

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації та інформаційних технологій

Управління проектами

(назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ

НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»

на тему: «Управління проектом розробки реляційних баз даних для
висококонкурентних додатків»

СОЛОДЕЙ НІКІТА ІВАНОВИЧ

(прізвище, ім'я та по батькові здобувача повністю)

Київ, 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: управління проектами

Освітній рівень: «магістр за ОПШ»

Спеціальність: 122. «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри УП

д.т.н., професор Бушуєв С.Д.

„___” _____ 2024 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

Солодей Нікіта Іванович

(прізвище, ім'я та по батькові студента)

1. Тема роботи: Управління проектом розробки реляційних баз даних для висококонкурентних додатків

затверджена наказом ректора КНУБА № 1665/2 від «20» 08 2024 р.

2. Керівник роботи: Веренич Олена Володимирівна, д.т.н., професор кафедри управління проектами

(прізвище, ім'я та по батькові, науковий ступінь, вчене звання)

3. Строк подання студентом роботи до захисту: листопад 2024 р.

4. Зміст пояснювальної записки за розділами:

Р. 1. Аналіз предметної області

Р. 2. Опис проблеми та шляхи вирішення

Р. 3. Управління проектом розробки баз даних

5. Інформаційні слайди

С. 1. Актуальність проведеної роботи

С. 2. Модель та архітектура баз даних

С. 3. Визначення поняття управління проектами

С. 4. Методології управління проектами

С. 5. Інструменти управління проектами

С. 6. Приклад створення та ведення проекту

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області	Вересень 2024 р.
Р. 2. Опис проблеми та шляхи вирішення	Жовтень 2024 р.
Р. 3. Управління проектом розробки баз даних	Листопад 2024 р.
Остаточне оформлення роботи	Листопад 2024 р.
Направлення роботи на рецензування, перевірку на плагіат	Листопад 2024 р.
Попередній захист роботи на кафедрі	Листопад 2024 р.

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Приєм програмного продукту			

8. Дата видачі завдання: 05 серпня 2024 року

Керівник

(підпис)

Веренич О.В.

(прізвище та ініціали)

Магістрант

(підпис)

Солодей Н.І.

(прізвище та ініціали)

РЕЗЮМЕ (summary)		Солодей Нікіта Іванович	
до кваліфікаційної роботи магістра здобувача:			
ЗВО	Київський національний університет будівництва і архітектури		
Тема	Управління проектом розробки реляційних баз даних для висококонкурентних додатків		
Освітній ступінь	Магістр за освітньо-професійною програмою навчання		
Факультет	Автоматизації і інформаційних технологій		
Кафедра	Управління проектами		
Спеціальність	122 “Комп’ютерні науки”		
Освітня програма	Управління проектами		
Керівник	Веренич Олена Володимирівна, д. т. н., професор		
Обсяг роботи:	пояснювальна записка, стор.	розділів	слайдів презентації
	85	3	15
Розділ 1.	Розглянуто складання списку робіт, тобто створення певного списку в якому буду зазначені всі роботи, які необхідно виконати в проекті; розподілення завдання, розподілення роботи між працівниками; складання загального плану, тобто план, що включає працівників, роботи, що необхідно виконати та терміни виконання; моніторинг діяльності персоналу, формування загального звіту включаючи інформацію про виконувані роботи, працівників та терміни виконання; моніторинг виконання робіт проекту, формування звіту по проекту включаючи список виконуваних робіт та терміни виконання, моделі та архітектуру баз даних.		
Розділ 2.	Запропоновано структуру реляційних баз даних, огляд методологій управління проектами, можливості інструментів для управління проектів, їх слабкі та сильні сторони. Розглянуто як методології та інструменти використовуються проектними менеджерами, а також для яких задач підходять наведені методології. Обумовлено вибір методологій та інструментів для проекту що розглядається відносно наданого технічного завдання та особливостей проекту.		
Розділ 3.	Запропоновано концептуальне проектування БД, даталогічне проектування БД, розробка фізичної моделі БД, обґрунтування вибору СУБД, діаграма бази даних, реалізація основних запитів до БД, підхід до постановки цілей проекту, аналіз ризиків, ключові етапи проекту: постановка цілей, створення команди проекту, створення матриці відповідальності, оцінка бюджету, ведення та завершення проекту. Наведено приклади використання інструментів для створення баз даних, плану проекту, звітності та організації проекту.		
Висновки по роботі:	Робота виконана на високому рівні, студент продемонстрував високий рівень теоретичної підготовки та сформованих		

	практичних навичок в області управління проєктами та сучасних інформаційних технологій.
--	---

Ключові слова: бази даних, управління проєктами, методології, реляційні бази даних, проєктний менеджер.

Студент: Нікіта Солодей

Керівник: Олена Веренич

« ____ » _____ 2024р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет автоматизації та інформаційних технологій

Управління проєктами

(назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри УП

д.т.н., професор Бушуєв С.Д.

„___” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

на тему: «Управління проєктом розробки реляційних баз даних для
висококонкурентних додатків»

Виконав: студент 2-го курсу, групи КН-УПм-23

Спеціальності: 122 «Компютерні науки»

ОПП: Управління проєктами

(шифр і назва спеціальності)

Солодей Н.І.

(прізвище та ініціали)

Керівник д.т.н., проф. Веренич О.В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ, 2024 р.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Визначення термінів баз даних.....	9
1.2 Архітектура баз даних.....	12
1.3 Визначення управління проєктами.....	15
1.4 Висновки.....	25
2 ОПИС ПРОБЛЕМИ ТА ШЛЯХИ ВИРІШЕННЯ.....	26
2.1 Реляційні бази даних.....	26
2.2 Методології управління проєктами.....	28
2.3 Інструменти для управління проєктами.....	45
2.4 Висновки.....	54
3 УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ БАЗ ДАНИХ.....	55
3.1 Розробка проєкту.....	55
3.1.1 Постановка цілей.....	55
3.1.2 Оцінка та аналіз ризиків.....	56
3.1.3 Створення команди проєкту.....	57
3.1.4 Розрахунок бюджету.....	64
3.1.5 Засоби контролю та комунікації.....	64
3.2 Ведення проєкту.....	64
3.2.1 Розробка бази даних.....	65
3.2.2 Контроль роботи та звітності.....	69
3.2.3 Запобігання ризикам.....	71
3.3 Завершення проєкту.....	72
3.3.1 Закриття контракту.....	72
3.3.2 Звільнення проєктної команди та оцінки після проєкту.....	73
3.4 Висновки.....	74
ЗАГАЛЬНІ ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	76
ДОДАТКИ.....	79

ВСТУП

Нині кожна організація, від малої до великої, та кожна людина реалізує нові ініціативи, які зазвичай є не що інше, як проекти. Істотна частка сучасного бізнесу в Україні, Європі та світі фактично орієнтована на проекти. В Україні вона наближається до позначки 50%, враховуючи всі сектори економіки. Це пов'язано з тим, що дедалі більше компаній зосереджуються на створенні значно модерніших продуктів або послуг для досягнення нових результатів у новітніх сферах.

Проекти – це вже не сукупність технічної документації та оцінок (як раніше), а вагомі заходи, плани, дії для досягнення конкретних цілей компанії. Тому успіх будь-якого підприємства залежить від уміння реалізувати їх у компанії. Тож управління проектами стає важливою та актуальною темою для менеджерів усіх рівнів [1].

В сучасних умовах нестабільності гостро постає питання створення та впровадження сучасних способів організації роботи підприємств. Головною вимогою до даних методів є гнучкість та готовність до змін, проте базуватись вони повинні на чітко сформованому плані дій. Найкраще в собі ці вимоги поєднує проектний менеджмент та набір різноманітних методологій управління проектами. Проте, методи та підходи є недостатньо часто задіяними у діяльності українських підприємств, а вибір одного зі способів управління проектами ускладнюється вибором одного серед кількох наявних.

Метою роботи є аналіз існуючих методологій управління проектами та дослідження використання сучасних методів управління проектів на прикладі проекту розробки реляційних баз даних для висококонкурентних додатків в сфері медицини.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення термінів баз даних

Характерною рисою сучасного суспільства є стрімке збільшення обсягів інформації, збільшення вимог до її точності та своєчасності. Традиційні способи збереження, пошуку та обробки інформації вже не задовольняють сучасним вимогам. Паперова технологія фактично вичерпала свої можливості щодо удосконалення методів роботи з інформацією. Для цього необхідні більш швидкі та ефективні методи.

Дані в інформаційній системі можуть зберігатися в неструктурованому або структурованому вигляді.

Неструктуровані дані – це звичайні текстові документи (статті, реферати, книги, тощо). Системи, в яких зберігаються неструктуровані дані, не завжди можуть дати конкретну відповідь на запит користувача, а можуть видати текст документи або перелік документів, в яких можна знайти відповідь. Наприклад, у відповідь на запит "Архітектура ПК" може бути виданий перелік статей, в яких згадується це поняття [2].

Під структурованими даними розуміються правила, що визначають їх форму, розмір, значення. Структуровані дані зазвичай зберігаються в базах даних (БД).

База даних являє собою комп'ютерний варіант організованої інформації. Зазвичай елементи інформації об'єднує спільна тема або призначення, як, наприклад, список студентів з оцінками в журналі групи.

Прикладами великих інформаційних систем є банківські системи, системи замовлень залізничних квитків, бібліотека, тощо.

Предметна область – це частина реального світу, яка вивчається з метою створення бази даних. Набори принципів, за якими організована структура зберігання даних в базі, називаються моделями даних.

Існують 3 основні моделі даних – реляційні бази даних, ієрархічні і мережеві структури.

Ієрархічна модель дозволяє будувати БД з деревовидної структурою. У них кожен вузол містить свій тип даних (сутність). На верхньому рівні дерева є один вузол – корінь, на наступному рівні розташовуються вузли, пов'язані з цим коренем, потім вузли, пов'язані з вузлами попереднього рівня, і т. д. Кожен вузол може мати тільки одного предка (рис. 1.1).

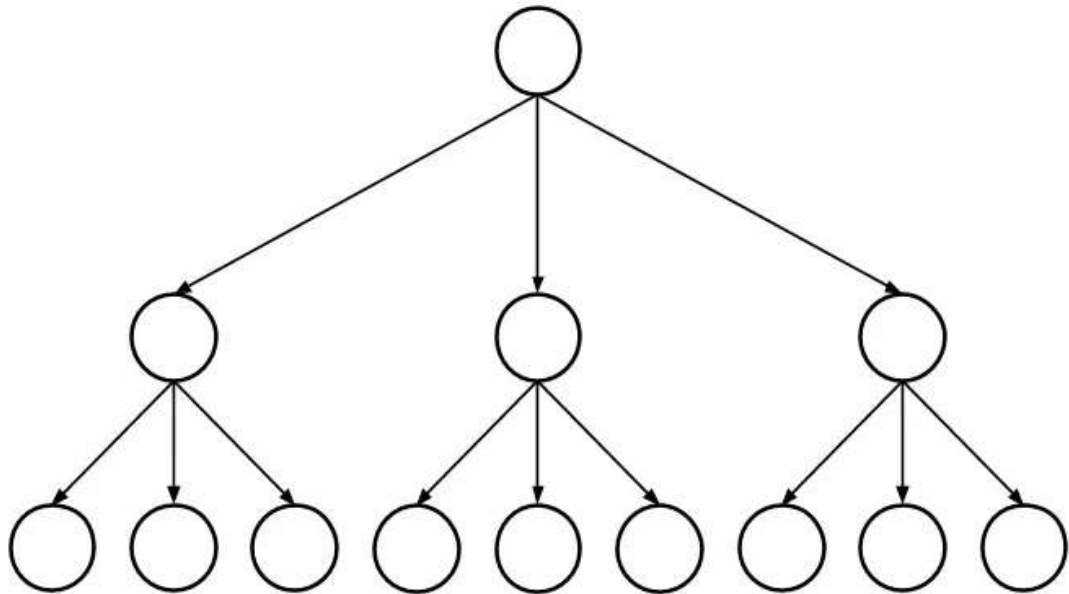


Рисунок 1.1 – Ієрархічна модель БД

Мережева модель є розширенням ієрархічної. В ієрархічних структурах запис-нащадок повинен мати тільки одного предка; в мережевій структурі даних нащадок може мати будь-яке число предків (рис 1.2).

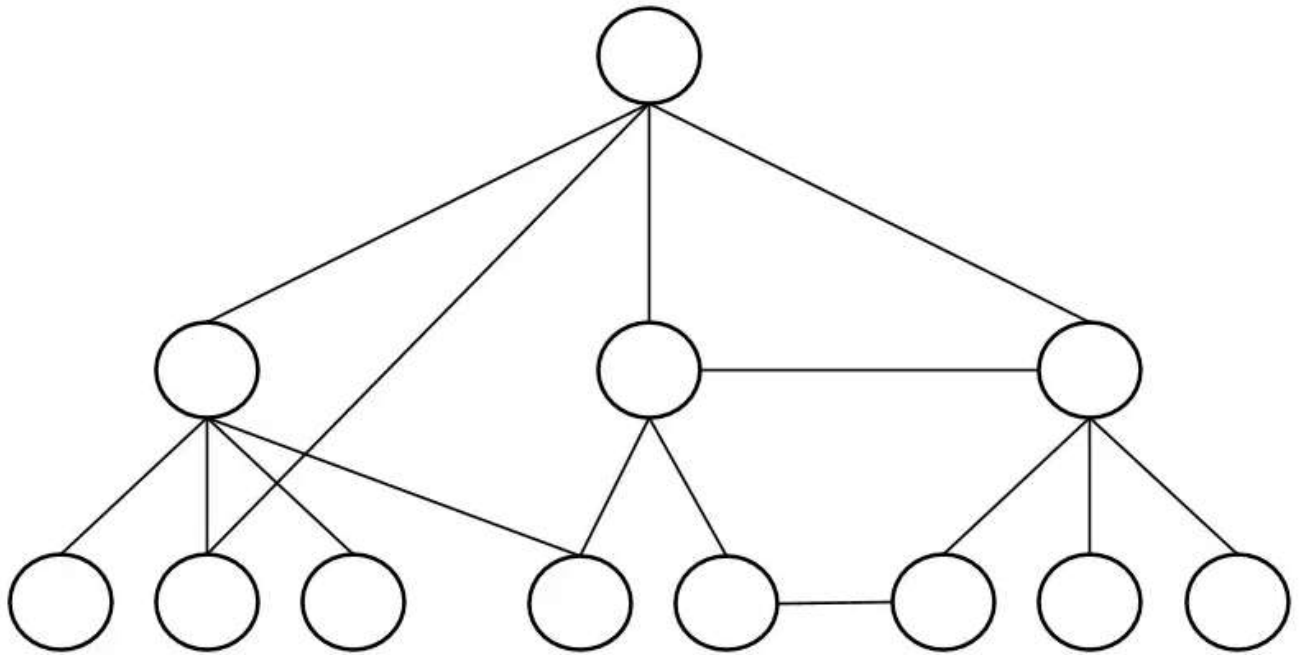


Рисунок 1.2 – Мережева модель БД

Реляційна модель даних являє собою сховище даних, які організовані у вигляді двовимірних таблиць. Будь таблиця реляційної бази даних складається з рядків, які називаються записами, і стовпців, які називаються полями. На перетині стовпця і рядка знаходяться конкретні значення даних, що містяться в таблиці.

База даних (БД) — це організована структура, призначена для зберігання, зміни й обробки взаємопов’язаної інформації, переважно великих обсягів. Бази даних активно використовують для динамічних сайтів зі значними обсягами даних — часто це інтернет-магазини, портали, корпоративні сайти. Такі сайти зазвичай розроблені за допомогою серверної мови програмування (наприклад, PHP) або на базі CMS (наприклад, WordPress), і не мають готових сторінок з даними за аналогією з HTML-сайтами. Сторінки динамічних сайтів формуються «на льоту» в результаті взаємодії скриптів і баз даних після відповідного запиту клієнта до веб-сервера [3].

В контексті баз даних варто розглянути поняття СКБД — система керування базами даних. СКБД — це комплекс програмних засобів, необхідних для створення структури нової бази, її наповнення, редагування вмісту і відображення інформації. Найбільш поширеними СКБД є MySQL, PostgreSQL, Oracle, Microsoft

SQL Server. Оренда віртуального хостингу від HOSTiQ.ua передбачає використання MariaDB — відгалуження СКБД MySQL, а також PostgreSQL. Якщо ж, наприклад, заплановано придбати VPS чи сервер в Європі або США, то можливість обрання ПО для баз даних залишається за клієнтом.

Це приклади СКБД типу клієнт-сервер, саме такі СКБД зустрічаються найчастіше у контексті поняття хостингу. Їх особливості:

- розташування СКБД на сервері з базами даних;
- безпосередній доступ до БД;
- централізована обробка клієнтських запитів на обробку даних;
- високий рівень надійності, доступності та безпеки;
- підвищене навантаження на сервер.

У свою чергу, для зручності роботи з СКБД використовують спеціальні веб-додатки, які дозволяють за допомогою графічного інтерфейса виконувати адміністрування сервера баз даних, запускати спеціальні команди, а також працювати з контентом таблиць і баз даних — дії, які при відсутності веб-додатка виконують за допомогою консолі. Приклади: phpMyAdmin використовують для адміністрування СКБД, а MySQL та pgAdmin — для PostgreSQL.

1.2 Архітектура баз даних

В наш час, в якості стандартної архітектури БД, використовується трьохрівнева система організації БД запропонована американським комітетом з стандартизації ANSI. Вона складається з зовнішньої моделі (схеми) даних, внутрішньої схеми даних (концептуальної моделі) та безпосередньо фізичної БД.

Перший рівень визначає точку зору на БД різних додатків користувачів. Кожен додаток бачить і обробляє тільки ті дані, які потрібні даному користувачеві.

Концептуальний рівень - центральна управляюча ланка. Дає представлення про базу даних у найбільш загальному вигляді. База даних представлена у

найбільш загальному вигляді, який об'єднує дані використовувані всіма додатками, що працюють з базою даних.

Фізичний рівень - безпосередньо дані розташовані в файлах або сторінкових структурах, що знаходяться на зовнішніх носіях інформації.

Ця архітектура дозволяє забезпечити логічну (між рівнями 1 і 2) та фізичну (між рівнями 2 і 3) незалежність при роботі з даними. Логічна незалежність передбачає можливість зміни одного додатка без коригування іншого. Фізична незалежність передбачає можливість перенесення інформації, що зберігається з одних носіїв на інші за умови збереження працездатності всіх додатків, що працюють з цією базою даних [4].

Розглядаючи сучасні реляційні СКБД можна виділити два класи - настільні та промислові, хоча ці класи мають однакове призначення: забезпечення коректного зберігання даних, надання доступу до даних та їх обробки. Настільні СКБД поставляються у складі офісних пакетів і призначені для вирішення задач збору і обробки інформації. До сучасних настільних СКБД відносяться: MS Access, Paradox, Visual dBASE, Visual FoxPro.

