'являється надпис про те, що користувач успішно зареєстрований(рис.4.21). У випадку коли користувач повторно заповнює форму та з такими ж даними та натискає «Зареєструватися», то з'являється попередження про те, що даний нікнейм вже зайнятий (рис.4.22).



The image shows a web registration form. It contains two input fields: the first is labeled 'Логін:' and contains the text 'test1'; the second is labeled 'Пароль:' and contains five dots. Below the fields are two buttons: 'Увійти' (Login) and 'Зареєструватися' (Register).

Рисунок 4.20 – Форма реєстрації



The image shows the same registration form as in Figure 4.20, but with a confirmation message at the bottom: 'Аккаунт успішно створений' (Account successfully created). The input fields are now empty.

Рисунок 4.21 – Повідомлення про успішну реєстрацію

Логін:

Пароль:

Цей нік зайнятий. Спробуйте інший!

Рисунок 4.22 – Попередження при реєстрації

Далі вводимо дані, введені при реєстрації та потрапляємо на головну сторінку(рис.4.23).

Підсистема аналітичного рішення

Ви вирішили отримати сертифікацію?
 Так Ні

Ви запускаєте проект / сервіс?
 Проект Сервіс

Ви прагнете(шукаєте) отримати гнучку сертифікацію?
 Так Ні

Місцезнаходження вашого клієнту?
 США Об'єднане Королівство Європа

Ви використовуєте Канбан?
 Так Ні

Ви працюєте на рівні портфелю?
 Так Ні

Ви працюєте на рівні програми?
 Так Ні

Ви працюєте на рівні проекту?
 Так Ні

Ви є членом команди Канбан?
 Так Ні

Powered by Alexander_Psaryov
2020

Рисунок 4.23 – Головна сторінка

Для отримання результату необхідно обрати всі потрібні пункти в формі заповнення та натиснути на кнопку «Отримати результат». Результат виконання

може відрізнятись в залежності від вибору різних показників в формі заповнення (на головній сторінці).

Результати виконання, з урахуванням вибору різних показників на головній сторінці(рис.4.24).