Сучасні промислові БД реалізовані на платформі клієнт-сервер, що дає можливість зберігати на сервері саме базу даних, а засоби представлення даних та алгоритми їх обробки (бізнес-логіку) перенести до клієнтської частини системи. Існують системи, в яких частину бізнес-логіки реалізовано вже на сервері БД, або на проміжному сервері, який спрощує структуру клієнтського додатку і залишає серверу БД традиційні функції збереження та підтримки цілісності даних. До клієнт-серверних СКБД відносяться широко відомі СКБД Oracl, MS-SQL-Server, Interbase, Informix та інші.

Розрізняють два види архітектури баз даних з розподіленим доступом в залежності від способу обробки даних: файл серверну та клієнт- серверну.

Файл-серверна архітектура централізованих БД з мережевим доступом передбачає функціонування одного з комп'ютерів мережі в якості виділеного сервера, на якому зберігаються файли централізованої БД (рис 1.3). Відповідно до запитів користувачів файли з файл-сервера передаються на робочі станції

користувачів, де здійснюється основна обробка даних. Централізований сервер виконує, в основному, лише роль сховища файлів, не приймаючи участі в обробці самих даних. Після завершення роботи користувачі копіюють файли з обробленими даними назад на сервер, звідки їх можуть копіювати і обробляти інші користувачі. Файл-серверна архітектура має ряд недоліків, а саме, при одночасному зверненні до одних тих самих даних користувачі змушені очікувати вивільнення даних, в результаті чого різко падає продуктивність роботи інформаційної системи. В протилежному випадку можливе взаємне затирання змін зроблених різними користувачами.



Рисунок 1.3 – Файл-серверна архітектура

В основу клієнт-серверної архітектури закладена ідея про те, що крім зберігання файлів бази даних, центральний сервер повинен виконувати і основну частину обробки даних (рис 1.4). Користувачі звертаються до центрального сервера за допомогою спеціальної мови структурованих записів - SQL, призначеної для програмування задач, які повинен виконати сервер. Звернення користувачів приймаються сервером і породжують на ньому процеси обробки

даних. В результаті користувач отримує уже оброблений набір даних. На відміну від технології "файл-сервер", між клієнтом і сервером передається не весь набір даних, а лише дані, які потрібні користувачеві. Технологія "клієнт-сервер" дає можливість уникнути передачі по мережі великих обсягів інформації завдяки перекладанню всієї обробки інформації на центральний сервер. Крім того, даний підхід дає можливість уникнути конфліктних ситуацій при зміні одних тих самих даних різними користувачами, що автоматично підтримує цілісність даних. Хоча слід відмітити, що в клієнт-серверних системах існують спеціальні засоби для підтримки цілісності даних. Недоліками клієнт-серверної архітектури є високі вимоги до продуктивності центрального сервера [5].

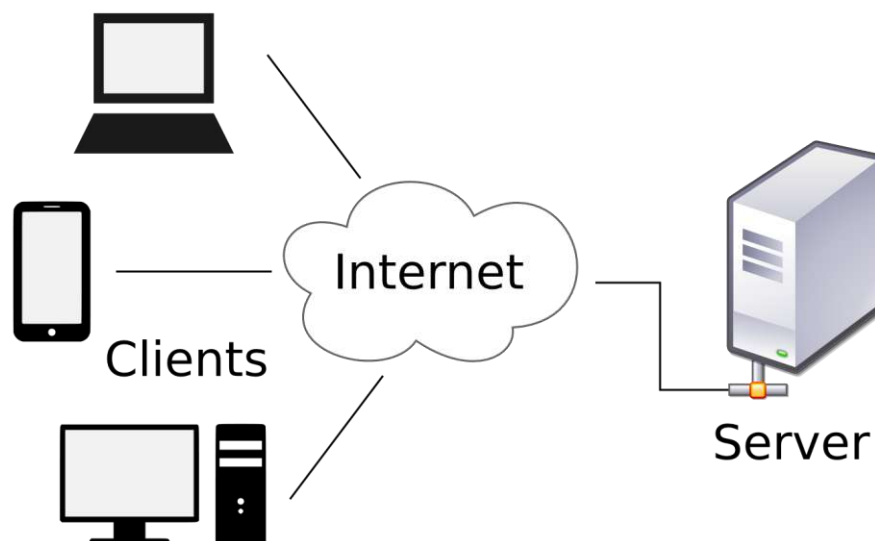


Рисунок 1.4 – Клієнт-серверна архітектура

1.3 Визначення управління проєктами

До появи офіційного визначення поняття «управління проєктами» люди керували підприємствами, будували заводи, будинки, пароплави, назагал цивілізація розвивалася доволі успішно і стрімко. Досі можна бачити результати деяких проєктів (скажімо, Єгипетські піраміди та іригаційні споруди, Велика Китайська стіна, тощо).

Сьогодні – це розробка засобів боротьби з пандемією вірусу COVID-19, проект з розвитку освітньої реформи в Україні, проект «Українська енергетика», проекти у сфері освіти, будівництво будинку, створення дослідницької програми, відновлення бізнесу, формування нової організації, розробка нових технологій та обладнання, будівництво кораблів, створення нового фільму і кінозйомка, розвиток регіону, організація найменшого свята, тощо. Усе вищеперелічене є проектами [6].

Нині вони є невід’ємною складовою реального життя будь-якого підприємства. У кожній компанії існує своя стратегія розвитку та цілі, які формуються в окремі проекти. Для менеджера важливо знати і розуміти, чим вони відрізняються від повсякденної роботи в організації.

По-перше, тим, що кожен проект має свою унікальну мету та обмеження у часі для її досягнення. У повсякденній діяльності мета повторюється, як і строки.

По-друге, проект закінчується, коли мети досягнуто. Повсякденні ж дії бізнесової діяльності мають потенційно безкінечний характер і їхня мета базується на отриманні прибутку шляхом підтримки бізнесової діяльності.

Існує чимало визначень проекту. І майже всі вони відповідають істині. Що б не робили: будівництво будинку, написання програми або організація дня народження – все це можна вважати проектом. Наведено декілька альтернативних тлумачень терміну «проект». Проект – це будь-що, що планується і включає завершальний етап.

Проект – це певне завдання зі встановленими цілями, досягнення яких визначає завершення проекту.

Проект – це сукупність взаємозалежних дій, спрямованих на досягнення поставленої мети у певний період та з чітко визначеними цілями.

Проект – це сукупність заходів для розроблення новітнього продукту або поліпшення вже існуючого продукту.

Проект – це конкретна подія або процес з чітко визначеними термінами, що спрямовані на створення унікального продукту чи досягнення інноваційних результатів.

Сучасна організація може існувати та успішно конкурувати на ринку, лише якщо вона постійно розвивається та адаптується до умов бізнесу. Це означає, що, плануючи конкретну мету, керівництво компанії постійно стикається з відповідними управлінськими проблемами: як планувати роботу, в потрібний час потрібні ресурси, скільки і коли саме, скільки це буде коштувати, коли буде досягнуто згоди з партнерами. Всі ці проблеми пов'язані з невизначеністю, яка зазвичай супроводжує реалізацію проектів.

Тож управління проектами – це процес управління людьми та координації людей, матеріальних і фінансових ресурсів у життєвому циклі проекту із застосуванням сучасних методів та прийомів управління, цілей проекту; це виконання комплексу взаємозалежних робіт в інтересах проекту [7].

Сучасні міждержавні проекти – надто дорога й відповідальна справа, аби доручити їх виконання маловідомим і недосвідченим менеджерам. І навіть якщо останні є знаними, то проводяться численні експертизи, щоб пересвідчитись у якості проекту.

Видається, що в цілому світі роль менеджера проекту не надто правильно розуміють. Позаяк чимало менеджерів проектів потрапили на цю посаду внаслідок підвищення посади інженера, програміста, дослідника тощо. Та попри те що і вони, і їхні керівники вважають цю роботу суто технічною, однак насправді така думка є хибною.

Суть механізму управління проектами полягає в управлінні цілями організації, що допомагає компанії швидко досягати успіху в конкурентній боротьбі, реагувати на зовнішні та внутрішні зміни, економити час і гроші (рис 1.5). З цією метою керівник постійно контролює три фактори, які через тісну взаємодію можна об'єднати в множину складових проектного менеджера: час, зміст і строки роботи [8].



Рисунок 1.5 – Механізм управління проектом

Управління проектами допомагає швидко та ефективно домогтися поставленої мети. Крім того, в процесі цього створюється ціла низка інструментів, які можна використовувати для досягнення загальних цілей компанії, а також під час розробки схеми розумного розподілу ресурсів. Нерідко успішний проект може започаткувати регулярну діяльність у якійсь галузі. У цьому аспекті можна назвати проекти «зачинателями» бізнесу. З часом у фундаторів такого роду проектів з'являються послідовники, інші учасники ринку намагаються копіювати успішні методики, прийоми та товари і послуги.

Нині управління проектами – це ціла наука, огорнута системою знань, правил і стандартів. Найвідоміше джерело – РМВОК (довідник з управління проектами). Це своєрідна «книга знань» Основне завдання менеджера проекту –

управляти проектом, а не виконувати технічну роботу у галузі управління проектами, де описано найкращі практичні поради й тематичні знання.

Істотною передумовою управління проектами є точне визначення та формулювання цілей, починаючи з найвищого рівня, поступово до найбільш конкретних цілей і завдань. Крім того, управління проектами доречніше розглядати не як послідовну реалізацію ретельно підібраних цілей, а як просування проекту вперед. Такий механізм пов'язаний з досягненнями цілей від управлінського рівня до досягнення кінцевої мети.

Проект у компанії виконують не лише для його реалізації. Тобто виконання не є самоціль проекту, і нерідко не заради успішного вихідного продукту. Проект реалізується з метою досягнення Головного обов'язку менеджера проекту – забезпечити, щоб усі роботи було завершено вчасно в межах бюджету та змісту з належними результатами конкретних цілей, поставлених перед компанією. Визначення цілей проекту починається з визначення його місії, зазвичай допитливий менеджер конкретизує проблему й пропонує проект як засіб її розв'язання [9].

Місію проекту слід розуміти як основні цілі, для яких проект розроблений і повинен бути досягнутий.

Місія визначає основне бачення проекту та його концепцію. Можна місію розглядати як «щось грандіозне та цінне», що проект принесе суспільству. Це має викликати позитивні емоції у зацікавлених сторін та учасників проекту. Місія також мусить бути доброю та придатною для майбутнього.

Найпоширенішими сферами діяльності (орієнтованими на проекти) є будівництво, автомобільна галузь, фармація, архітектура, охорона здоров'я, розробка комп'ютерного програмного забезпечення тощо. Крім проектів у традиційному розумінні, можна говорити про здійснення соціального проекту, наприклад пенсійної реформи, та політичного – це можуть бути парламентські вибори, або побутові проекти, скажімо, сімейна відпустка чи тривала подорож.

Прикладом масштабного соціально значущого проекту може бути запровадження електронного квитка у межах міста, обладнання сільських шкіл

району спортивним приладдям чи комп'ютерною технікою, різноманітні екологічні проекти.

Після того як визначено місію проєкту, менеджер може рухатися вперед із його цілями. Вони набагато конкретніші, ніж сама місія, і визначають результати, яких потрібно досягти задля виконання загальної місії. Також вони визначають бажаний кінцевий результат.

Правильна постановка цілей – це радше мистецтво, аніж наука. Незрозумілий текст, відсутність часових рамок та критеріїв, яким слід відповідати, – найпоширеніші помилки у формулюванні цілей проєкту. Для запобігання цим та іншим помилкам при правильному плануванні проєкту застосовують SMART-аналіз (рис. 1.6).



Рисунок 1.6 – SMART-цілі

Система встановлення цілей SMART дає можливість узагальнити всю наявну інформацію на початковому етапі проєктної діяльності, встановити прийнятні терміни, визначити достатність ресурсів та надати всім учасникам процесу чіткі, точні та конкретні завдання. Назва аналізу базується на англійських термінах, що формують цю аббревіатуру і мають конкретне смислове навантаження:

- Specific, конкретність. Характеризується чітко визначеними цілями, що збільшує ймовірність досягнення мети.

- Measurable, вимірюваність. Обумовлюється наявними критеріями або метриками для вимірювання процесу досягнення цілей, а саме такими показниками, як якість, кількість та ціна.

- Achievable, досяжність. Розумні цілі повинні бути досяжними, оскільки вони впливають на реальність завдання для мотивації підрядника. Якщо мета недосяжна – ймовірність її досягнення дорівнюватиме нулю.

- Relevant, актуальність. Щоб визначити актуальність мети, важливо зрозуміти, як вирішення конкретного завдання сприятиме досягненню глобальних стратегічних цілей компанії. При встановленні поточної мети виникає питання: які переваги матиме компанія? Якщо суспільство загалом не виграє від цієї мети – ця ціль вважається непотрібною і марною тратою ресурсів суспільства.

- Time-bound, обмеженість у часі. Ціль для SMART повинна бути обмежена в часі, що встановлює кінцевий термін, перевищення якого вказує на те, що мети не досягнуто; у протилежному випадку марно й говорити про досяжність – це може засвідчувати, що певна мета теоретично досяжна, проте неможливо вказати часові рамки, у які її вдасться досягнути. Встановлення часових рамок та меж реалізації мети допомагає керувати процесом управління. У такому разі часові рамки мають установлюватися з урахуванням своєчасного досягнення цілі.

Для встановлення цілей та моніторингу процесу їх досягнення важливо звернути увагу на очікуваний результат і те, чого досягли від результату. Це допоможе зосередитися на тому, чого менеджер намагається досягти, а не на витрачених зусиллях.

Кожен проєкт, незалежно від складності та обсягу робіт, необхідних для його реалізації, залежить від певних умов його розробки та реалізації.

Організаційне управління проєктами (ОПМ) – це основа, яка використовується для узгодження практик управління проєктами, програмами та портфелем з організаційною стратегією і цілями, а також налаштування або пристосування цих практик до контексту, ситуації чи структури організації. Стандарт ОПМ надає керівництво організаційним управлінням, працівникам ОУП

та практикам щодо цих тем. Він охоплює процес визначення вартості і може бути використаний з усіма підходами до реалізації проєкту.

Організаційне управління проєктами допомагає організаціям набувати ваги за допомогою таких принципів:

- узгодження з організаційною стратегією;
- інтеграція з організаційними можливостями;
- послідовність навчання;
- організаційна інтеграція;
- ставлення до організації;
- постійний розвиток.

Хоча стандарт ОРМ корисний для будь-якої організації, яка прагне щонайкраще відповідати своїм стратегічним цілям, він є особливо корисним для організацій, які не мають єдиного підходу до управління проєктами, і тих, хто перебуває в процесі вдосконалення чи утримання своєї поточної системи управління проєктами. В міру того як ваша організація зростає зі зміною часу та адаптується до руйнівних технологій, цей стандарт допомагає менеджеру підтримувати стабільну структуру, щоби не відставати від своєї організаційної стратегії.

Керуючись тенденціями управління проєктами, ви завжди можете бути одним із перших, хто застосував останні інновації. Ці нововведення, як правило, допомагають працювати ефективніше, поліпшувати спілкування та тішити співробітників. Пропонуємо декілька тенденцій управління проєктами, які починають з'являтися і, найімовірніше, матимуть великий вплив незабаром. 1. Сфера, на яку слід звернути особливу увагу, – це безпека даних та конфіденційність. Більша частина нашого часу в системі управління проєктами зараз відбувається в Інтернеті, і хоча це має масу переваг, також це містить певні недоліки. Щоб захистити свій бізнес, потрібно вивчити систему управління інформаційною безпекою (ISMS – СУІБ), або Управління інформаційною безпекою, – це програмне рішення, яке містить набір засобів контролю,

призначених для захисту активів управління проектами. Перш ніж вибрати своє рішення ISMS, варто ознайомитись із найновішими стандартами, які вимагають ISMS. Потрібно дізнатися, що таке ISO 27001, щоби краще зрозуміти, як це все працює. За результатами ознайомлення із програмним забезпеченням СУІБ та протоколами, яких вони мають дотримуватись, можна захистити свої активи управління проектами в майбутньому.

Управління проектами віддаленої роботи – особливо актуальне за карантинних обмежень періоду пандемії. Більше підприємств, ніж будь-коли, мають свої команди, що працюють вдома. Частково це пов'язано з нещодавною кризою в галузі охорони здоров'я, але також із тим, що підприємства усвідомлюють вигоди від дистанційної роботи. Надання дозволу командам працювати вдома може заощадити бізнесу великі кошти на офісних приміщеннях, однак може і створити деякі проблеми для керівників проєктів. Дбаючи за свою команду, менеджери проєктів повинні вибрати альтернативний спосіб управління підлеглими задля досягнення максимальної ефективності. З огляду на це, керівники проєктів мають навчитися застосовувати гібридні моделі віддаленої роботи. Ці моделі передбачають найкраще програмне забезпечення для управління командами як особисто, так і вдома, крім стратегій для поліпшення комунікації та ефективності. Оскільки найближчими роками дедалі більше підприємств покладатимуться на віддалених працівників, зараз ідеальний час для керівників проєктів вивчити провідні практики [10].

Автоматизація – це ще одна сфера, на яку доцільно звернути увагу керівникам проєктів. Завдяки автоматизації менеджери проєктів можуть не тільки заощадити багато часу, а й краще обмірковувати свої проєкти. Нові засоби штучного інтелекту дають можливості вирішувати багато простих, але трудомістких завдань, з якими стикаються чимало менеджерів проєктів. Наприклад, інструменти штучного інтелекту можуть впоратися з плануванням зустрічей та термінів і допомогти зібрати важливі дані. Менеджерам проєктів, зацікавленим у використанні штучного інтелекту та автоматизації, слід почати з вивчення того, які їхні повсякденні завдання займають найбільше часу. Далі треба

шукати сучасні інструменти управління проектами, які здатні вирішити ці завдання. Нині вже існує програмне забезпечення, яке може зменшити ваше щоденне навантаження, його треба тільки віднайти і грамотно підібрати.

Головною тенденцією сьогодення є те, що менеджери проектів мало зосереджуються на особистості та емоціях членів своєї команди. І це вагомий недолік. Найкращий підхід – розглянути кожного окремого члена команди і віднайти способи максимізувати їхню ефективність. Припустимо, є учасники команди, які не працюють. Після проведення співбесіди із ними з'ясується, що у них є певні особисті проблеми вдома, і якщо дозволити їм працювати вдома – це значно зменшить їхній стрес. Кожен індивід – це неповторна особистість зі своїми емоціями. Керівники проектів повинні дізнатись більше про свою команду та скоригувати свій стиль управління відповідно до тих, хто працює на них. Це може підвищити продуктивність праці та виявити індивідуальність співробітників, одночасно зменшивши обсяги витрат.