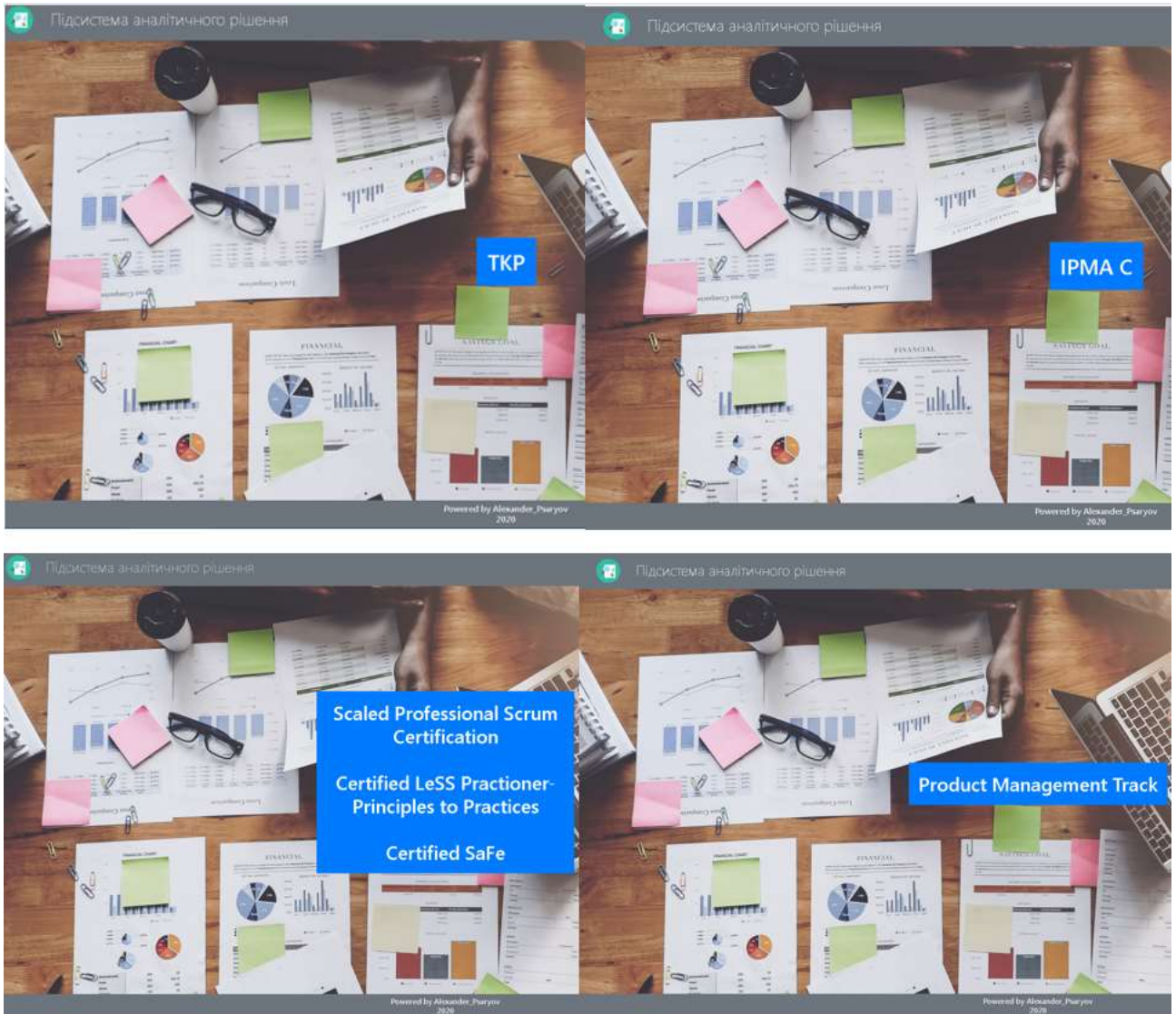


Рисунок 4.24 – Результати виконання, з урахуванням вибору різних показників

ВИСНОВКИ

Методологія управління проектами - це суворо визначена комбінація практик, методів та процесів, що визначають, як найкраще планувати, розробляти та контролювати проект протягом безперервного процесу його реалізації та успішного завершення. В управлінні проектами існують різні підходи та методи, які можуть бути використані в різних типах управління проектами. Усі типи методологій проектів можна розділити на традиційні та сучасні підходи.

У ході дослідження було запропоновано аналітичну підсистему підтримки прийнятті рішень по вибору методології управління ІТ-проектами, що має мету допомогти менеджерам ІТ-проектів ефективно визначити, оцінити та зважити плюси та мінуси кожної методології та обрати найкращу методологію для команди, організації та зацікавлених сторін, вдосконалити свої щоденні задачі протягом всього проекту, зменшити час на прийняття рішень та підвищити якість виконання ІТ-проектів.

В результаті проведеного наукового дослідження було розроблено підсистему у вигляді веб-орієнтованого аналітичного рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. S. M. Qureshi and C. Kang, “Analysing the organizational factors of project complexity using structural equation modelling,” *International Journal of Project Management*, vol. 33, no. 1, pp. 165–176, 2015.
2. E. C. Conforto, E. Rebentisch, and D. Amaral, “Learning the art of business improvisation,” *MIT Sloan Management Review*, vol. 57, p. 8, 2016.
3. Sanchez, M.A. Integrating sustainability issues into project management. *J. Clean. Prod.* 2015, 96, 319–330.
4. Huemann, M.; Silvius, G. Projects to create the future: Managing projects meets sustainable development. *Int. J. Proj. Manag.* 2017, 35, 1066–1070.
5. Duan, S.; Deng, H. A Multi-Criteria Analysis Approach for the Evaluation and Selection of IS Projects—A Sustainability Perspective. In *Proceedings of the Pacific Asian Conference on Information Systems, Yokohama, Japan, 2–9 July 2018*.
6. Han, Y.; Deng, Y. An Enhanced Fuzzy Evidential DEMATEL Method with its Application to Identify Critical Success Factors. *Soft Comput.* 2018, 22, 5073–5090.
7. What is a Decision Support System (DSS)?, [Електронний ресурс], – Режим доступу до ресурсу: <https://corporatefinanceinstitute.com/resources/knowledge/other/decision-support-system-dss/>
8. PRINCE2: An Introduction to the Project Management Methodology, [Електронний ресурс], – Режим доступу до ресурсу: <https://www.projectmanager.com/blog/prince2-methodology>
9. Kagan, E., Leider, S., & Lovejoy, W.S. (2017) Ideation–Execution Transition in Product Development: An Experimental Analysis. *Management Science*, forthcoming.
10. Besteiro, E. N. C., de Souza Pinto, J., Novaski, O. (2015). Success factors in project management. *Business Management Dynamics*, 4(9), 19–34.

11. Carvalho, M. M., Patah, L. A., Bido, D. S. (2015). Project management and its effects on project success: Cross-country and cross-industry comparisons. *International Journal of Project Management*, 33, 1509–1522.
12. Project Management Institute (PMI) . (2017). A guide to the project management body of knowledge (PMBOK® guide) – Sixth edition. Newtown Square, PA: Author.
13. Joslin, R. Müller, Relationships between a project management methodology and project success in different project governance contexts, *International Journal of Project Management* 33, 2015, pp. 1377–1392.
14. The Definitive Guide to Project Management Methodologies, [Электронный ресурс], – Режим доступа до ресурсу: <https://www.workamajig.com/blog/project-management-methodologies>
15. Al-Baik O, Miller J (2015) The kanban approach, between agility and leanness: a systematic review. *Empir Softw Eng* 20:1861– 1897.
16. Heikkilä VT, Paasivaara M, Lassenius C (2016) Teaching University students Kanban with a collaborative board game. In: *Proceedings of the 38th international conference on software engineering companion*. ACM, New York, pp 471–480.
17. What is Agile Software Development?, [Электронный ресурс], – Режим доступа до ресурсу: <https://www.guru99.com/agile-scrum-extreme-testing.html>
18. Joslin, R., & Müller, R. (2016). The impact of project methodologies on project success in different project environments. *International Journal of Managing Projects in Business*, 9(2), 364–388. <http://doi.org/10.1108/IJMPB-03-2015-0025>
19. Biggins, David. 2015. Perspectives on Project Management Methods. British Academy of Management Conference, Portsmouth, 8 - 10 Sep 2015. [Online]. <http://eprints.bournemouth.ac.uk/22483/3/PMM%2520perspectives%2520-%2520BAM%25202015.docx.pdf>

20. Bondale, K. (2017). ProjectTimes.com. Retrieved from PM times Resources for Project Managers: <https://www.projecttimes.com/kiron-bondale/the-rights-and-responsibilities-of-project-managers.html>
21. Zeni, T. A., Buckley, M. R., Mumford, M. D., & Griffith, J. A. (2016). Making “sense” of ethical decision-making. *The Leadership Quarterly*, 27(6), 838-855.
22. Jones, G. R., & George, J. M. (2016). *Contemporary Management Ninth Edition*. 2 Penn Plaza, New York, NY 10121: McGraw-Hill Education.
23. Guan, D. (2007). Conflicts in the project environment. 2007 PMI Global Congress Proceedings.
24. Rzeszutek, M., Szyszka, A., & Czerwonka, M. (2015). Investors’ expertise, personality traits and susceptibility to behavioral biases in the decision making process. *Contemporary Economics*, 9(3), 337-352.
25. Cooper, R., Edgett, S., & Kleinschmidt, E. (2001). Portfolio management for new product development. *r&D Management*, 31(4), 361-380.
26. De Reyck, B., Grushka-Cockayne, Y., Lockett, M., Calderini, S. R., Moura, M., & Sloper, A. (2005). The impact of project portfolio management on information technology projects. *International Journal of Project Management*, 23(7), 524- 537.
27. Decision Making Models In Project Management?, [Электронный ресурс], – Режим доступа до ресурсу: <https://www.projecttimes.com/articles/decision-making-models-in-project-management.html>
28. J. Sheffield, J. Lemétayer, Factors associated with the software development agility of successful projects, *International Journal of Project Management*, (31) 2 (2013), pp. 459–472.
29. Jiang, J.J. & Klein, G., 1999. Project selection criteria by strategic orientation. *Information & Management*, 36(2), pp.63 75.
30. Meredith, J. R. and Mantel, Jr., S. J. (2003), *Project Management*, 5th ed. New York: Wiley

Додаток А

Планування робіт

Структуру розподілу робіт зображено на рис. А.1.

Організаційна структура проекту (ОBS) представлена на рис. А.2.

Діаграму Ганта наведено на рис. А.3.