Складно зрозуміти, як поліпшити свою роботу з управління проектами, не маючи надійної бази даних для аналізу. Менеджери проектів повинні мати систему для збору якомога більшої кількості даних про свої проекти та команди. Потім їм слід виробити звичку аналізувати цю інформацію та шукати шляхи поліпшення ситуації. Застосовуючи аналітику даних, менеджер може з'ясувати певні аспекти:

- які завдання займають найбільше часу;
- які члени команди виконують найшвидше роботу;
- які члени команди регулярно пропускають дедлайни, чому;
- які сфери вашого розвитку мають проблемні місця;
- чи перевищуєте ви бюджет, чи виявляєте нестачі;
- наскільки правильне використання ресурсів;
- які прогнози доходів.

Тенденції управління проектами, наведені вище, є гарним стартом для зародження ідеї проекту. Але за невеликий час з'являться нові переконання

споживачів, нові потреби ринку, нові методи, цілі і завдання. Щоб отримати конкурентну перевагу, менеджеру слід не тільки регулярно шукати нові тенденції управління проектами, а й бути готовими випробовувати їх у власному бізнесі. Деякі з них працюватимуть краще за інших, деякі слабше. Але мета полягає в тому, щоби створити атмосферу інновацій та адаптивності. Якщо можна це зробити – то необхідно переконатися, що обрані методи управління проектами продовжують вдосконалюватися з часом і що бізнес отримує від цього велику користь [11].

1.4 Висновки

В даному розділі досліджено особливості моделей та архітектури баз даних. Відповідно до технічного завдання проекту обрано клієнт-серверну модель, що була затверджена організацією та замовником проекту. Серед архітектур баз даних була обрана реляційна, як найбільш поширена в сфері медицини завдяки можливості пов'язувати між собою значну кількість таблиць з даними.

Розглянуто основні принципи та етапи управління проектом, визначено необхідні для проектного менеджера компетенції та знання, що сприятимуть покращенню процесу управління проектами.

Визначено основні тенденції управління проектом, на основі яких буде сформований проект управління розробкою реляційних баз даних для висококонкурентних додатків.

2. ОПИС ПРОБЛЕМИ ТА ШЛЯХИ ВИРІШЕННЯ

2.1 Реляційні бази даних

Реляційні бази даних (РБД) – один із найпоширеніших типів баз даних, що використовуються для зберігання, організації та управління інформацією. Засновані на реляційній моделі даних, яку було запропоновано Едгаром Коддом у 1970 році.

Уявимо, що у комп'ютері кожен додаток має свої власні, ізольовані файли для зберігання даних. Вони ніяким чином не взаємодіють. Щоб використовувати дані з різних застосунків, потрібно відкрити щонайменше два різні файли, скопіювати потрібні дані і зберегти в третьому файлі. І так до нескінченності (рис. 2.1).

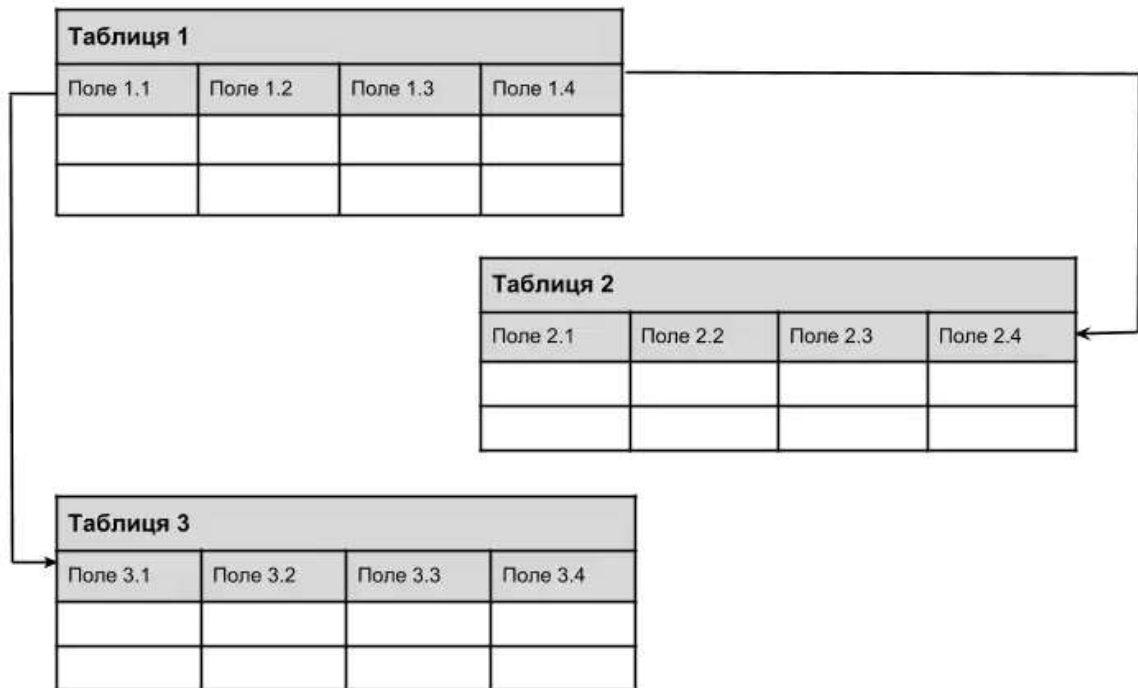


Рисунок 2.1 – Схема реляційної моделі даних

Звичайно, щойно з'явилася технічна можливість, одразу ж було винайдено рішення: зберігати дані в таблицях, пов'язаних між собою. Нагадуємо, що *relation*, за англійської мови – це зв'язок, або відношення. Тому таблиці з даними називають реляційними.

Доступ до реляційних таблиць здійснюється за допомогою мови структурованих запитів (SQL). Реляційні бази даних – це гнучкий і структурований спосіб керування даними, який забезпечує вищий рівень абстракції, що дає змогу розробникам ефективніше працювати з даними.

Уявимо таблицю під назвою “Співробітники”, де кожен рядок представляє інформацію про конкретного співробітника, а стовпці можуть містити його ім’я, прізвище, посаду, зарплату тощо. Кожен рядок – це кортеж (запис), а кожен стовпець – атрибут.

Швидкий доступ і управління даними реалізуються за допомогою SQL-запитів. Це робить реляційні бази даних ідеальними для широкого спектра додатків, від обліку та управління замовленнями до зберігання інформації про клієнтів і багато чого іншого.

Реляційна база даних включає в себе наступні елементи необхідні для її роботи.

- **Реляційні таблиці.** Данні організуються у вигляді таблиць із рядками та стовпцями. Кожна таблиця має ім’я та структуру, визначену схемою даних.
- **Рядки (кортежі).** Кожен рядок у реляційній таблиці представляє окремий запис і містить інформацію про об’єкт. Кожен запис є унікальним та ідентифікується за допомогою ключа.
- **Стовпці (атрибути).** Стовпці таблиці являють собою атрибути або характеристики об’єктів, що описуються в таблиці. Кожен стовпець має ім’я і тип даних, який визначає, якого роду інформацію можна зберігати в цьому стовпчику (наприклад, текст, числа, дати і так далі).
- **Ключі.** У реляційній моделі використовуються ключі для унікальної ідентифікації рядків у таблиці. Основний ключ (Primary Key) забезпечує унікальність кожного рядка, а зовнішній ключ (Foreign Key) створює зв’язки між таблицями.
- **Зв’язки.** Реляційні бази даних дають змогу встановлювати зв’язки між таблицями, що дає змогу об’єднувати дані з різних таблиць для виконання складних запитів.

- **SQL (Structured Query Language).** SQL використовується для виконання операцій з даними в реляційних базах даних. Він дає змогу створювати, змінювати, видаляти та витягувати дані з таблиць, а також визначати правила для цілісності даних.
- **Нормалізація.** Реляційні бази даних використовують процес нормалізації для зменшення надмірності даних і забезпечення їхньої цілісності.
- **Транзакції.** РБД підтримують транзакції, що забезпечує атомарність, узгодженість, ізолюваність і довговічність даних (ACID-властивості) під час одночасного доступу кількох користувачів до бази даних.

Нормалізація даних – це процес організації даних у реляційній БД з метою мінімізації надлишковості даних, поліпшення цілісності даних і забезпечення ефективності виконання запитів.

Якщо говорити простішою мовою, нормалізація дає змогу розбити великі таблиці на дрібніші, щоб уникнути повторення інформації та зменшити ймовірність помилок.

2.2 Методології управління проектами

Методологія в управлінні проектом — це набір певних правил, умов, принципів, дій, яких необхідно дотримуватись задля якісного ведення проекту, від його ініціалізації до закриття. Свого роду це стандартизація підходу до роботи. Наприклад технологія може визначати з яких кроків складаються робочі процеси, як контролюється виконання задач та відбувається взаємодія між членами команди, приймаються рішення, включати інші стандарти ведення проекту.

Використання вибраної методології в управлінні проектами дозволяє:

- Мати спільне для усієї команди розуміння процесів, обов'язків, ролей — робота за методологією (будь-якою) передбачає прозорість в колективі;
- Точніше визначати ресурси та прогнозувати кінцевий результат — на старті ведення проекту в межах методології визначаються обсяги робіт, необхідних для виконання, щоб визначити фаховість співробітників,

кількість коштів та часу для упровадження. Навіть якщо застосовується фреймворк із Agile-філософії Scrum, все одно є певні обмеження та орієнтири;

- Підвищити продуктивність співробітників — визначення компетенцій та потреб членів команди проекту, підвищення мотивації;
- Швидше та краще закривати робочі питання — знову ж таки завдяки чіткому розподіленню ролей і компетенцій кожного залученого до проекту, розумінню задач, цілей, необхідних кроків для їх досягнення. Дефіциту інформації немає в жодного з членів команди — від trainee-тестувальника до провідного UI/UX-дизайнера;
- Зростати професійно — зокрема, проєктному менеджеру. Коли команда працює в рамках методології, процеси налагоджені і є чітка послідовність дій, координація та контроль співробітників проходять легше. Це дозволяє цілком зосередитись на успішній реалізації продукту. Крім того, робота за методологією дає РМ-у цілісне розуміння проєкту, що дозволяє глибше «бачити» і аналізувати всі процеси;
- Покращити загальну атмосферу в команді та згуртувати співробітників — в роботі за методологією відсутня невизначеність. Кожен співробітник чітко знає хто чим займається та за що відповідальний. Це сприяє зменшенню кількості конфліктних ситуацій в колективі;
- Не повторювати помилок, яких команда припускалась в минулих проєктах — використання методології дозволяє сформувати своєрідну культуру успішного проєктного управління — прописуються практики, що дають кращий результат та дії, яких слід уникати.

Хоч найчастіше робота за певним підходом практикується в ІТ-компаніях, методологія актуальна всюди, де є проєкт та команда: від розробки мобільних застосунків або формування маркетингових стратегій просування для компаній-клієнтів, до автомобільної промисловості чи будівельної галузі.

Методологія, метод, методика — ці три терміни дуже часто ототожнюють, помилково приймаючи їх за синоніми. В результаті виникає плутанина, через яку новачкам-«проджекам» складніше розібратися в теорії проектного управління.

Метод — набір прийомів для здійснення якоїсь діяльності, для тобто техніка виконання роботи.

Методологія — набір правил, принципів і цінностей, підкріплених теорією, які застосовуються для рішення проектних завдань. Її також називають моделлю.

Методика (або фреймворк) — готовий алгоритм використання різних технік для досягнення певної мети.

Тож розбирати можна не лише методології, а й методи та методики — найбільш популярні та часто застосовувані в проектному управлінні [12].

Каскадна модель Waterfall або Водоспад є однією з найбільш традиційних і поширених моделей в управлінні проектами. Її основа — це послідовність. Тобто кожен етап проекту «перетікає» в наступний, неухильно стікаючи вниз, як водоспад. Зазвичай Waterfall застосовується в проектах, які можуть бути поділені на послідовні логічні частини. Наприклад кроки в ІТ-проекті за моделлю Waterfall виглядатимуть наступним чином:

- визначення вимог проекту та їх аналіз — проектний менеджер збирає вимоги до майбутнього продукту, планує графік виконання робіт та оцінює можливі ризики;
- проектування — на цьому етапі готуються документи, в яких детально описується план реалізації попередньо сформованих вимог; команда створює прототип та дизайн-макети майбутнього продукту;
- реалізація — втілення готового проекту згідно із затвердженими планом, макетами та вимогами. Розробники працюють чітко за технічним завданням;
- тестування — реалізований продукт тестують та усувають помилки і недоліки;
- впровадження — продукт офіційно запускають в роботу;

- підтримка — надається технічна допомога після запуску комерційної експлуатації.

За Waterfall кожен крок обов'язково йде слідом за попереднім — завершено один, команда переходить до наступного (рис. 2.2). При цьому пропуск окремого етапу чи повернення до вже закінченого (наприклад, з метою внесення доопрацювань) моделлю не передбачені. Всі етапи обов'язково документуються — якщо в процесі розробки вимоги до функціоналу змінились, вони завжди фіксуються в технічному завданні. При цьому клієнта (замовника) до роботи не залучають. Результат проєкту він бачить вже на фінальному етапі.

Через логічність та структурованість, Waterfall — оптимальна відправна точка для вивчення методологій управління проєктами. Вона найпростіша і у використанні — проєктний менеджер складає план, розробляє структуру, а далі його головне завдання — слідкувати, щоб цього плану дотримувалась команда чітко і у строк. Серед інших переваг Waterfall — стабільність завдань, реальна оцінка вартості та дати впровадження продукту.

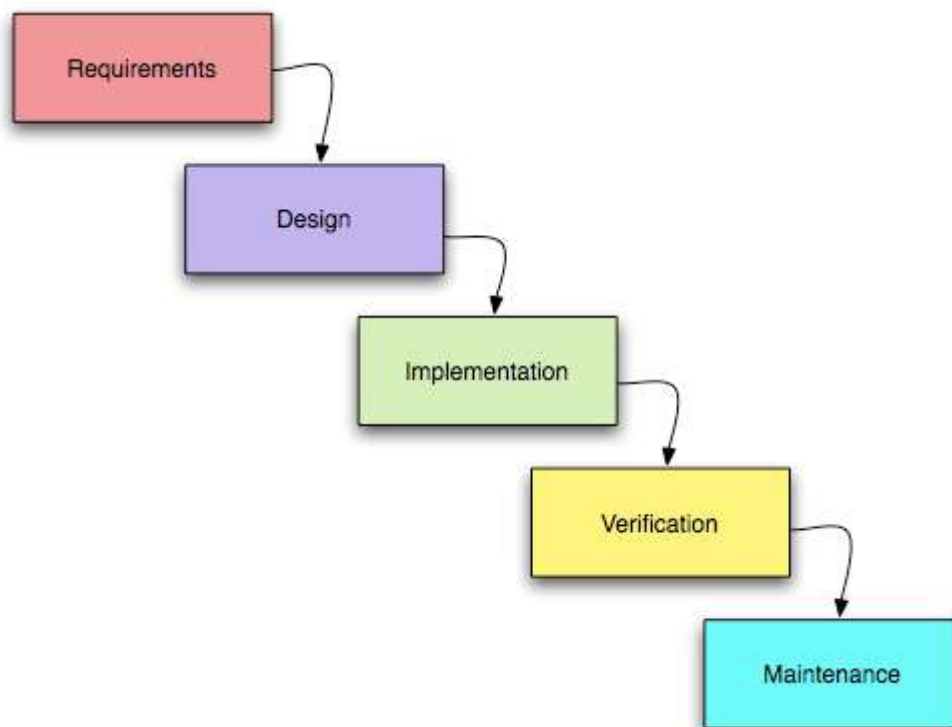


Рисунок 2.2 – Модель Waterfall

Хід проєкту легко відстежувати та контролювати. До того ж, зважаючи на те, що модель Waterfall в значній мірі спирається на документацію та записи в процесі розробки, це спрощує залучення нових учасників у проєкт. Наприклад, якщо співробітник звільнився, новачок зможе продовжити роботу майже миттєво — завдяки чіткій регламентації та наданим технічним матеріалам.

Разом з тим, через недостатню гнучкість для змін та переглядів етапів, ця модель не підійде проєктам, де вимоги можуть змінитися. Так, якщо це співпраця з клієнтом, замовник повинен точно знати, що він хоче отримати. В іншому випадку є високий рівень ризику отримати або не той результат, що знецінить весь робочий процес, або ж доведеться починати планування з нуля, а це значно відтермінує задачу проєкту.

Waterfall підійде:

- Великим проєктам, де фігурує багато стейкхолдерів. Ця модель передбачає максимум деталізації та наявність розгорнутої документації про необхідні дії протягом усього проєкту, полегшує відстеження роботи, яку слід виконати для досягнення цілей.
- Проєктам з дуже суворими вимогами до бюджетів та термінів.
- Коротким, простим проєктам з невеликим бюджетом, де обсяг робіт можна легко визначити та сформулювати в ТЗ.

Модель Waterfall показує себе найкраще при розробці фізичних об'єктів (обладнання, будівлі тощо) — коли конкретне завдання має призводити до конкретного кінцевого результату. Також ця модель полегшить майбутню реалізацію схожих проєктів: наприклад будівництво типових будинків. А ось в сучасній ІТ-розробці, де вимоги змінюються регулярно, а оновлення необхідно випускати якнайчастіше, Waterfall застосовується доволі рідко.

Слово Agile перекладається з англійської як «гнучкий», «спритний», «верткий». Саме гнучкість і покладено в основу цієї філософії. Загалом Agile є своєрідною «відповіддю» на обмеження та жорсткий регламент Waterfall. Офіційно її було представлено у 2001 році в «Agile Manifesto», створеному 17

провідними розробниками спеціально для ІТ-галузі. Згідно з маніфестом, підхід Agile ґрунтується на чотирьох цінностях:

- Люди та взаємодія з ними важливіші за процеси та інструменти.
- Працюючий продукт (результат проєкту) важливіший за вичерпну документацію.
- Співпраця з клієнтами важливіша перемовин за договором.
- Готовність до змін важливіша за дотримання плану.

Основний сенс полягає в наступному: проєкт може розвиватись та змінюватись, відповідно продукт, рішення чи результат проєкту можуть змінюватись разом з ним (рис. 2.3).

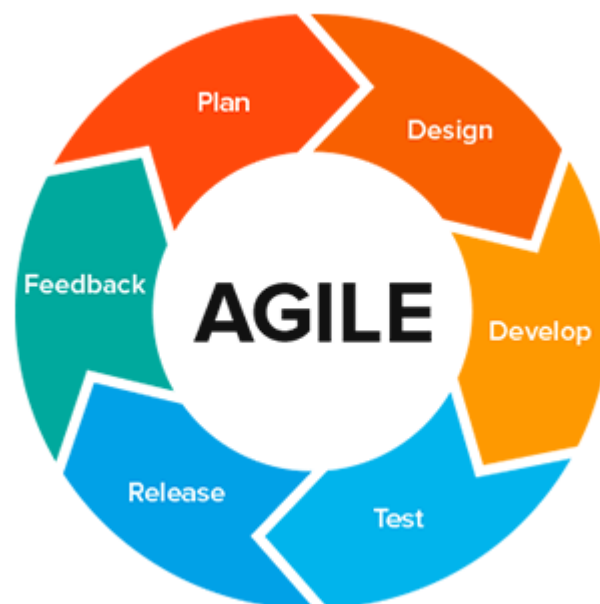


Рисунок 2.3 – Філософія Agile

На практиці робота за філософією Agile означає, що команда працює невеликими циклами, результат кожного — готова функція чи продукт. В наступному циклі вже працездатну функцію можуть допрацьовувати та удосконалювати. В Agile допускаються зміщення пріоритетів, паралельне виконання різних процесів (наприклад, тестування попереднього функціоналу

одночасно з розробкою нових опцій), повернення до попередніх етапів, якщо змінились вимоги до продукту, відсутність деталізованого ТЗ.

Agile характеризується постійним спілкуванням, взаємодією та наданням зворотного зв'язку всім причетним — від веб-розробників до клієнтів. При чому в команді всі рівні. Немає суворої регламентації і думка стажера-розробника така ж важлива, як досвідченого full-stack девелопера.

Переваги Agile:

- Максимальна гнучкість та свобода у внесенні змін. Наприклад, якщо конкуренти випустили новий функціонал, команда може швидко впровадити аналогічний також у свій продукт навіть якщо попередній план його не передбачав.
- Низький ризик невдалої реалізації проекту. Agile орієнтована на клієнта, адже зміна вимог вітається навіть на пізніх стадіях розробки, якщо це сприятиме створенню кращого рішення. Основні стейкхолдери постійно взаємодіють між собою. Результати кожного етапу презентуються замовнику, команда отримує фідбек і може швидко адаптувати продукт. Подібний підхід значно знижує ризик провалу проекту.
- Висока залученість команди. Філософія передбачає максимальну орієнтованість на людей — кожен учасник проекту однаково важливий, це підвищує працездатність та вмотивованість співробітників.

Звісно, є й недоліки. Наприклад, складно замінити когось із членів команди так як від кожного потрібна активна залученість до робочих процесів. Крім того, відсутність чіткого плану та структури, ускладнює управління ресурсами. Інколи, новачки-«проджекти», які працюють за Agile, стикаються з тим, що проект перетворюється на каскад постійних та безрезультативних змін. Успішне впровадження цієї філософії складніше, ніж моделі Waterfall, і проектним менеджерам без досвідченого наставника, із досвідом роботи над проектами за Agile, не обійтись.

Agile підійде: проектам в різних нішах, в яких є загальне уявлення про продукт, але немає бачення конкретного результату. Вони зазвичай вимагають гнучкості, швидких змін і здатності підлаштовуватися під нові вимоги ринку. Зважаючи на те, що на сьогодні інформаційні технології — найдинамічніша галузь, концепція Agile присутня в більшості ІТ-компаній, а якщо точніше — одна з варіацій цієї філософії.

В реальності, в проектному управлінні впроваджують не самостійну філософію Agile, а її фреймворки, які є частиною концепції — Scrum, eXtreme Programming (XP), Kanban, Scrumban. Всі вони відповідають гнучкості Agile і відрізняються окремими інструментами та підходами до управління. Ознайомимось з особливостями кожного фреймворка.

Scrum – найпопулярніший фреймворк за філософією Agile. Багато компаній, які практикують Scrum-підхід, навіть наймають в штат окремого фахівця — Scrum-майстра. Його основне завдання — усунути всі перешкоди на шляху до більш ефективного виконання роботи.

Особливість Scrum полягає в командному підході та нестандартному розподілі обов'язків всередині колективу. До процесів залучені не лише співробітники компанії, а й бізнес-замовники, які приймають участь в створенні продукту активніше, ніж за інших підходів, і роблять це переважно у форматі особистого спілкування, а не через документацію.

На старті створюється та пріоритизується перелік усіх вимог до продукту — беклог — своєрідний загальний список задач, які необхідно реалізувати на всіх етапах. Далі робота поділяється на короткі фази — спринти, тривалістю 2-4 тижні. Так, команда проекту ділить масштабні завдання на невеликі задачі, працює над ними протягом спринту, щодня проводячи невеликі мітинги — стендапи. На стендапах кожен учасник звітує, що було зроблено, що потрібно зробити, та які проблеми є в реалізації тієї чи іншої фічі. В кінці спринту команда має презентувати вимірний результат, який буде частиною готового продукту (рис 2.4).

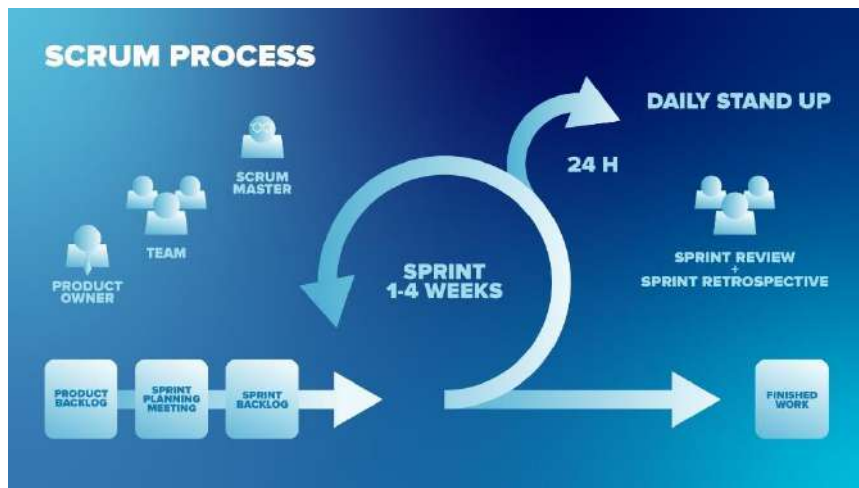


Рисунок 2.4 – Фреймворк Scrum

Фреймворк Scrum спрямований на стимулювання гнучкості, креативності та надання високоякісних результатів. Завдяки розподіленню великої задачі на частини, дозволяє запобігти втоми та виснаження, а постійна взаємодія в колективі сприяє оперативному усуненню проблем та труднощів, що можуть завадити успішному закриттю завдань. Фреймворк Scrum актуальний для невеликих команд з 5-10 учасниками.

Kanban-фреймворк ще більш гнучкий, ніж Scrum. Його сенс полягає в візуалізації робочих процесів за допомогою Kanban-дошок з картками, в яких описуються задачі. Відповідно до зміни статусу задачі, картки переміщуються у різні стовпці, наприклад від «Прийнято в роботу» до «Готово», а між цими етапами може бути ще декілька: «В роботі», «Тестується» та ін. При цьому майже завжди картки пріоритизуються і найбільша увага приділяється завданням з позначкою «Високий пріоритет». Крім чітких критеріїв переходу задачі від стадії до стадії, Kanban передбачає обмеження в кількості завдань, не взятих до роботи. Це дозволяє уникнути неефективності багатозадачності. Спринтів в Kanban не передбачено.

Kanban-дошки формуються в спеціалізованих онлайн-інструментах — Trello, Jira та аналогічних. Особливо зручним є цей підхід для віддалених команд (рис 2.5).

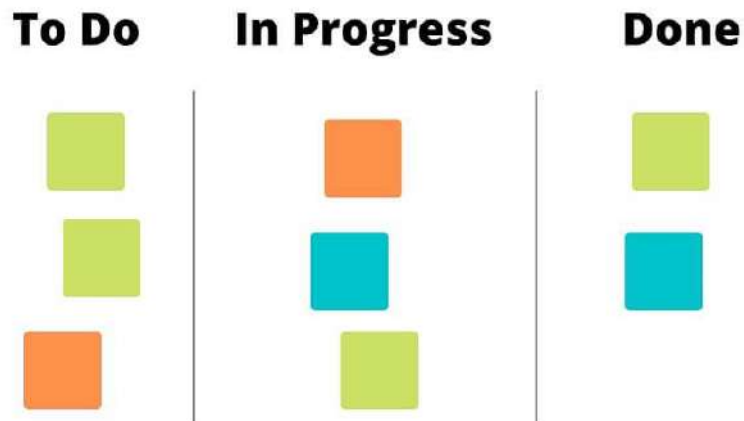


Рисунок 2.5 – Фреймворк Kanban

Методологія Scrumban – гібрид двох вищеописаних підходів, що об’єднує в собі кращі риси обох фреймворків. У Scrumban використовується цикл зі спринтами (згідно зі Scrum) та робота з дошками (як у Kanban). Поєднання ключових елементів різних фреймворків дозволяє:

- точніше визначати тривалість ітерацій. Спринт планується на підставі показників минулих результатів, з урахуванням кількості виконаних завдань;
- правильно розставляти пріоритети для кожного завдання, обмежуючи їхню кількість;
- краще контролювати роботу команди шляхом проведення щоденних коротких стендапів.

Scrumban використовує систематизацію складнішого фреймворку Scrum і візуалізацію легшого у сприйнятті Kanban. Це дає змогу командам отримувати об’єднані переваги двох популярних підходів в управлінні ІТ-проектами.

Scrumban

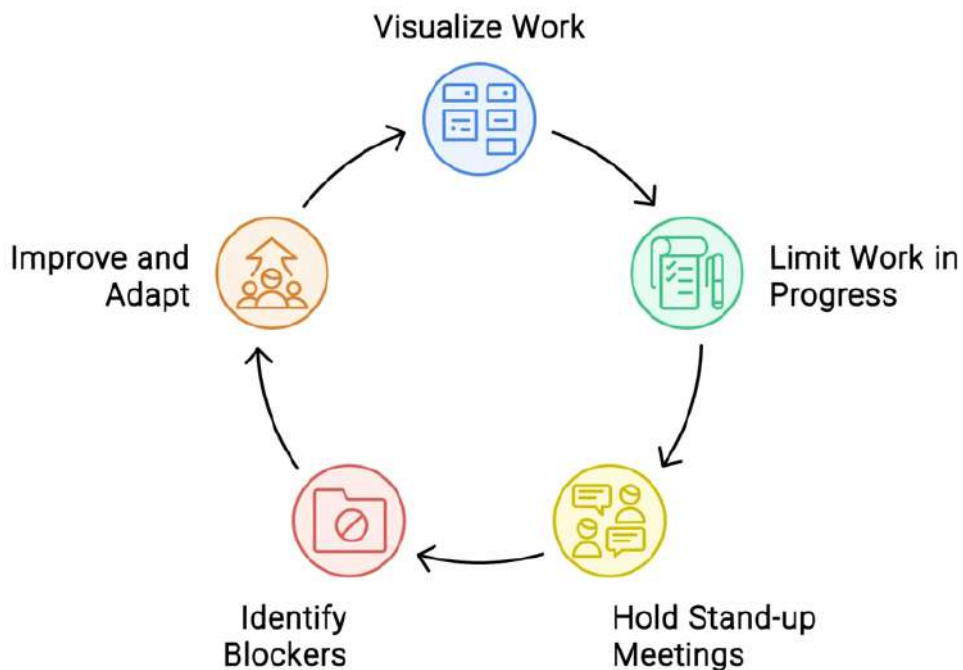


Рисунок 2.6 – Методологія Scrumban

eXtreme Programming – ще один гнучкий фреймворк, який використовується для динамічних проєктів із стислими термінами. Ціль XP — впоратися зі швидкими змінами вимог до продукту — і в результаті підвищити якість і процесів, і результатів. Фреймворк передбачає свій набір цінностей, в який входять простота, комунікація, зворотний зв'язок і сміливість. А також — певні правила eXtreme Programming, що охоплюють всі етапи — від планування до тестування.

Lean — це скоріше філософська концепція, яку часто відносять до методології. Її головна ідея — максимізувати цінність для споживача, мінімізуючи витрати. Простіше кажучи, більше цінності = менше ресурсів.

Будучи філософією, Lean сама по собі не пропонує конкретних інструментів візуалізації чи управління. Суть Lean у концептуальному підході до дослідження цінності продукту та створення стратегії покращень: покажи на ринку продукт, зрозумій чи потрібен він, а потім покращуй в залежності від реальних вимог ринку.

Тобто компанія надає користувачам не до кінця удосконалену пропозицію, щоб оцінити її затребуваність, отримати відгуки та виправити недоліки.

Lean підійде: стартапам, оскільки ідеально «вписується» під теорію роботи в умовах невизначеності — коли потрібно працювати без прибутку та знизити витрати. Хоча загалом актуальна будь-якому бізнесу, де можна створити MVP — *minimum viable product*, мінімально життєздатний продукт. Наприклад, служба таксі планує розробити окремий застосунок виклику автівки. Компанія може запуснути програму з простим дизайном та мінімальним функціоналом, зібрати зворотний зв'язок і випустити удосконалений додаток із врахованими побажаннями від клієнтів.

З PRINCE2, як і у випадку з Lean, також виникає плутанина. За рахунок жорсткої регламентації її інколи називають підвидом Waterfall. Насправді — це повноцінна, офіційна методологія, основана за каскадним типом. Вона була розроблена Центральним телекомунікаційним та комп'ютерним агентством (Central Computer and Telecommunications Agency — ССТА) Великої Британії у 1989 році як стандарт управління ІТ-проектами, і наразі використовується здебільшого в соціальній сфері та для масштабних ІТ-ініціатив. Методологія PRINCE2 базується на 7 принципах, 7 темах та 7 процесах. Всі вони повинні вписуватися в 6 обмежень: час, гроші, ризики, вигоди, якість та зміст проекту. При чому принципи, теми та процеси в PRINCE2 пов'язані та взаємодіють один з одним. Принципи формують загальну основу управління проектом. Теми допомагають керувати важливими аспектами проекту відповідно до цих принципів. Процеси, у свою чергу, забезпечують застосування цих принципів і тем у конкретних діях протягом усього проекту.

Ось, наприклад, сім принципів PRINCE2:

- Постійна оцінка доцільності проекту.
- Врахування попереднього досвіду: аналіз кожного етапу проекту, робота над помилками.
- Чітко визначені ролі та обов'язки.

- Поетапне управління.
- Комунікація за необхідністю — немає потреби в постійних мітингах, обговореннях, взаємодії із командою — достатньо надати вичерпні завдання. Втручатись варто лише коли порушується якийсь із шести обмежень.
- Фокус на продукті. Головна ціль — кінцевий продукт та його якість. Кожен етап необхідно закінчувати проміжним результатом, щоб переконатись, що він відповідатиме початковому плану.
- Адаптація принципів методології в залежності від потреб кожного проєкту — з урахуванням його розмірів, кількості залучених співробітників і т.п.

Компоненти (теми) PRINCE2 — обґрунтування проєкту, організація, якість, ризики, плани, відстеження розвитку проєкту, зміни.

Сім процесів методології PRINCE2: початок проєкту, управління, ініціація, контроль етапу, управління розробкою продукту, управління межами стадії (перехід на наступний етап), закриття проєкту (рис. 2.7).



Рисунок 2.7 – Методологія PRINCE2

Деталізований підхід та документування всіх процесів PRINCE2 надає проєктним менеджерам та керівництву більше контролю над ресурсами, продуктивністю, персоналом, витратами та ризиками. Більше того, методологія

пропонує чітко визначені ролі та полегшує управління. Однак коли потрібна гнучкість, постійно змінюються вимоги — вона не дасть бажаного результату. Команда не встигне швидко відреагувати на зміни, заповнюючи купу звітів та списків помилок. Крім того, PRINCE2 мінімізує спілкування з усіма учасниками проекту, що може призвести до непорозумінь, конфліктів та зниження вмотивованості членів команди. Ще один мінус — відсутність рекомендацій до застосування конкретних інструментів, наприклад Kanban-дошок, а зважаючи на «бюрократичний» підхід PRINCE2, самодіяльність за цією методологією не вітається.

Методологія Six Sigma зазвичай застосовується для контролю якості створення продукту, а не для управління проектом (рис 2.8). Основна ціль Six Sigma — постійне покращення процесів та усунення помилок. Загальний підхід до її упровадження виглядає наступним чином: попередньо визначається, що має бути в майбутньому проекті, і в процесі його реалізації відбувається пошук недоліків, їх виправлення та відповідно покращення продукту. Теорія Six Sigma спирається на шість принципів:

- Орієнтованість на клієнта — необхідно відслідковувати потреби клієнтів, аналізувати та враховувати їх.
- Управління процесами на основі перевірених фактів — не можна покладатися на припущення.
- Занурення в робочі процеси — щоб зрозуміти, що потребує покращення, спочатку потрібно вивчити принципи взаємодії процесів у проекті.
- Робота на випередження — продумувати хід проекту, враховувати можливі наслідки та результати діяльності.
- Залученість членів команди — максимальна прозорість в робочих процесах.
- Постійне покращення і готовність до ризиків — власне на удосконаленні роботи і ґрунтується методологія. А ось з ризиками складніше — до них варто ставитися лояльно і не боятися допустити, при цьому долати невдачі, що виникають, і отримувати з них уроки.



Рисунок 2.8 – Модель Six Sigma

Передбачені методологією Six Sigma і етапи:

- Define — визначення. Формується зміст проекту і його цілі.
- Measure — вимір. Визначаються побажання та очікування замовника (кінцевого споживача), його вимоги. В подальшому зібрані дані допомагають ідентифікувати те, що потребує покращення.
- Analyze — аналіз. Досліджуються всі проблеми у проекті, потенційні недоліки продукту та варіанти їх рішення/усунення.
- Improve — покращення. Виявлені недоліки усуваються та удосконалюються бізнес-процеси.
- Control — контроль. Після того, як проблеми виправлені, відбувається перевірка, наскільки це позитивно вплинуло на продукт.

Зважаючи на те, що Six Sigma спрямована саме на процес, а не на проект, а в її основі — постійне удосконалення роботи, зазвичай її застосовують в поєднанні з Agile та називають — Agile Six Sigma.

Метод критичного шляху (CPM) — це метод планування процесів проекту, що дозволяє визначати критичні та некритичні завдання, і запобігати їх невчасному закриттю.

Згідно із CPM, в розрахунок беруться всі завдання, оцінюється приблизний час їх реалізації. При цьому обов'язково враховуються зв'язки між завданнями. Тобто вплив одних на інші. Найдовша серія завдань, які мають бути виконані

відповідно до графіка, щоб проект був завершений вчасно, називається «критичним шляхом». Будь-які затримки в роботі вплинуть на решту завдань, і відповідно на терміни реалізації всього проєкту.

Метод СРМ дає розуміння найважливіших задач в хронології проєкту, залежностей між ними, дозволяє визначити, що можна виконувати паралельно, а які завдання потребують чіткої послідовності.

В СРМ основний фокус саме на часі, при цьому упускаються інші важливі фактори, такі як оптимізація витрат та ресурсів. Тому доцільніше Метод критичного шляху використовувати саме як інструмент планування робочих процесів, пріоритизації завдань та порівняння очікувань з фактичним ходом проєкту, поєднавши його, наприклад, з Waterfall.

Більш того, в сучасному ІТ-світі ця тактика дуже часто практикується. Об'єднують не лише фреймворки, як у випадку із Scrumban, а й комбінують гнучкі та каскадні підходи. Наприклад, в управлінні проєктами навіть виділяють окрему гібридну методологію з елементами Waterfall та Agile, яку ще називають «Структурованим Agile». В ній зібрані ретельне планування, збір та аналіз вимог від першої, і можливість змін — від другої.