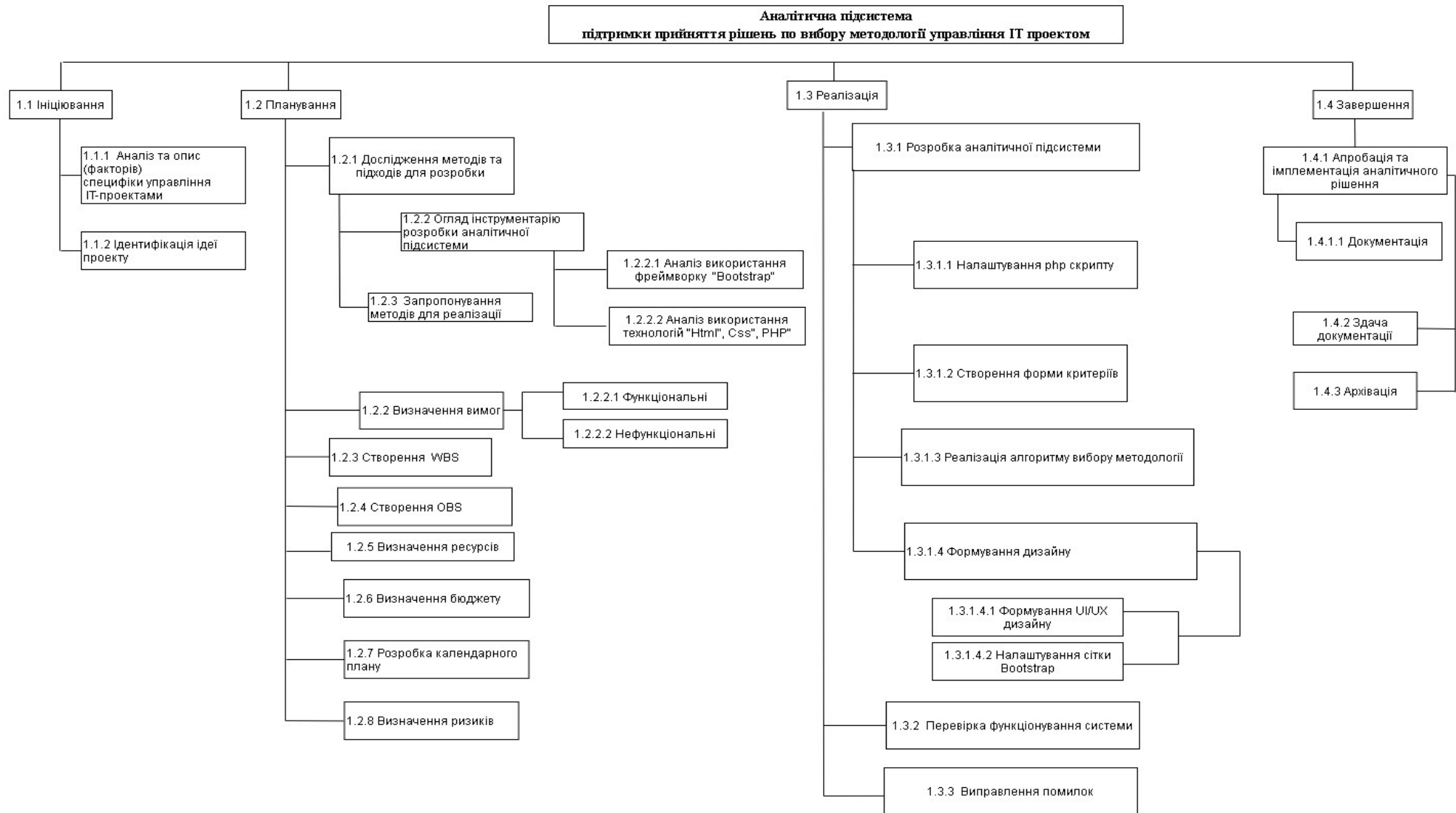


Рисунок А.1 – Діаграма WBS

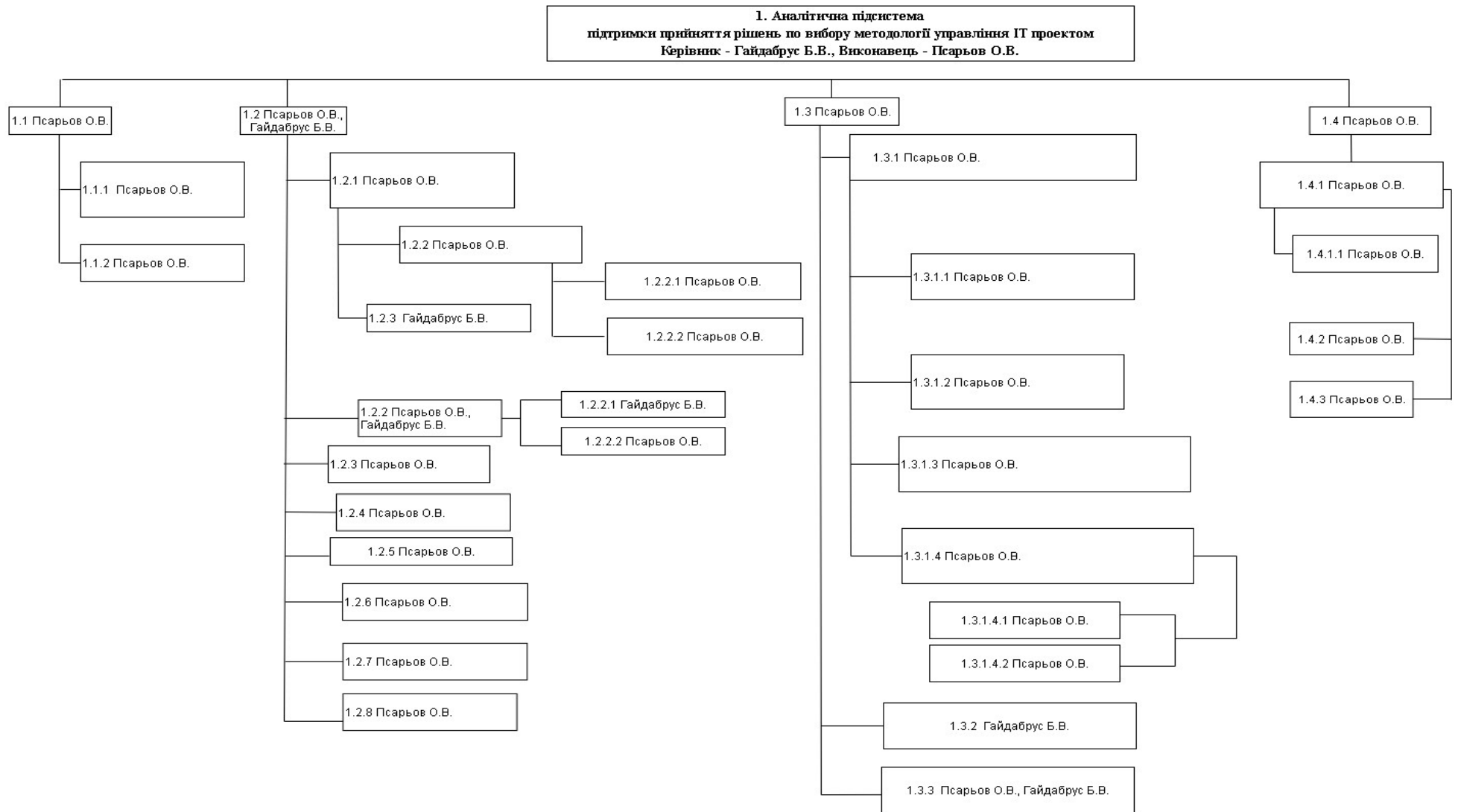


Рисунок А.2 – OBS структура

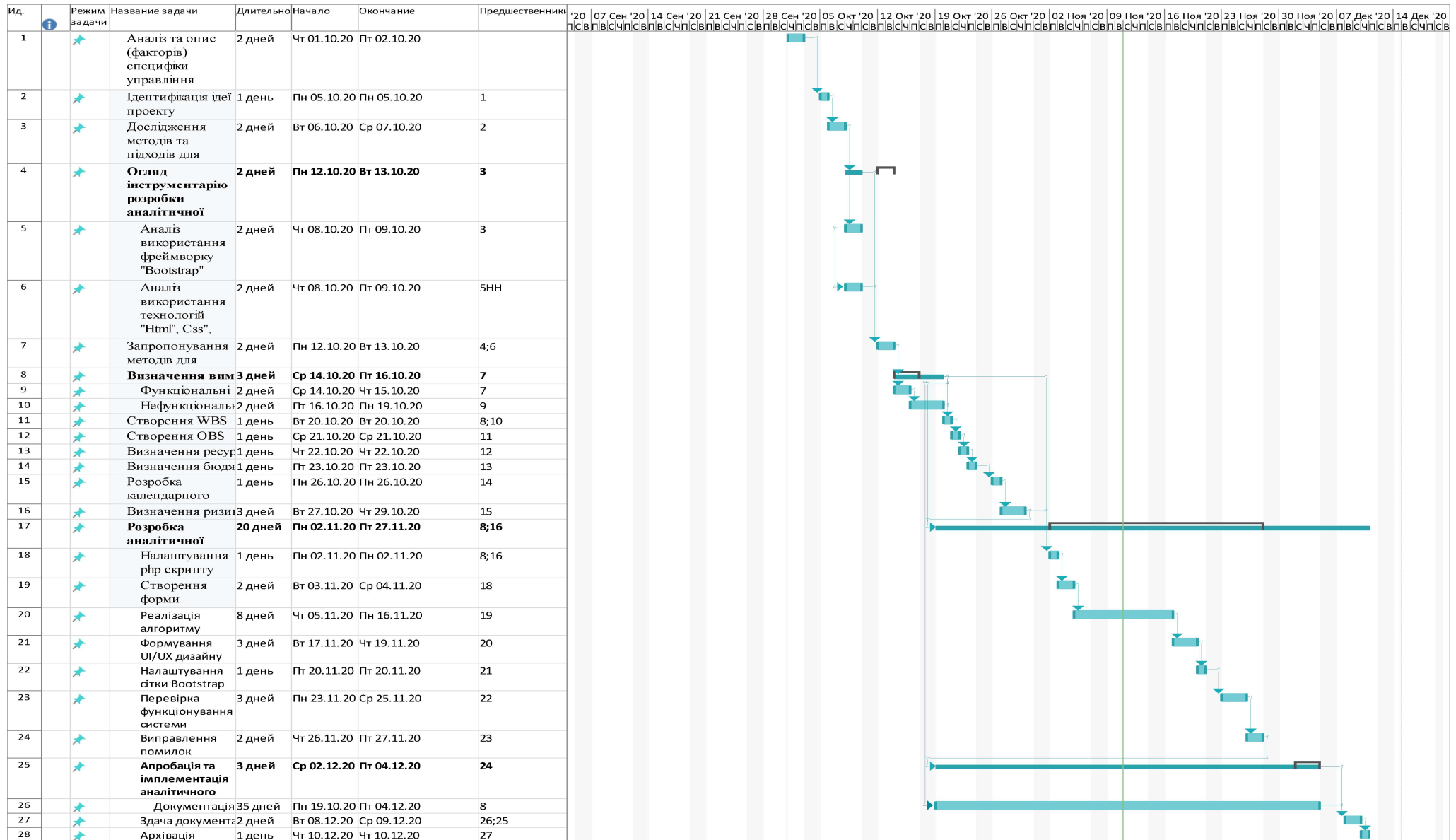


Рисунок А.3 – Діаграма Ганта


```

</div>
<div class="col-3">

</div>
</div>
</div>
<div class="row">
  <div class="col-12">
    <div class="container">
      <div class="row">
        <div class="col-12"><h6 class="text-light text-center" style="padding-top:5px">Powered by
Alexander_Psaryov<br>2020</h6>
      </div>
    </div>
  </div>
</div>
</div>
</nav></div>
</body>
</html>

```

Лістинг коду файлу index.php:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Підсистема аналітичного рішення</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div>
      <nav class="navbar bg-secondary fixed-top navbar-light">

```



```

}

}
?>
<?php