Зазвичай, проєктний менеджер беруть за основу певну методологію чи фреймворк та адаптують його під індивідуальні особливості та потреби проєкту, додаючи елементи іншого підходу. І тут дуже важливо відмінно знати специфіку, плюси та мінуси основних моделей і методик [13].

Головне в гібридному управлінні проєктами — можливість для проєктного менеджера вибирати методології, які на його думку даватимуть найкращий результат з урахуванням стартових позицій та кінцевих цілей. Це той випадок, коли багато чого залежить саме від досвіду та рівня менеджера.

Коли йдеться про методології управління проєктами, не може бути універсального підходу. Кожна з моделей пропонує унікальні принципи ведення проєкту від початкової стадії до завершення. Звертати увагу слід, перш за все, на стиль роботи колективу та на специфіку продукту, який потрібно впровадити. Ось ще кілька факторів, які необхідно враховувати під час вибору:

Сфера діяльності. Варто дослідити, чи є методології, загальноприйняті в галузі, і знайти відповідь чому саме вони. Тобто потрібно зважати на досвід компаній-лідерів в ніші, як позитивний, так і негативний. Наприклад, більшість ІТ-фірм користується фреймворками Agile за рахунок стрімких змін на ринку, а у промисловому секторі, де присутні давно сформовані стандарти, часто застосовують каскадні моделі, що передбачають багато документації, звітів і ретельний контроль.

Складність проекту та розмір команди. Деякі методології, наприклад PRINCE2, є недоцільними для простих та короткотривалих проектів, тоді як популярний Scrum не підійде великим командам, в яких більше 10 співробітників.

Комунікація. Важливо враховувати, наскільки тісними можуть бути контакти зі стейкхолдерами (виконавці, підрядники, замовники). Як часто є змога з ними спілкуватися та наскільки швидко вони готові реагувати на запитання та пропозиції.

Пріоритети проекту. Зрозуміло, що кожен проект передбачає у підсумку випуск продукту чи вимірні результати. Але слід з'ясувати, що важливіше: дотримання бюджету та дедлайнів, а може доведений до ідеалу функціонал, експерименти та інновації чи точне врахування стандартів і умов замовника, однакова залученість всіх причетних або ж розподілення ролей з чітко розмежованою зоною відповідальності кожного учасника. Цінності та пріоритети методології і проекту мають збігатися.

Проектним менеджерам, що починають працювати, потрібно розбиратися хоча б в найпопулярніших моделях управління проектами — Waterfall та Agile (і його фреймворках), і розуміти як їх застосовувати на практиці. Відповідні питання майже завжди присутні на співбесідах з кандидатами, які претендують на посаду Project Manager. Причому рекрутери можуть не просто запитати суху теорію на кшталт в чим відрізняються каскадний концепт від ітеративного (гнучкого), а й змодельовати певні ситуації та запропонувати претенденту поміркувати як він буде діяти в рамках того чи іншого підходу. Від обраної методології буквально

залежить хід проекту: планування робочих процесів, взаємодія із командою і клієнтами, актуальні інструменти та ін [14].

2.3 Інструменти для управління проектами

Управління завданнями включає створення проєктів, постановку цілей і завдань, призначення завдань певним членам команди, визначення дедлайнів, а також обмін документами та коментарями. У деяких випадках управління завданнями включає не тільки планування роботи, а й ширше коло обов'язків: управління ресурсами, збір аналітики та звітності, а також співпрацю.

Trello, Jira, Asana — це інструменти, які переводять управління проектами у цифровий формат і допомагають зберігати все, що стосується виконання ваших проєктів (цілі, деталі завдань, посилання, референси, списки відповідальних осіб, дедлайни та робочу документацію) в одному місці.

Майже над кожним робочим проєктом працює більше однієї людини. Крім того, деякі проєкти вимагають онлайн-залучення сторонніх експертів, клієнтів або фрілансерів.

Інструменти комунікації дозволяють спілкуватися в чаті або проводити онлайн-зустрічі з колегами, партнерами чи будь-ким іншим. А інструменти співпраці призначені для спільної роботи над проєктами, редагування робіт один одного, коментування в ході роботи та швидкого оновлення робочих документів.

Якщо говорити про приклади інструментів управління проектами з продуманим функціоналом для спільної роботи, то Slack, Basecamp, G Suite, Office 365, Trello та Yammer є одними з лідерів індустрії. Для простої співпраці над проєктом буде достатньо месенджера, за допомогою якого можна обмінюватися документами або організувати групові дзвінки. Однак не кожна платформа є достатньо захищеною для обміну конфіденційною інформацією. Серед більш безпечних рішень — Microsoft Teams, Zoom, Flock та Discord.

Інструменти управління проектами можуть включати функції для планування завдань, що є основним способом автоматизації робочого процесу. Планування завдань може бути прямим і простим (коли ставиться завдання з

дедлайном) або містити взаємозалежності. Більшість інструментів, які використовують менеджери, також пропонують налаштування нагадувань для виконавців завдань. Ця функція надзвичайно важлива, якщо команда працює над кількома проєктами одночасно.

Інструменти тайм-менеджменту дозволяють вам поставити велике завдання, яке потім розбивається на менші завдання для членів команди. Вони повинні виконати їх за кілька днів або тижнів з власними дедлайнами. Деякі платформи можуть запропонувати підтримку на основі штучного інтелекту.

Якщо управління часом на рівні команди є важливою частиною роботи, то для подібних проблем розроблено додатки GanttPro, TimeCamp, Airtable, Wrike, Clockify, Time Doctor або Trello. Хоча ці інструменти не спрямовані виключно на планування або управління часом, вони все одно мають розширений функціонал.

Інструменти, які використовують в управлінні проєктами продуктиві команди, а також середній і великий бізнес, повинні містити функції для планування та відстеження розподілу ресурсів. До переліку завдань менеджера виробничої або цифрової продуктової компанії входить управління фінансовими, товарно-матеріальними, нерухомими, інтелектуальними та іншими ресурсами, що належать компанії.

Іноді планування ресурсів стосується питань накопичення або придбання ресурсів на майбутнє (наприклад, комп'ютерних програм, обладнання, наймання нових спеціалістів). У таких випадках менеджер також звертається до аналітики для прийняття рішень.

Залежно від сфери бізнесу та ролі в ньому є сенс використовувати більш галузеві інструменти. Однак, щоб почати, необхідно розглянути найбільш універсальні рішення в контексті управління ресурсами: Mavenlink, Clarizen, LiquidPlanner, Paymo, NetSuite і Teamdeck.

Звітність — важлива частина роботи менеджерів та членів команди. Однак звіти та аналітика щодо ефективності роботи команди, використання ресурсів компанії та багатьох інших аспектів відрізняються за рівнем складності.

Порівнюючи інструменти управління проектами, можна дійти висновку, що більшість платформ пропонують динамічні дані про проекти у вигляді дощок та діаграм Ганта. Якщо вам цього недостатньо, обирайте інструменти, які дозволяють інтегрувати відомі інструменти аналітики (Python, Apache Spark, Microsoft Power BI або Tableau).

Monday.com, Height, Kintone та Hub Planner — це інструменти для відстеження управління проектами, які дозволяють бачити інфографіку на основі того, що відбувається у проектах. Можете легко експортувати дані, щоб використовувати їх у своїх щомісячних і річних звітах та презентаціях.

Існує безліч інструментів управління проектами. До того ж у кожній категорії є багато варіантів, які на перший погляд здаються дуже схожими. Однак деякі інструменти управління проектами й завданнями більше підходять для фрілансерів, ніж для менеджерів великих команд. Крім того, популярні інструменти управління проектами можуть коштувати значно дорожче, ніж рішення від стартапів, які, своєю чергою, можуть бути більш прогресивними й зручними. І є багато інших нюансів, які варто врахувати.

Щоб уникнути помилок при виборі інструмента управління проектами, необхідно дотримуватися наступних кроків:

Зрозумійте, навіщо вам потрібен інструмент управління проектами — визначте свої цілі. Основні інструменти управління маркетинговими проектами дозволяють управляти завданнями, планувати роботу і ресурси, спілкуватися з командою й клієнтами, розраховувати робочий час та впорядковувати робочі документи. Також треба визначити, зі скількома людьми вам потрібно спілкуватися за допомогою інструмента.

Проаналізуйте, які програми управління проектами доступні на ринку — виходячи зі своїх потреб, складіть список інструментів управління проектами, які пропонують необхідні вам функції та інтеграцію. Зверніть увагу на відгуки про обрані інструменти, зокрема відгуки схожих команд, або компаній, що працюють у тій самій сфері.

Протестуйте платні або безкоштовні онлайн-інструменти управління проектами – використовуйте безкоштовні або пробні версії програм. Рекомендовано перевірити їх самостійно, щоб оцінити зручність та функції. Якщо необхідно знайти рішення для управління великою кількістю проектів або у компанії є кілька спеціалізованих команд, спочатку протестуйте обраний інструмент з одним проектом і невеликою командою. Враховуйте, що деякі інструменти управління проектами для малого бізнесу — не найкращий вибір для управління великими командами та кількома проектами.

Необхідно обрати план або підписку відповідно до свого бюджету – кожен інструмент має кілька варіантів підписки. Зазвичай ціна визначається функціоналом, обсягом хмарного сховища для ваших проектів та кількістю користувачів. Необхідно розрахувати, який план впишеться у бюджет.

Оцініть інструменти управління проектами, які використовуються – через певний час важливо оцінити, чи приносять користь команді, компанії обрані інструменти управління проектами. Це може бути місяць, конкретний період реалізації проекту або рік. Якщо помітного впливу на ефективність роботи немає, необхідно знайти інше рішення.

Програми, які використовують менеджери та їхні команди, можуть відрізнятися залежно від специфіки роботи. Деякі онлайн-інструменти управління проектами орієнтовані на злагоджену співпрацю в команді, тоді як інші допомагають контролювати графіки або розподіляти ресурси компанії.

Trello – простий інструмент для спільної роботи над проектом. Ключові особливості: різні формати відображення проектів, автоматизація робочого процесу, шаблони, інтеграції.

Trello – один з найпопулярніших інструментів для управління проектами, який найкраще підходить для співпраці в малих та середніх командах. Ця платформа має інтуїтивно зрозумілий інтерфейс і дозволяє створювати списки, дошки та картки з описами завдань. Також у Trello можна ділитися документами та спілкуватися всередині команди. Серед недоліків — обмежена кількість

інструментів для створення звітів і складність управління великою кількістю проектів одночасно (рис 2.9) [15].

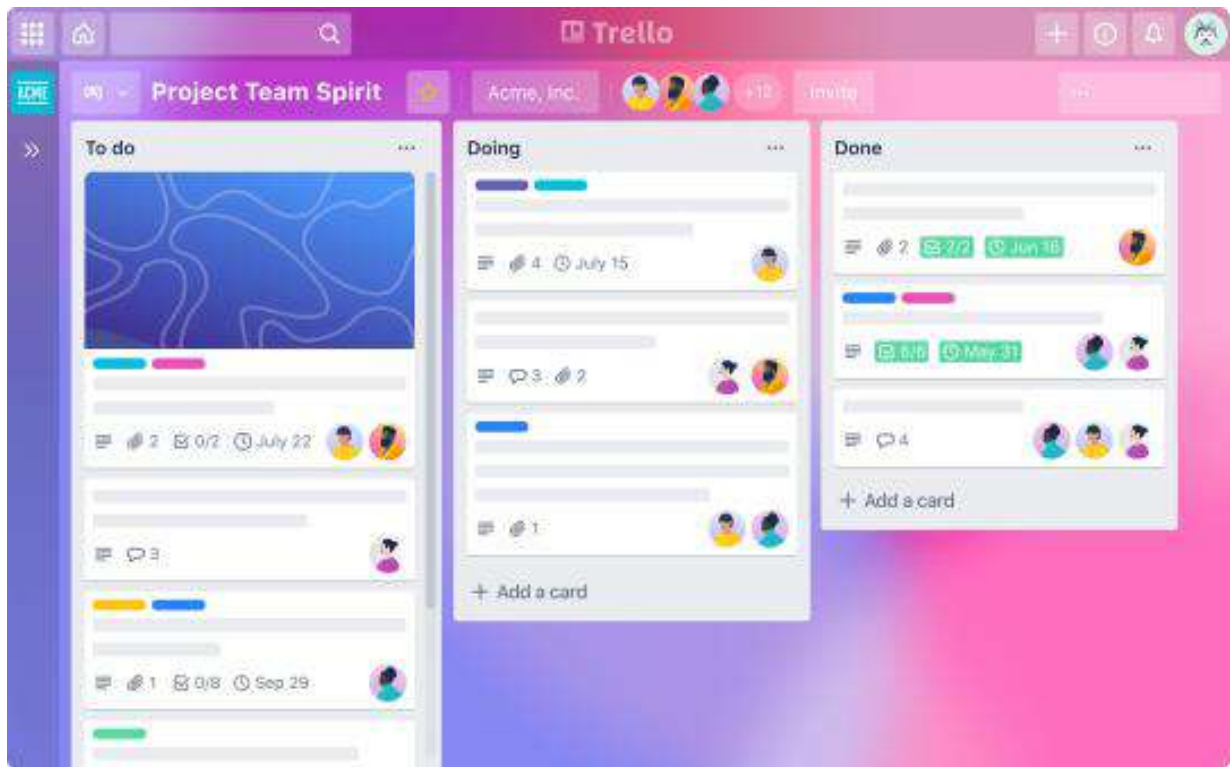


Рисунок 2.9 – Інтерфейс Trello

Asana – найкращий інструмент для організації проектів. Ключові особливості: ШІ для бізнес-рішень, конструктор робочого процесу, відображення часової шкали, дошки, календар, інтеграції, звітність, постановка цілей, форми, автоматизація.

Asana є одним з найкращих інструментів для управління проектами, котрий використовують як невеликі компанії, так і великі корпорації. Ця платформа призначена для командної взаємодії, відстеження завдань і планування роботи. Однією з її переваг є функція кастомізації проектів, а також можливість запрошувати необмежену кількість гостей для перегляду проектів. Asana чудово підходить для тих, хто працює над багатьма проектами одночасно, а її інтерфейс простий і зрозумілий. Платформа також підтримує інтеграцію з іншими інструментами, які використовують у роботі, наприклад, календарями (рис 2.10) [16].

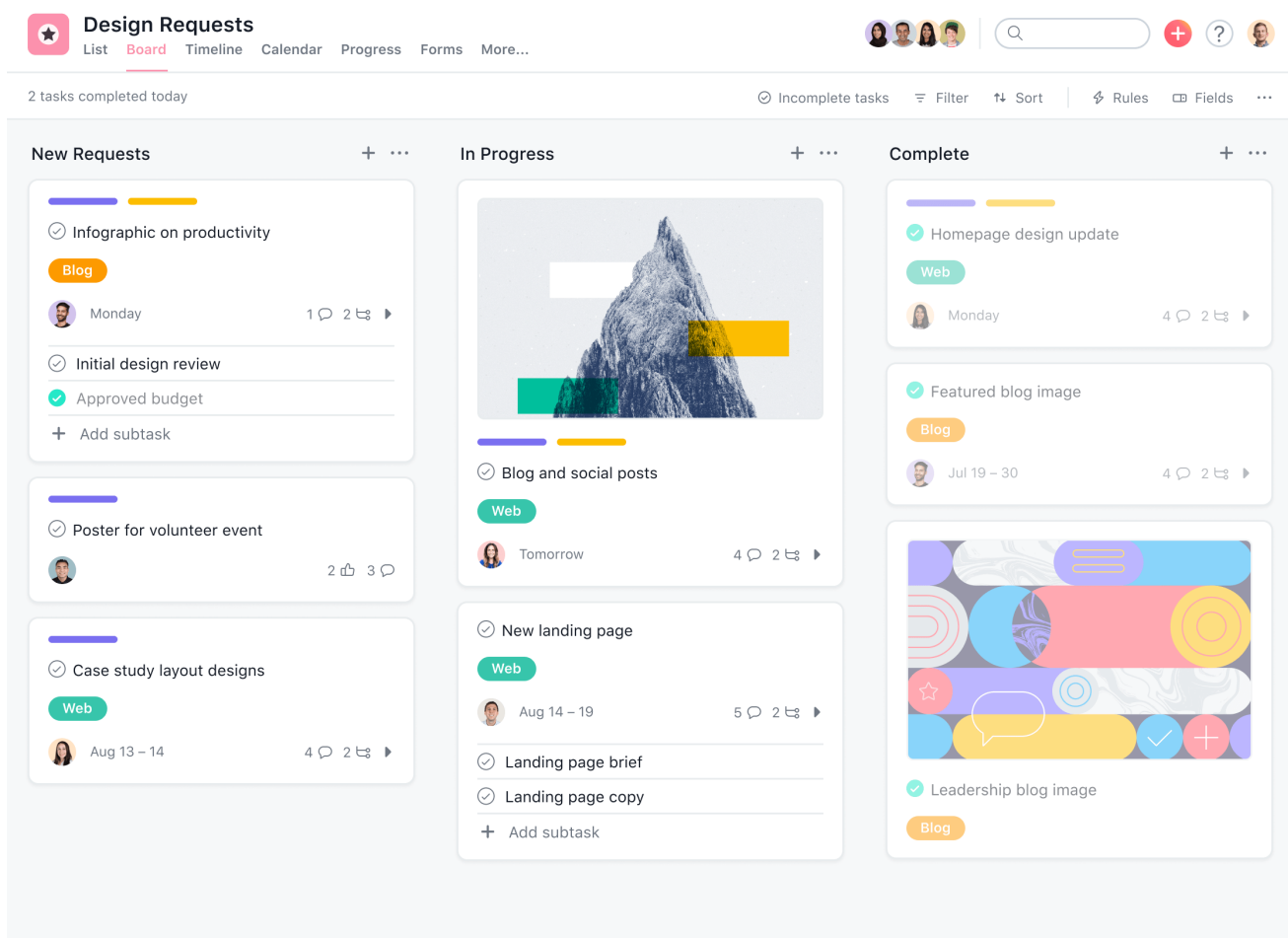


Рисунок 2.10 – Інтерфейс Asana

Jira – популярний інструмент управління проектами за філософією Agile. Ключові функції: Scrum- і Kanban-дошки, дорожні карти, відображення часової шкали, звіти та інсайти, співпраця та комунікація.

Jira – один із гнучких інструментів для управління проектами, котрий підходить для компаній будь-якого розміру. Ключовою перевагою цієї платформи є можливість інтеграції як з інструментами Atlassian Group, так і зі сторонніми платформами. Також можна налаштувати функціонал Jira для своєї команди. Ця платформа дозволяє створювати завдання та відстежувати прогрес і дедлайни у форматі інформаційної панелі. Ще однією перевагою є унікальна та зручна навігація, яка значно економить ваш час. Недоліком інструменту є обмежені можливості для управління ресурсами та звітності (рис. 2.11) [17].