if(isset($_POST["btn2"])){
if(!empty($_POST['login1']) && !empty($_POST['password1'])) {
    $connect=mysqli_connect('localhost', 'id15315864_admin', 'Qwerty12345@', 'id15315864_methodologydb');
    $login1=mysqli_real_escape_string($connect,$_POST['login1']);
    $password1=mysqli_real_escape_string($connect,$_POST['password1']);
    $query=mysqli_query($connect,"SELECT * FROM `users` WHERE login='{$_login1}'");
    $numr=mysqli_num_rows($query);
    if($numr==0)
    {
    $sql_q="INSERT INTO `users`
(login,password)
VALUES('{$_login1}','{$_password1}')";
    $res=mysqli_query($connect,$sql_q);
    if($res){
    echo "Аккаунт успішно створений";
    }
    else {
    echo "Не вдалося додати інформацію";
    }
    }
    else {
    echo '<p class="formstyle">Цей нік зайнятий. Спробуйте інший!</p>';
    }
    }else {
    echo '<p class="formstyle">Всі поля обов'язкові для заповнення!</b>';
    }
    }
?>
</div>
</div>
<div class="col-2">

```

```

</div>
</div>
</div>
<nav class="navbar-light bg-secondary fixed-bottom">
  <div class="container ">
    <div class="row ">
      <div class="col-12">
        <h6 class="text-light text-center" style="padding-top:5px">Powered by Alexander_Psaryov<br>2020</h6>
      </div>
    </div>
  </div>
</nav></div>
</body>
</html>

```

Лістинг коду файлу test3.php:

```

<?php

error_reporting(0);

$question1 = "Ви вирішили отримати сертифікацію?";
$question2 = "Ви запускаєте проект / сервіс?";
$question3 = "Ви прагнете(шукаєте) отримати гнучку сертифікацію?";
$question4 = "Місцезнаходження вашого клієнту?";
$question5 = "Ви використовуєте Канбан?";
$question6 = "Ви працюєте на рівні портфелю?";
$question7 = "Ви працюєте на рівні програми?";

$question8 = "Ви працюєте на рівні проекту?";
$question9 = "Ви є членом команди Канбан?";
$question10 = "Ви управляєте командою Канбан?";
$question11 = "Ви тренуєте інші команди?";
$question12 = "Ваша команда > 25 осіб?";
$question13 = "Ви хочете відвідувати курси?";
$question14 = "Ви Скрам Майстер?";

```

```

$question15 = "Ви Власник Продукту?";
$question16 = "Ви Проектний менеджер?";
$question17 = "Ви Менеджер Продукту?";

if($_POST['q1'] == null){
    $check = 0;
} else {
    $check = 1;
}

switch($check){

    case 0:

        echo '<form action="home.php" method="POST"><div class="centerstyle"><br><br><br><br>';

        echo '<h3 class="display-3">'. $question1. '</h3>';
        echo '<p><label><input name="q1" type="radio" value="yes">&nbsp;Так</label>&nbsp;&nbsp;&nbsp;';
        echo '<label><input name="q1" type="radio" value="no">&nbsp;Ні</label></p><hr>';

        echo '<h3 class="display-3">'. $question2. '</h3>';
        echo '<p><label><input name="q2" type="radio" value="Проект">&nbsp;Проект</label>&nbsp;&nbsp;&nbsp;';
        echo '<label><input name="q2" type="radio" value="Сервіс">&nbsp;Сервіс</label></p><hr>';

        echo '<h3 class="display-3">'. $question3. '</h3>';
        echo '<p><label><input name="q3" type="radio" value="yes">&nbsp;Так</label>&nbsp;&nbsp;&nbsp;';
        echo '<label><input name="q3" type="radio" value="no">&nbsp;Ні</label></p><hr>';

        echo '<h3 class="display-3">'. $question4. '</h3>';
        echo '<p><label><input name="q4" type="radio" value="США">&nbsp;США</label>&nbsp;&nbsp;&nbsp;';
        echo '<label><input name="q4" type="radio" value="Об'єднане Королівство">&nbsp;Об'єднане
Королівство</label>&nbsp;&nbsp;&nbsp;';
        echo '<label><input name="q4" type="radio" value="Європа">&nbsp;Європа</label></p><hr>';

        echo '<h3 class="display-3">'. $question5. '</h3>';
        echo '<p><label><input name="q5" type="radio" value="yes">&nbsp;Так</label>&nbsp;&nbsp;&nbsp;';
        echo '<label><input name="q5" type="radio" value="no">&nbsp;Ні</label></p><hr>';

```



```

echo '<h3 class="display-3">'. $question16. '</h3>';
echo '<p><label><input name="q16" type="radio" value="yes">&nbsp;&nbsp;&nbsp;Так</label>&nbsp;&nbsp;&nbsp;';
echo '<label><input name="q16" type="radio" value="no">&nbsp;&nbsp;&nbsp;Hi</label></p><hr>';

echo '<h3 class="display-3">'. $question17. '</h3>';
echo '<p><label><input name="q17" type="radio" value="yes">&nbsp;&nbsp;&nbsp;Так</label>&nbsp;&nbsp;&nbsp;';
echo '<label><input name="q17" type="radio" value="no">&nbsp;&nbsp;&nbsp;Hi</label></p><hr>';
echo '<p><button type="submit" class="btn btn-success">Отримати
результат</button></p><br><br></div></form>';
break;

case 1:

if($_POST['q1'] == "yes"){
    if($_POST['q2'] == "Проект"){
        if($_POST['q3'] == "yes"){
            if($_POST['q5'] == "yes"){
                if($_POST['q9'] == "yes"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">TKP</h1></div>';

                }elseif($_POST['q9'] == "no"){
                    if($_POST['q10'] == "yes"){
                        echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">KMP ( I або II )</h1></div>';

                    }elseif($_POST['q10'] == "no"){
                        if($_POST['q11'] == "yes"){
                            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">AKT.KCP</h1></div>';

                        }elseif($_POST['q11'] == "no"){
                            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">AKT.KCP</h1></div>';

                        }

                    }

                }

            }elseif($_POST['q5'] == "no"){
                if($_POST['q12'] == "yes"){

```

```

echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">Scaled Professional Scrum Certification<br><br>Certified LeSS
Practioner- Principles to Practices<br><br>Certified SaFe</h1></div>';

```