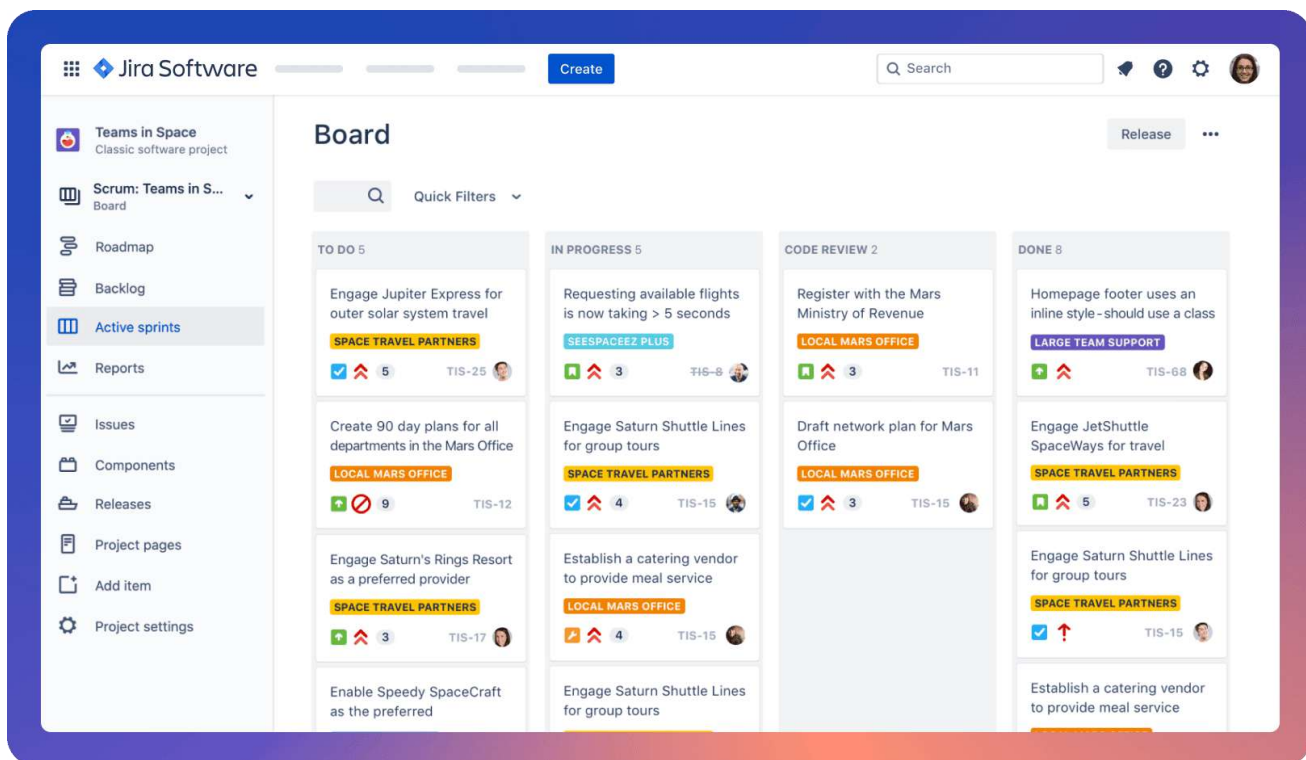


Рисунок 2.11 – Інтерфейс Jira

Microsoft Project – ефективний інструмент для управління проектами. Ключові особливості: різні формати відображення проектів, управління ресурсами, інтеграція із застосунками Microsoft 365, дорожні карти, звіти, спільна робота.

У Microsoft є кілька інструментів управління проектами, але саме Microsoft Project дозволяє використовувати переваги всіх, поєднуючи потрібні вам функції. Платформа інтегрується з продуктами Microsoft для командної роботи або аналітики (включаючи Teams, Outlook і Power BI), а також Slack, Google Calendar і багатьма іншими. Перевагою інструменту є його спрямованість на конкретний проект, а також функції для управління ресурсами та бюджетного планування. Ще один аргумент на користь Microsoft Project — наявність діаграми Ганта й індивідуальне відображення прогресу у форматі інфографіки (рис 2.12) [18].

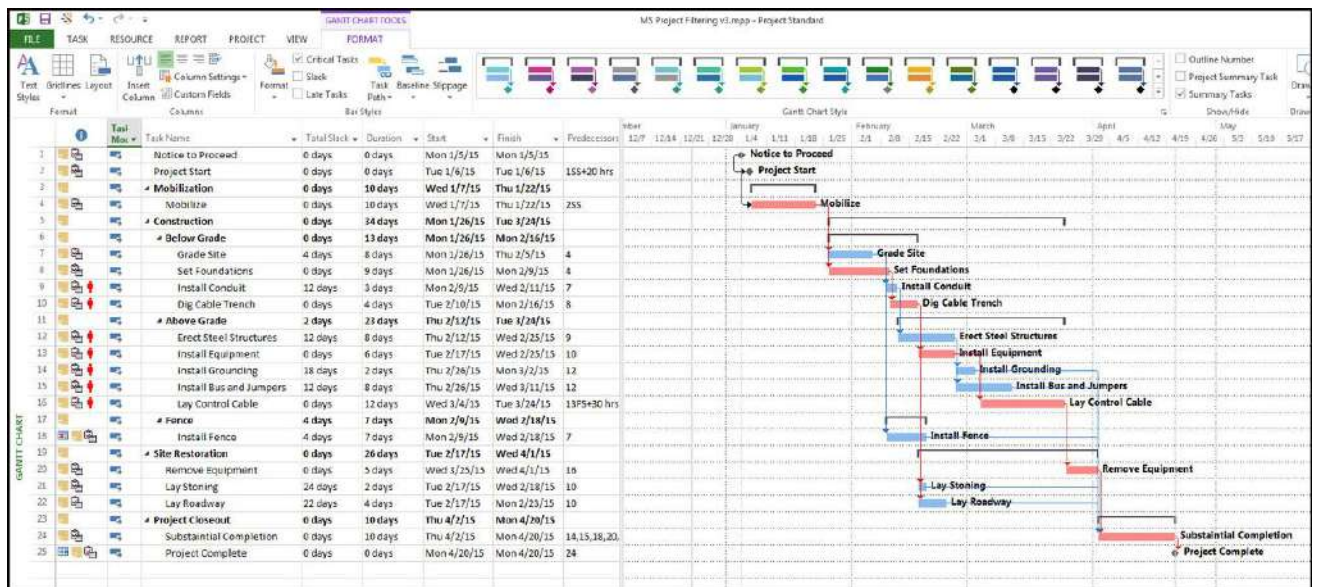


Рисунок 2.12 – Інтерфейс Microsoft Project

Notion – найкращий інструмент управління проектами для креаторів. Ключові особливості: API, інформаційні панелі активності, управління бюджетом, співпраця та комунікація, шаблони, управління документами, відстеження етапів, звітність, планування, конструктор робочих процесів.

Notion є одним зі зручних інструментів управління проектами для агенцій, невеликих команд та незалежних креаторів, але він також пропонує рішення для підприємств. Цей інструмент дозволяє зберігати дані про ваші проекти, створювати дошки, додавати ключові дані про них до календаря та переглядати терміни виконання проектів. Notion найкраще підходить для роботи над невеликою кількістю творчих проектів та зберігання документів в одному місці. Платформа пропонує зрозумілу навігацію й інструменти для співпраці з членами команди. Остання версія Notion працює на основі штучного інтелекту. Наприклад, платформа може запропонувати вам спосіб розділити велике завдання на частини (рис 2.13) [19].

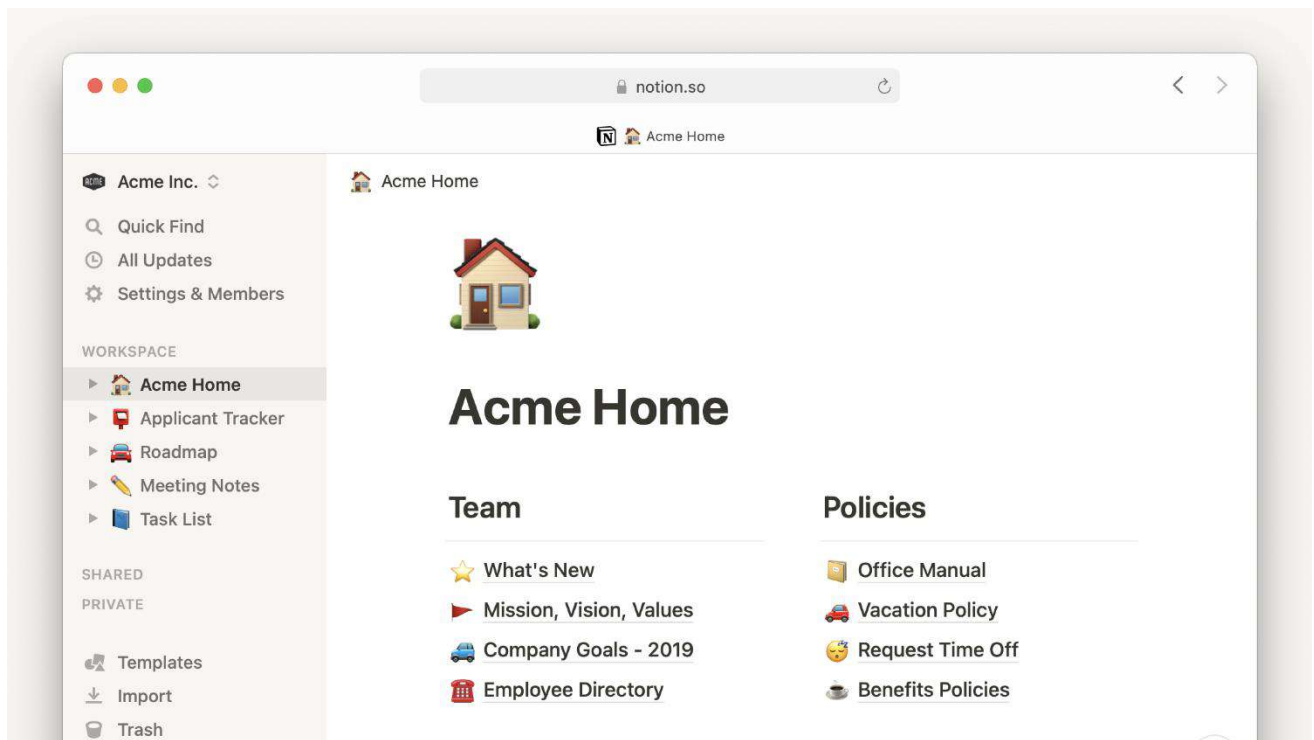


Рисунок 2.13 – Інтерфейс Notion

Щоб отримати загальне уявлення про функціональні можливості сучасних платних і безкоштовних онлайн-інструментів для управління проектами, розглянемо їхні ключові функції:

- Постановка і коригування завдань;
- Список справ, дошки завдань (Kanban), робота з календарями або діаграмами Ганта;
- Комунікація та співпраця через коментарі, особисті та групові повідомлення, відеодзвінки та онлайн-конференції;
- Відстеження робочого часу та виставлення рахунків;
- Звітність щодо статусу проекту та роботи команди в цілому;
- Інтеграція з іншими хмарними інструментами (для аналітики, планування або управління ресурсами);
- Управління ресурсами;

- Використання ШІ для планування роботи (наприклад, автоматизований розподіл завдань, встановлення дедлайнів або поділ великих завдань на менші).

Вибір найкращої програми для управління проєктами залежить від потреб команди та сфери діяльності[20].

2.4 Висновки

За результатами досліджень для управління даним проєктом було обрано філософію Agile, завдяки її гнучкості. Враховуючи досвід команди проєкту та організації, на яку вона працює, можливість швидко вносити і реагувати на зміни в технічному завданні проєкту стала ключовим аспектом, через який було обрано саме Agile.

Слід зазначити що в проєкті використовувалася не безпосередньо Agile, а методологія що поєднує два її фреймворки Scrum і Kanban – Scrumban. Ця методологія була обрана через свою зрозумілість і легкість реалізації. Подібність до Kanban-дошок було реалізовано завдяки MS Project, а щотижневі спринти та зв'язок між усіма зацікавленими сторонами відбувався через MS Teams.

При виборі інструменту для управління проєктами менеджер проєкту надав перевагу MS Project, так як вище керівництво організації користується середою Microsoft. Окрім самого MS Project також використовувалася платформа MS Teams, де проводилися щотижневі спринти та проходила основна комунікація команди проєкту, а також дозволяла імпортувати таблиці з MS Project, що дозволяло робити швидкі правки в проєкті.

3. УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ БАЗ ДАНИХ

3.1 Розробка та планування проєкту

Будь-який проєкт це сукупність багатьох фаз, починаючи від планування і постановки цілей проєкту, до завершальних етапів його реалізації.

Добре підготовлений проєкт – це запорука його вдалого завершення. Щоб закінчити проєкт в наведені терміни та оминати ризики необхідно одразу визначити його цілі, команду, засоби контролю, точки виконання і терміни проєкту.

3.1.1 Постановка цілей

Використовуючи постановку цілей проєкту за методом SMART були описані наступні цілі:

- **Specific (Конкретна):** Розробити базу даних для зберігання та управління інформацією про клієнтів медичного закладу, включаючи їхні контакти, історію хвороб та статус платежів.
- **Measurable (Вимірювана):** База даних повинна включати від 6 до 15 таблиць з відповідними зв'язками та підтримувати обробку до 10000 записів щодня.
- **Achievable (Досяжна):** Команда з семи розробників завершить розробку протягом чотирьох місяців.
- **Relevant (Актуальна):** Проєкт спрямований на покращення клієнтського обслуговування та оптимізацію внутрішніх бізнес-процесів лікарень та медичних закладів.
- **Time-bound (Обмежена в часі):** База даних має бути завершена до 17 липня 2024 року.

Після встановлення цілей проєкту необхідно провести SWOT-аналіз. Він дає змогу чітко побачити, на чому варто зосередитись, які загрози врахувати та

які можливості використати для успішної реалізації проекту розробки бази даних. Розроблений SWOT-аналіз проекту наведений на рисунку 3.1.

<p style="text-align: center;">Strengths (Сильні сторони)</p> <ol style="list-style-type: none"> 1. Технічна експертиза: Досвідчена команда розробників із знаннями в SQL, NoSQL, та сучасних технологіях баз даних. 2. Гнучкість в адаптації: Можливість швидкого внесення змін у базу даних відповідно до змін бізнес-процесів. 3. Висока продуктивність: Використання оптимізованих алгоритмів для швидкої обробки великих обсягів даних. 4. Інтеграція з іншими системами: Простота інтеграції бази даних з існуючими веб-додатками, ERP та CRM системами. 	<p style="text-align: center;">Weaknesses (Слабкі сторони)</p> <ol style="list-style-type: none"> 1. Високі початкові витрати: Витрати на розробку, налаштування інфраструктури та навчання персоналу. 2. Залежність від технічної команди: Обмежена кількість людей, які мають глибокі знання про архітектуру бази даних. 3. Ризик затримок: Можливі затримки через технічні проблеми або неправильне визначення вимог на початковому етапі. 4. Складність у масштабуванні: У випадку швидкого зростання бізнесу можуть виникнути труднощі з масштабуванням бази даних.
<p style="text-align: center;">Opportunities (Можливості)</p> <ol style="list-style-type: none"> 1. Автоматизація бізнес-процесів: База даних дозволить автоматизувати рутинні завдання, підвищуючи ефективність роботи. 2. Покращення клієнтського обслуговування: Завдяки швидкому доступу до актуальної інформації про клієнтів. 3. Вихід на нові ринки: Розширення бази даних під нові регіони або додаткові бізнес-лінії. 4. Використання сучасних технологій: Інтеграція з хмарними сервісами, аналітичними платформами, AI/ML для прогнозування та аналізу даних. 	<p style="text-align: center;">Threats (Загрози)</p> <ol style="list-style-type: none"> 1. Кіберзагрози: Ризик несанкціонованого доступу, витоку даних або атак на базу даних. 2. Зміни у законодавстві: Необхідність відповідності нормативним актам (наприклад, GDPR). 3. Конкуренція: Інші компанії можуть впроваджувати аналогічні або кращі системи. 4. Залежність від постачальників: У разі використання сторонніх сервісів (хмарні рішення, ліцензійні ПЗ) можуть виникнути ризики, пов'язані з їх надійністю та цінами.

Рисунок 3.1 – SWOT-аналіз проекту

3.1.2 Оцінка та аналіз ризиків

Одним з основних завдань, які розв'язують у межах управління проектами, є управління ризиками проектної діяльності, або управління ризиками проекту. Це завдання не відокремлюється від більшості інших функцій управління проектами. Ризики існують на всіх фазах і етапах проектної діяльності, тому функція управління ними є актуальною аж до закриття проекту. Відомі чотири основних методи управління ризиками: скасування, запобігання та контролювання, страхування та поглинання ризиків.

Скасування ризику означає відмову від певної діяльності чи таку істотну (радикальну) її трансформацію, у результаті якої ризик зникає.

Запобігання та контролювання ризику - це ефективна організація проектної діяльності, тобто коли її учасники мають змогу ефективно впливати на чинники ризику і зменшувати можливість настання несприятливої події. Контролювання

ризикую полягає в реалізації комплексу заходів, спрямованих на мінімізацію збитків після настання несприятливої події.

Страховання ризику передбачає зменшення збитків від діяльності за рахунок фінансової компенсації зі спеціальних страхових фондів.

Поглинання ризику - це такий спосіб діяльності, коли при матеріалізації ризику збитки повністю несе його учасник (учасники). Цей метод управління ризиками застосовують тоді, коли можливість ризику невелика чи збитки в разі його настання неістотно впливають на учасників проектної діяльності.

На основі визначених цілей та SWOT-аналізу проекту було визначено ризики проекту, та надано їх аналіз із зазначенням ймовірності виникнення та вагомості впливу на проект, де значенню 0 відповідає найменший шанс виникнення та впливу, а 1 – найбільший (рис. 3.2).

№	Назва ризику	Імовірність виникнення	Вплив на хід проекту
1	Недостатня продуктивність системи	0.2	0.8
2	Неправильне моделювання даних	0.2	0.6
3	Невідповідність вибраної технології	0.1	0.9
4	Невірне визначення вимог	0.5	0.5
5	Недотримання термінів	0.4	0.6
6	Перевищення бюджету	0.3	0.2
7	Несанкціонований доступ	0.5	0.9
8	Відсутність резервного копіювання	0.2	0.7
9	Невідповідність нормативним вимогам	0.2	0.3
10	Недостатня кваліфікація персоналу	0.2	0.4

Рисунок 3.2 – Аналіз ризиків

3.1.3 Створення команди проекту

Створення команди для проекту розробки баз даних – це важливий крок, оскільки успіх проекту значною мірою залежить від професійного складу команди та чіткого розподілу ролей. На момент обговорення необхідних ролей для реалізації проекту було обрано таких членів команди:

Керівник проєкту (Project Manager) – відповідає за загальну координацію проєкту: від складання плану, управління бюджетом та ресурсами до контролю термінів і якості виконання завдань. Він також підтримує комунікацію з замовником і зацікавленими сторонами, забезпечуючи узгодження всіх дій команди з очікуваннями клієнта. Зона відповідальності:

- Планування проєкту: складання графіків, визначення ключових етапів проєкту та контроль їх виконання.
- Розподіл ресурсів: забезпечення достатніх ресурсів (людських, фінансових, технічних) для успішного виконання проєкту.
- Управління ризиками: визначення потенційних ризиків, розробка планів їх пом'якшення.
- Комунікація з замовником: забезпечення зв'язку між командою та зацікавленими сторонами, отримання зворотного зв'язку та затвердження ключових рішень.
- Контроль якості та бюджету: моніторинг відповідності проєкту вимогам, бюджету та термінам виконання.

Архітектор бази даних (Database Architect) – відповідальний за загальну архітектуру бази даних, включно з вибором структури, типом даних, їх нормалізацією та визначенням взаємозв'язків між сутностями. Архітектор формує концептуальну та логічну модель бази даних і тісно співпрацює з бізнес-аналітиком та командою розробників. Зона відповідальності:

- Проектування структури бази даних: розробка концептуальної та логічної моделей, вибір структур даних та їх взаємозв'язків.
- Оптимізація структури: застосування методів нормалізації та оптимізації для підвищення продуктивності бази даних.
- Технологічний вибір: визначення, яку СУБД використовувати, вибір методів для оптимальної обробки даних.

- Довгострокова масштабованість: проектування бази з урахуванням можливого росту та розширення в майбутньому.
- Забезпечення відповідності стандартам: впровадження галузевих стандартів у проектування бази даних.

Бізнес-аналітик (Business Analyst) – збирає вимоги від замовника або користувачів, документує їх та перетворює на технічні вимоги. Бізнес-аналітик також допомагає архітектору зрозуміти бізнес-процеси, які база даних має підтримувати. Зона відповідальності:

- Збір вимог: збір вимог від замовників і кінцевих користувачів, щоб зрозуміти, які дані та функціонал потрібні.
- Переклад вимог у технічне завдання: створення детального ТЗ, що пояснює, як база даних буде підтримувати бізнес-процеси.
- Розробка специфікацій: підготовка документів, які пояснюють цілі, функціональні та нефункціональні вимоги до бази.
- Комунікація з командою: співпраця з розробниками, архітектором та іншими членами команди для забезпечення правильного розуміння вимог.
- Тестування на відповідність бізнес-вимогам: перевірка, чи відповідає фінальний продукт поставленим цілям та очікуванням замовника.

Розробники бази даних (Database Developers) – займаються безпосередньою розробкою та реалізацією структури бази даних, створюють таблиці, запити, процедури, функції тощо. Вони реалізують проекти, побудовані на схемах, розроблених архітектором, та оптимізують код для забезпечення ефективності та швидкості роботи бази. Зона відповідальності:

- Створення таблиць, індексів, ключів: розробка фізичної структури бази даних згідно з проектом архітектора.
- Розробка запитів і збережених процедур: створення запитів SQL для взаємодії з даними, а також збережених процедур і тригерів для обробки даних.

- Оптимізація запитів: поліпшення продуктивності бази даних за допомогою оптимізації SQL-запитів.
- Забезпечення цілісності даних: розробка рішень для забезпечення цілісності даних (обмеження, ключі, зв'язки).
- Інтеграція з іншими системами: створення механізмів для обміну даними з іншими системами, такими як ERP, CRM тощо.

Адміністратор бази даних (Database Administrator, DBA) – забезпечує налаштування і підтримку бази даних у робочому стані: керує правами доступу, стежить за продуктивністю, налаштовує резервне копіювання і відновлення даних. DBA також відповідає за безпеку бази та виправлення потенційних збоїв. Зона відповідальності:

- Налаштування та підтримка бази даних: встановлення та налаштування серверів бази даних, забезпечення стабільної роботи системи.
- Безпека бази даних: управління доступом, реалізація заходів безпеки, захист від атак.
- Резервне копіювання та відновлення: налаштування автоматичного резервного копіювання, відновлення після збоїв.
- Моніторинг продуктивності: відстеження швидкодії, виявлення та усунення вузьких місць.
- Технічна підтримка: вирішення проблем, що виникають під час роботи бази даних, надання підтримки користувачам.

Інженер з тестування (QA Engineer) – відповідальний за тестування функціоналу бази даних. QA Engineer розробляє тестові сценарії для перевірки коректності даних, продуктивності та безпеки бази. Виявлені помилки або недоліки передаються розробникам на доопрацювання. Зона відповідальності:

- Планування тестування: розробка плану тестування, що включає всі ключові аспекти бази даних.

- Розробка тестових сценаріїв: створення сценаріїв для функціонального, навантажувального, безпекового тестування.
- Автоматизоване тестування: розробка скриптів для автоматизованого тестування бази даних.
- Виявлення та документування помилок: знаходження помилок у системі, передача розробникам для виправлення.
- Підтвердження якості: перевірка відповідності бази даних вимогам замовника та загальним стандартам якості.

Фахівець з безпеки (Security Specialist) – забезпечує захист бази даних від зовнішніх загроз і внутрішніх вразливостей. Він відповідає за встановлення політик доступу, шифрування даних, розробку заходів протидії SQL-ін'єкціям та іншим типам атак. Зона відповідальності:

- Оцінка ризиків: виявлення потенційних загроз безпеці бази даних, проведення аудитів.
- Впровадження заходів захисту: розробка політик доступу, шифрування даних, впровадження технологій для запобігання SQL-ін'єкціям та іншим атакам.
- Моніторинг безпеки: постійне відстеження бази даних на предмет загроз, аналіз логів для виявлення підозрілої активності.
- Проведення тестувань на проникнення: імітація атак, щоб перевірити стійкість бази даних до зовнішніх загроз.
- Навчання команди: консультування інших учасників проекту щодо безпеки та найкращих практик захисту даних.

DevOps-інженер – підтримує процеси CI/CD (безперервної інтеграції та розгортання) для розробки та впровадження бази даних. DevOps забезпечує автоматизовані процеси деплою, налаштовує середовище розробки та тестування, щоб команда працювала в максимально ефективних умовах. Зона відповідальності:

- Автоматизація розгортання: налаштування середовища для швидкого розгортання оновлень бази даних.
- CI/CD-процеси: створення безперервної інтеграції та розгортання, щоб пришвидшити релізи та спростити оновлення.
- Налаштування середовищ: створення середовищ для розробки, тестування та продакшну, забезпечення їх відповідності вимогам проекту.
- Моніторинг і журналювання: налаштування систем для моніторингу продуктивності бази даних та збору логів, щоб швидко виявляти й усувати проблеми.
- Співпраця з DBA та розробниками: забезпечення ефективного потоку змін між командою розробників і адміністраторами бази даних.

Аналітик даних (Data Analyst) – залучається, якщо база даних містить значні обсяги інформації для аналізу та звітності. Аналітик готує інструменти для аналітики та допомагає створити інтерфейс для взаємодії з базою, щоб витягувати з неї потрібні бізнесові дані. Зона відповідальності:

- Аналіз даних: отримання, обробка та аналіз даних для створення звітів і аналітичних висновків.
- Підтримка бізнесу: надання інформації, яка може бути використана для ухвалення рішень, рекомендацій для бізнесу.
- Створення запитів: розробка складних SQL-запитів для аналізу даних, обробка великих обсягів даних.
- Візуалізація та звітність: побудова звітів, графіків і діаграм для надання інформації в зрозумілому вигляді.
- Розробка рекомендацій: допомога в оптимізації процесів на основі аналізу даних, пропозиції щодо покращення бази даних для збору потрібних бізнес-даних.

Кожна роль має свої унікальні завдання та обов'язки, що сприяє успіху проекту. Така чітка структура дозволяє ефективно організувати процес розробки

бази даних і гарантує, що всі критичні аспекти проекту будуть належним чином покриті (рис 3.3). Також було розроблено матрицю відповідальності для команди проекту (рис 3.4), завдяки чому кожен член команди знав свою зону відповідальності під час реалізації проекту.

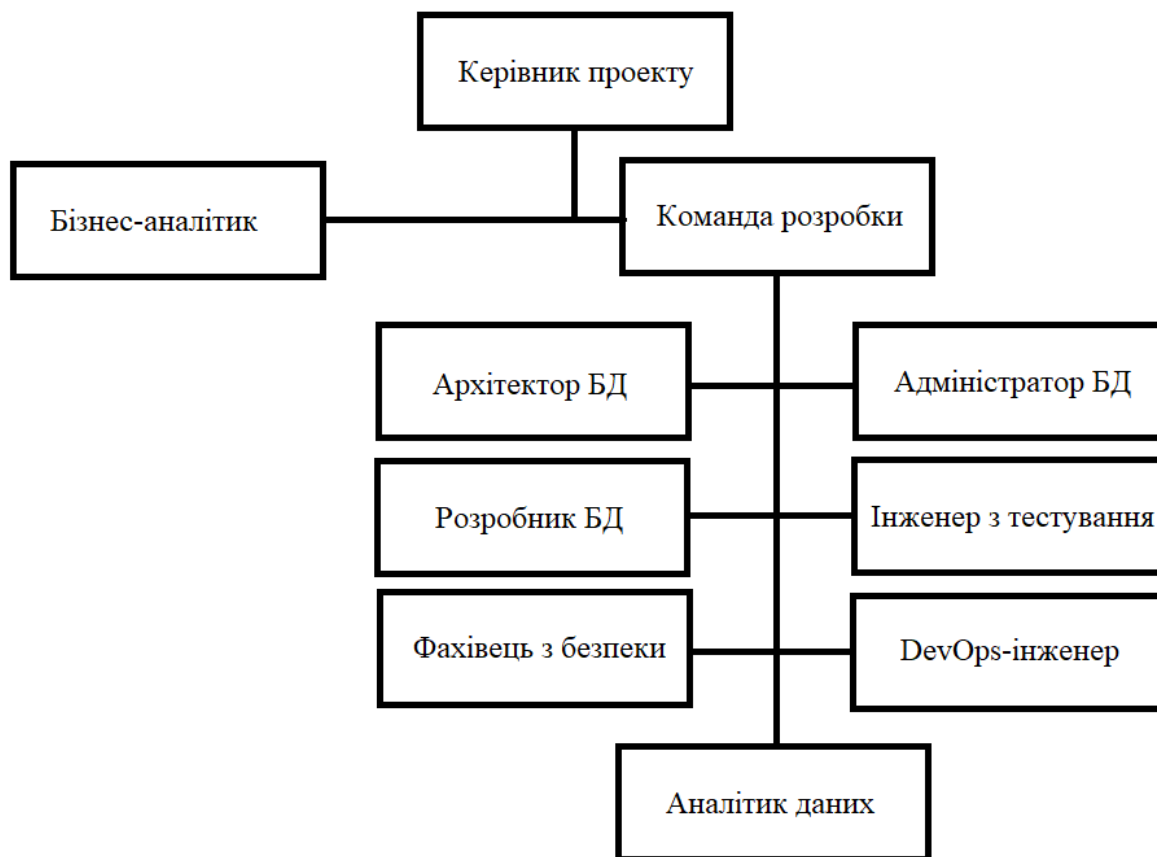


Рисунок 3.3 – Ієрархічна структура команди проекту

Назва роботи	Керівник	Адміністратор БД	Архітектор БД	Розробник БД	Інженер з тестування	DevOps-інженер	Аналітик даних	Бізнес аналітик	Фахівець з безпеки
Створення плану проекту	R	I	I	I	I	I	I	I	I
Формування вимог	R					C		C	
Аналіз ринку	I							R	
Проектування бази даних	I	C	R/A	C		R	C		C
Розробка бази даних	I	I	C	R/A		R	C		C
Нормалізація бази даних	I	R	C				C		C
Тестування	I			C	R/A	C			C
Завершення проекту	R	I	I	I	I	I	I	C	I

Рисунок 3.4 – Матриця відповідальності

3.1.4 Розрахунок бюджету

Для реалізації проекту організацією було виділено кошти у розмірі 300000 гривень, що включали витрати на необхідно програмне забезпечення для реалізації проекту, заробітну плату команди проекту та врахований бюджет на ризики (рис 3.5).

Ресурс	Заплановано	Фактично	Перевищення
Матеріальний	50000	50000	0
Трудовий	200000	190000	-10000
Бюджет на витрати	50000	64870	14870
Загальний бюджет	300000	304870	4870

Рисунок 3.5 – Розподіл бюджету

Через великі додаткові витрати бюджету на ризики збільшився загальний бюджет проекту, що призвело до змін в контракті проекту та встановленню нових домовленостей із замовником.

3.1.5 Засоби контролю та комунікації

Одним із важливих чинників інтеграції керівництва є спілкування, тобто комунікація людей у процесі їх спільної діяльності: обмін ідеями, думками, почуттями та інформацією. Без цього жодна організована група людей не може існувати.

Тому ефективне міжособистісне спілкування важливе для успіху проекту. Це обумовлюється двома факторами: по-перше, вирішення багатьох завдань керівництва базується на безпосередній взаємодії людей (начальник з підлеглим, підлеглий з іншим підлеглим) між різними подіями; по-друге, міжособистісне спілкування може бути найкращим способом обговорення та розв'язання проблем, що характеризуються невизначеністю та двозначністю. Тож менеджер повинен пам'ятати, що процес комунікації потребує особливої уваги.

Управління комунікаціями проекту передбачає виконання, планування та контроль за придбанням і розповсюдженням усієї інформації, яка стосується потреб зацікавлених сторін проекту.

Для проекту що розробляється була обрана методологія Scrum, що була описана раніше. В Microsoft Project було розроблено план проекту з відповідними датами, зазначено відповідальності за поставлені задачі. Там відбувався контроль продуктивності команди розробки.

Також кожен тиждень відбувався зв'язок із командою на платформі Microsoft Teams, для вирішення термінових питань та надання робітниками звітів щодо виконаної роботи. Завдяки даному методу була змога швидко реагувати на будь-які зміни в технічному завданні, та робити правки що надавав замовник.

3.2 Ведення проекту

Після планування та документування цілей та статуту проекту починається безпосередня робота. За описаними цілями, ризиками та враховуючи створену команду проекту було розроблено загальний план проекту за допомогою Microsoft Project (рис. 3.6).

задачі	Назва задачі	Длительность	Начало	Окончание	Предшественники	Названия ресурсов
	Розробка ПЗ	124 днів	Чт 12.10.23	Вт 02.04.24		
	Початковий етап	15 днів	Чт 12.10.23	Ср 01.11.23		
	Визначення цілей	2 днів	Чт 12.10.23	Пт 13.10.23		Керівник проекту; Системний аналітик
	Створення команди	5 днів	Пн 16.10.23	Пт 20.10.23	3	Керівник проекту
	Забезпечення робочого простору	8 днів	Пн 23.10.23	Ср 01.11.23	4	Менеджер по МТЗ[50%]; Оренда [1]
	Технологічне забезпечення	8 днів	Пн 23.10.23	Ср 01.11.23	4	Ліцензії[1]; Менеджер по МТЗ[50%]; ПЗ[1]; Техніка[1]
	Етап аналізу та планування	36 днів	Чт 02.11.23	Чт 21.12.23		
	Огляд ринку	8 днів	Чт 02.11.23	Пн 13.11.23	5;6	Керівник проекту; Системний аналітик
	Опитування потенційних користувачів	6 днів	Вт 14.11.23	Вт 21.11.23	8	Керівник проекту[50%]
	Аналіз конкурентів	14 днів	Вт 14.11.23	Пт 01.12.23	8	Керівник проекту[50%]
	Визначення технологій	10 днів	Пн 04.12.23	Пт 15.12.23	10;9	Системний аналітик
	Визначення бюджету	4 днів	Пн 18.12.23	Чт 21.12.23	11	Бухгалтер
	Етап розробки	57 днів	Пт 22.12.23	Пн 11.03.24		
	Створення прототипу	28 днів	Пт 22.12.23	Вт 30.01.24	12	Електронщик; Програміст
	Розробка інтерфейсу	12 днів	Ср 31.01.24	Чт 15.02.24	14	Програміст
	Додавання унікальних функцій	16 днів	Ср 31.01.24	Ср 21.02.24	14	Програміст
	Тестування	8 днів	Чт 22.02.24	Пн 04.03.24	16	Програміст
	Фінальні виправлення	5 днів	Вт 05.03.24	Пн 11.03.24	17	Програміст
	Завершальний етап	16 днів	Вт 12.03.24	Вт 02.04.24		
	Остаточні тести та вивчення	10 днів	Вт 12.03.24	Пн 25.03.24	18	Програміст
	Рекламна кампанія	5 днів	Вт 26.03.24	Пн 01.04.24	20	Керівник проекту; Маркетинг[1]
	Випуск ПЗ на ринок	1 день	Вт 02.04.24	Вт 02.04.24	21	Керівник проекту

Рисунок 3.6 – Розроблений план проекту

На вказані задачі було розплановано хід роботи, вказано трудові та матеріальні ресурси, а також надано приблизну вартість та ставку до кожного з них (рис 3.7).

Назва ресурса	тип	матеріал	назва	група	єдиниц	ставка
Керівник проекту	Трудовой		К		100%	250,00 €/ч
Системний аналітик	Трудовой		С		200%	120,00 €/ч
Менеджер по МТЗ	Трудовой		М		100%	180,00 €/ч
Програміст	Трудовой		П		300%	200,00 €/ч
Електронщик	Трудовой		Е		100%	150,00 €/ч
Бухгалтер	Трудовой		Б		100%	190,00 €/ч
Ліцензії	Материальный		Л			5 000,00 €
ПЗ	Материальный		П			3 500,00 €
Техніка	Материальный		Т			10 000,00 €

Рисунок 3.7 – Ресурси проекту

3.2.1 Розробка бази даних

Проектування баз даних — це ітераційний, багатоетапний процес прийняття обґрунтованих рішень у процесі аналізу інформаційної моделі предметної області, вимог до даних з боку прикладних програмістів і користувачів, синтезу логічних і фізичних структур даних, аналізу та обґрунтування вибору програмних і апаратних засобів.

Проектування баз даних відбувається в чотири етапи:

- Системний аналіз і словесний опис інформаційних об'єктів;
- Проектування інфологічної моделі або Концептуальне проектування;
- Логічне проектування Баз Даних;
- Фізичне проектування Баз Даних.

Системний аналіз передбачає мовний опис реальних об'єктів предметної області, визначення зв'язків між об'єктами, дослідження характеристик об'єктів і зв'язків. Результати дослідження використовуються при концептуальному проектуванні БД.

Для визначення складу і структури предметної області застосовуються або функціональний, або предметний підходи.

Функціональний підхід застосовує рух "від задач" і використовується у тих випадках, коли заздалегідь відомі функції майбутніх користувачів БД, а також відомі всі задачі, для інформаційних потреб яких створюються БД. В цьому випадку на основі виробничих документів, опитувань замовників можна чітко визначити мінімальний набір об'єктів предметної області та їх взаємозв'язок.