```

}elseif($_POST['q12'] == "no"){
    if($_POST['q13'] == "yes"){
        if($_POST['q14'] == "yes"){
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">(A-)CSM</h1></div>';

```

```

        }elseif($_POST['q14'] == "no"){
            if($_POST['q15'] == "yes"){
                echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">(A-)CSPO</h1></div>';

```

```

            }elseif($_POST['q15'] == "no"){
                //-----
                echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">(A-)CSPO</h1></div>';

```

```

            }
        }
        if($_POST['q16'] == "yes"){
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">Delivery Management Track</h1></div>';

```

```

        }elseif($_POST['q16'] == "no"){
            if($_POST['q17'] == "yes"){
                echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">Product Management Track</h1></div>';

```

```

            }elseif($_POST['q17'] == "no"){
                echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">Product Management Track</h1></div>';
                //-----

```

```

            }
        }
    }elseif($_POST['q13'] == "no"){
        if($_POST['q14'] == "yes"){
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">PSM I -III</h1></div>';

```

```

    }elseif($_POST['q14'] == "no"){
        if($_POST['q15'] == "yes"){
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">PSPO I -III</h1></div>';

        }elseif($_POST['q15'] == "no"){
            //-----
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">PSPO I -III</h1></div>';

        }
    }
}
}
}
} elseif($_POST['q3'] == "no"){
    if($_POST['q4'] == "CIIIA"){
        if($_POST['q6'] == "yes"){
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">PfMP</h1></div>';

        }elseif($_POST['q6'] == "no"){
            if($_POST['q7'] == "yes"){
                echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">PgMP</h1></div>';

            }elseif($_POST['q7'] == "no"){
                if($_POST['q8'] == "yes"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">PMP</h1></div>';

                }elseif($_POST['q8'] == "no"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">CAMP</h1></div>';

                }
            }
        }
    } elseif($_POST['q4'] == "Обєднанє Королївство"){
        if($_POST['q6'] == "yes"){

```

```

        echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">P30</h1></div>';

    }elseif($_POST['q6'] == "no"){
        if($_POST['q7'] == "yes"){

            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">MSP</h1></div>';
            }elseif($_POST['q7'] == "no"){
                if($_POST['q8'] == "yes"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">PRINCE2 Practioner</h1></div>';

                }elseif($_POST['q8'] == "no"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">PRINCE2 Foundation</h1></div>';

                }
            }
        }
    }elseif($_POST['q4'] == "Европа"){
        if($_POST['q6'] == "yes"){
            echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">IPMA A</h1></div>';

        }elseif($_POST['q6'] == "no"){
            if($_POST['q7'] == "yes"){
                echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center;
height: 100vh"><h1 class="bg-primary text-white">IPMA B</h1></div>';

            }elseif($_POST['q7'] == "no"){
                if($_POST['q8'] == "yes"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">IPMA C</h1></div>';

                }elseif($_POST['q8'] == "no"){
                    echo '<div class="text-center" style="display: flex; justify-content: center; align-items:
center; height: 100vh"><h1 class="bg-primary text-white">IPMA D</h1></div>';

                }
            }
        }
    }
}
}
}

```

```

        }
    }
    } elseif($_POST['q2'] == "Сервіс"){
        echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center; height:
100vh"><h1 class="bg-primary text-white">ITIL</h1></div>';

        }
    } elseif($_POST['q1'] == "no") {

        echo '<div class="text-center" style="display: flex; justify-content: center; align-items: center; height:
100vh"><h1 class="bg-primary text-white">Немає потреби в цьому графіку</h1></div>';
        }

        break;
    }

?>

```

Лістинг коду файлу style.css:

```

body{
    background-image: url(background.png);
    background-size: cover;
    background-attachment: fixed;

}

h1{
    padding-left: 20px;
    padding-right: 20px;
    padding-top: 20px;
    padding-bottom: 20px;
}

.container{

```

```
}  
.display-3 {  
    font-size: 20px;  
}  
.display-4 {  
    font-size: 30px;  
}  
h2 {  
    font-size: 20px;  
}  
  
.centerstyle {  
    background-color: white;  
    padding-left: 30px;  
  
}  
.formstyle {  
    background-color: white;  
    margin-left: 40px;  
    margin-right: 40px;  
    margin-top: 20px;  
  
}  
  
hr {  
    display: block;  
    margin-top: 0.5em;  
    margin-bottom: 0.5em;  
    margin-left: auto;  
    margin-right: auto;  
    border-style: inset;  
    border-width: 1px;  
    color: black;  
  
}
```