Предметний підхід застосовується у тому випадку, коли інформаційні потреби майбутніх користувачів чітко не визначені. В цьому випадку не можна чітко визначити мінімальний набір об'єктів предметної області. В опис предметної області включаються об'єкти та зв'язки, які є найбільш характерними та найбільш суттєвими для неї. БД називається предметною і може використовуватися для розв'язання задач, які заздалегідь не визначені.

У практичній діяльності використовується комплексний підхід, який з одного боку дозволяє розв'язувати конкретні інформаційні та функціональні задачі, а з іншого боку – враховує можливість додавання нових застосувань. У загальному випадку існує два підходи до проектування БД: низхідне проектування і висхідне проектування.

Низхідне проектування починається з визначення наборів даних, потім визначаються елементи даних для кожного з таких наборів. Цей процес включає в себе ідентифікацію різних типів сутностей і визначення атрибутів кожної сутності.

Низхідне проектування включає операції декомпозиції, що передбачає заміну вихідної множини відношень, що входять в схему БД, іншою множиною відношень, які є проєкціями вихідних відношень. Цей підхід рекомендується застосовувати у тих випадках, коли кількість, різноманітність та складність сутностей, зв'язків і транзакцій значна за розмірами. Найбільш поширеними моделями для цього проектування є моделі "сутність – зв'язок" (ER-моделі, Entity-Relationship model).

Вихідне проектування починається з виявлення елементів даних, які потім групуються в набори даних. Спочатку визначаються атрибути, які потім об'єднуються в сутності. Вихідне проектування включає операції синтезу, що передбачає виконання компоновки із заданої множини функціональних залежностей між об'єктами предметної області вихідних відношень схеми БД.

Цей підхід рекомендується застосовувати у тому випадку, якщо розробляється невелика БД з незначною кількістю об'єктів, атрибутів і транзакцій.

Етап концептуального проектування – з концептуального проектування починається створення концептуальної схеми БД, в основі якої лежить концептуальна модель даних. Концептуальна модель представляє загальний погляд на дані. Розрізняють два головних підходи до моделювання даних при концептуальному проектуванні:

- семантичні моделі;
- об'єктні моделі.

Семантичні моделі головну увагу приділяють структурі даних. Найбільш поширеною семантичною моделлю є модель "сутність – зв'язок" (Entity Relationship model, ER-модель). ER-модель складається із сутностей, зв'язків, атрибутів, доменів атрибутів, ключів. Моделювання даних відображає логічну структуру даних, так само, як блок-схеми алгоритмів відображають логічну структуру програми.

Об'єктні моделі головну увагу приділяють поведінці об'єктів даних і засобам маніпуляції даними. Головне поняття таких моделей – об'єкт, тобто сутність, яка має стан і поведінку. Стан об'єкта визначається сукупністю його атрибутів, а поведінка об'єкта визначається сукупністю операцій специфікованих для нього. Зближення цих моделей реалізується в розширеному ER-моделюванні (Extended Entity Relationship model, EER-модель).

Етап логічного проектування – логічне проектування полягає в створенні логічної моделі на основі вибраної моделі даних. На цьому етапі необхідно вже знати яка СУБД буде застосовуватися в системі (ієрархічна, мережна, реляційна,

об'єктно-орієнтована). Для перевірки вірності логічної моделі застосовується нормалізація. Крім того логічна модель перевіряється на умову забезпечення всіх транзакцій користувачів. Фізичне проектування полягає в описі засобів фізичної реалізації логічного проекту БД. Фізичні моделі визначають засоби розміщення даних в середовищі зберігання і засоби доступу до цих даних, які підтримуються на фізичному рівні. У процесі проектування визначається структура реляційної БД (склад таблиць, їх структура і логічні зв'язки). Структура таблиці визначається складом стовпців, типом даних і розмірами стовпців, ключами таблиці.

Етап фізичного проектування – фізичне проектування полягає в описі засобів фізичної реалізації логічного проекту БД. Фізичні моделі визначають засоби розміщення даних в середовищі зберігання і засоби доступу до цих даних, які підтримуються на фізичному рівні.

Командою розробників було створено таблиці, запити та схему даних, що відповідали критеріям та технічному завданню замовників. На рисунках 3.8 – 3.10 надано приклади проведеної роботи.

Имя поля	Тип данных
ID_pat	Счетчик
Name_pat	Короткий текст
Document_pat	Вложение
Disease_pat	Числовой
Adress_pat	Короткий текст
Birth_pat	Дата и время
Contact_pat	Короткий текст
TrustedName_pat	Короткий текст
TrustedContact_pat	Короткий текст
Hospital_pat	Числовой
Doctor_pat	Числовой
Treatment_pat	Длинный текст
DateHosp_pat	Дата и время
Department_pat	Короткий текст
HospRoom_pat	Числовой

Рисунок 3.8 – Вигляд конструктора таблиці БД

ID_pat	Doctor_pat	Name_pat	@	Disease_pat	Address_pat	Birth_pat	Contact_pat	TrustedName	TrustedCom	Hospital_pa	Treatment_	DateHosp_g	Department_pat	HospRoom_
1	9	Микола	(0)	1	Бабенна, 12	13.06.1980	+38(239)-573-57-35			4		26.11.2022	Інфекційне	667
2	1	Іван	(0)	2	Звиринецька, 1	12.04.1999	+38(687)-562-36-53			1		29.10.2022	Реабілітаційне	124
3	8	Анастасія	(0)	5	Луначарського, 4	24.04.2000	+38(687)-688-77-75			3				
4	6	Валерія	(0)	3	Освітня, 4	12.08.1999	+38(457)-537-75-77			3				
5	8	Дмитро	(0)	5		05.07.1999	+38(575)-757-57-57			3				
6	9	Олександр	(0)	1		09.12.1999	+38(527)-685-67-86			4		16.01.2022	Інфекційне	245
7	1	Жанна	(0)	2		07.11.1999	+38(658)-268-26-58			1		26.07.2022	Хірургічне	611
8	8	Євгеній	(0)	5		08.06.1999	+38(658)-268-62-84			3				
9	1	Михайло	(0)	2		04.07.1999	+38(464)-647-47-42			1		28.11.2022	Хірургічне	125
10	2	Андрій	(0)	4		09.12.1999	+38(246)-464-76-77			1				
11	2	Борис	(0)	4		09.02.1999	+38(246)-464-26-46			1				
12	2	Сніжанна	(0)	2		27.04.1999	+38(246)-785-75-53			1		13.06.2023	Реабілітаційне	231
13	6	Ігор	(0)	3		17.08.1999	+38(247)-275-75-66			3				
14	8	Богдан	(0)	5		12.12.1999	+38(246)-775-43-26			3				
15	9	Олена	(0)	1		31.07.1999	+38(248)-877-35-75			4		08.02.2021	Інфекційне	511
16	9	Валерій	(0)	1		25.11.1999	+38(246)-754-64-46			4		19.10.2020	Інфекційне	151
17	2	Ольга	(0)	4		14.09.1999	+38(246)-747-46-67			1				
18	6	Дарина	(0)	3		19.10.1999	+38(247)-287-42-66			3				
19	1	Мирослав	(0)	2		12.04.1999	+38(212)-747-35-74			1		31.07.2022	Хірургічне	135
20	8	Назар	(0)	5		26.12.1999	+38(984)-625-13-13			3				
21	2	Володимир	(0)	4		21.07.1999	+38(136)-467-43-51			1				
22	6	Юлія	(0)	3		18.01.1999	+38(254)-746-35-31			3				
23	6	Костянтин	(0)	3		23.02.1999	+38(151)-613-53-56			3				
24	2	Роман	(0)	2		24.03.1999	+38(131)-667-82-11			1		21.12.2022	Реабілітаційне	123
25	9	Віра	(0)	1		12.05.1999	+38(135)-667-43-63			4		28.08.2019	Інфекційне	513

Рисунок 3.9 – Вигляд однієї з таблиць БД

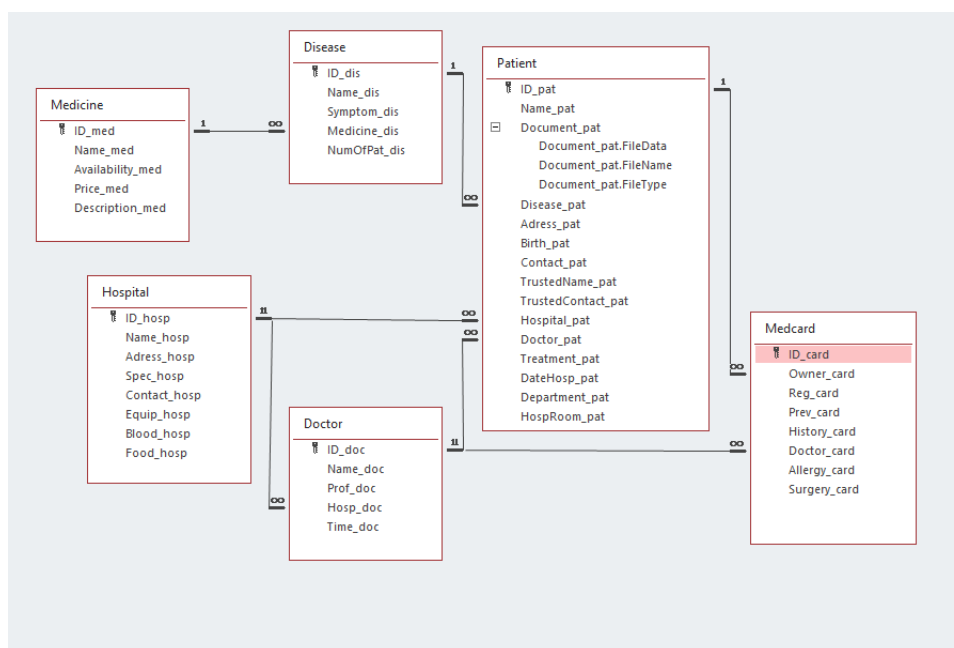


Рисунок 3.10 – Схема даних

3.2.2 Контроль роботи та звітності

Початковий план проекту було реалізовано за допомогою Microsoft Project, тому було прийняте рішення продовжувати роботу використовуючи даний продукт, так як він може бути інтегрований з усіма продуктами Microsoft, створюючи єдину область для контролю проекту.

Завдяки цьому кожен член команди проекту мав змогу ставити запитання, надавати та робити правки в поставлених задачах в найбільш короткі терміни. За методологією Scrumban проводилися щотижневі зустрічі проектної команди, та надавались звіти щодо поточного виконання робіт. Завдяки внутрішньому

функціоналу Microsoft Project були створені діаграми відстеження ходу роботи, а також звітність відносно всього проекту (рис. 3.11, рис 3.12).

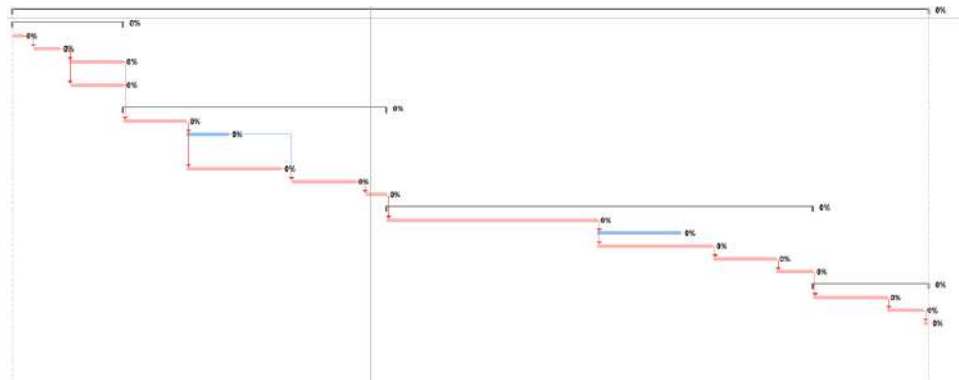


Рисунок 3.11 – Діаграма Ганта проекту, створена в Microsoft Project

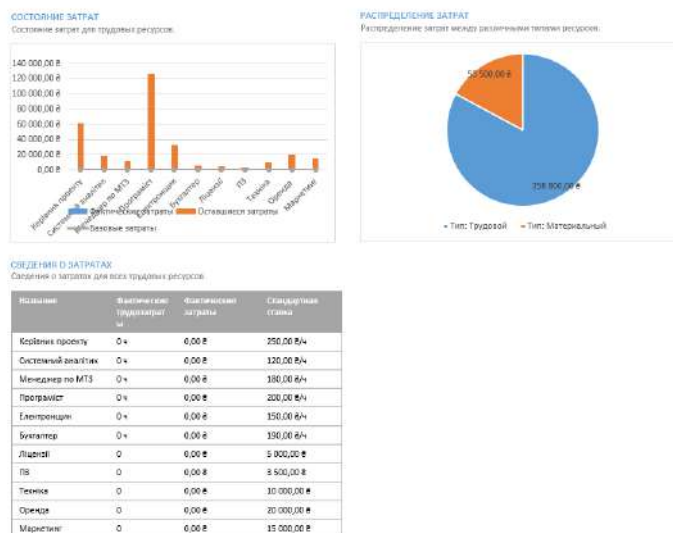


Рисунок 3.12 – Звітність, створена в Microsoft Project

3.2.3 Запобігання ризикам

Провівши аналіз можливих ризиків було виявлено які з них можуть трапитися найбільш ймовірно. Однак навіть маючи систему страхування від ризиків та план дій на випадок його появи, не завжди можна вирішити проблему без додаткових втрат.

Під час розробки бази даних один з ключових розробників, відповідальний за проектування таблиць, вимушений був переключити свою увагу на паралельний проєкт. Проектному менеджеру довелося змінювати графік робіт інших членів команди, та перерозподіляти їх зайнятість, через що термін

виконання робіт збільшився на два тижні від початкового плану, зображеного на рисунку 3.1 (рис. 3.13). Також було перевищено бюджет проєкту

➔	➤ Розробка ПЗ	136 днів	Чт 12.10.23	Чт 18.04.24		
➔	➤ Початковий етап	15 днів	Чт 12.10.23	Ср 01.11.23		
➔	Визначення цілей	2 днів	Чт 12.10.23	Пт 13.10.23		Керівник проєкту; Системни
➔	Створення команди	5 днів	Пн 16.10.23	Пт 20.10.23	3	Керівник проєкту
➔	Забезпечення робочого простору	8 днів	Пн 23.10.23	Ср 01.11.23	4	Менеджер по МТЗ[50%]; Орк
➔	Технологічне забезпечення	8 днів	Пн 23.10.23	Ср 01.11.23	4	Ліцензії[1]; Менеджер по МТ
➔	➤ Етап аналізу та планування	36 днів	Чт 02.11.23	Чт 21.12.23		
➔	Огляд ринку	8 днів	Чт 02.11.23	Пн 13.11.23	5;6	Керівник проєкту; Системни
➔	Опитування потенційних користувачів	6 днів	Вт 14.11.23	Вт 21.11.23	8	Керівник проєкту[50%]
➔	Аналіз конкурентів	14 днів	Вт 14.11.23	Пт 01.12.23	8	Керівник проєкту[50%]
➔	Визначення технологій	10 днів	Пн 04.12.23	Пт 15.12.23	10;9	Системний аналітик
➔	Визначення бюджету	4 днів	Пн 18.12.23	Чт 21.12.23	11	Бухгалтер
➔	➤ Етап розробки	69 днів	Пт 22.12.23	Ср 27.03.24		
➔	Створення прототипу	28 днів	Пт 22.12.23	Вт 30.01.24	12	Електронщик; Програміст
➔	Розробка інтерфейсу	12 днів	Ср 31.01.24	Чт 15.02.24	14	Програміст
➔	Додавання унікальних функцій	16 днів	Пт 16.02.24	Пт 08.03.24	15	Програміст
➔	Тестування	8 днів	Пн 11.03.24	Ср 20.03.24	16	Програміст
➔	Фінальні виправлення	5 днів	Чт 21.03.24	Ср 27.03.24	17	Програміст
➔	➤ Завершальний етап	16 днів	Чт 28.03.24	Чт 18.04.24		
➔	Остаточні тести та впровадження	10 днів	Чт 28.03.24	Ср 10.04.24	18	Програміст
➔	Рекламна кампанія	5 днів	Чт 11.04.24	Ср 17.04.24	20	Керівник проєкту; Маркетинг
➔	Випуск ПЗ на ринок	1 день	Чт 18.04.24	Чт 18.04.24	21	Керівник проєкту

Рисунок 3.13 – План розробки після корегування

3.3 Завершення проєкту

Кожен проєкт повинен закінчитися, і саме про це йде завершення проєкту на останній фазі життєвого циклу проєкту. Вся суть проєкту полягає в тому, щоб доставити те, що було обіцяно. Доставляючи все, що було сказано, проєктний менеджер переконується, що всі зацікавлені сторони задоволені і всі критерії прийняття були дотримані. Завершення проєкту часто є найбільш запущеним етапом життєвого циклу проєкту.

Ключовими видами діяльності в завершенні проєкту є збір записів проєкту; поширення інформації для формалізації прийняття продукту, послуги або

проекту; і виконання закриття проекту. Мені, як проєктному менеджеру, було потрібно переглянути проєктні документи, щоб переконатися, що вони актуальні. Наприклад, можливо, були реалізовані деякі запити на зміну обсягу, які змінили деякі характеристики кінцевого продукту. Інформація про проєкт, яка була зібрана на цьому етапі, повинна відображати характеристики та технічні характеристики кінцевого продукту. Також необхідно оновити призначення ресурсів. Деякі члени команди прийдуть і пройдуть хід проєкту. Потрібно ще раз перевірити, що всі ресурси і їх ролі та обов'язки відзначені.

Після того, як результати проєкту будуть задокументовані, необхідно запитати офіційне прийняття від зацікавлених сторін або замовника. Вони зацікавлені в тому, щоб знати, чи продукт або послуга проєкту відповідає цілям проєкту, поставленим для виконання. Якщо документація актуальна, проєктний менеджер буде мати під рукою результати проєкту, щоб поділитися з ними.

3.3.1 Закриття контракту

Контракти завершуються так само, як проєкти закінчуються. Закриття контракту стосується завершення та врегулювання умов договорів, орендованих для проєкту. Він підтримує процес завершення проєкту, оскільки процес закриття контракту визначає, чи виконана робота, описана в контрактах, точно та задовільно.

В контракті даного проєкту було зазначено його терміни та виділений бюджет. Через ризик, що виникнув, було перевищено наданий бюджет, тому керівник проєкту разом із замовником розробляли нові домовленості щодо витрат та окупності проєкту.

Закриття контракту оновлює записи проєкту, деталізуючи кінцеві результати роботи над проєктом. Основні умови контракту, а саме терміни виконання робіт та реалізація технічного завдання проєкту, були дотримані і після вирішення питань перевищення бюджету контракт було закрито.

3.3.2 Звільнення проєктної команди та оцінки після проєкту

Звільнення членів команди проєкту не є офіційним процесом. Однак слід зазначити, що по завершенні проєкту буде звільнено членів проєктної команди, а вони повернуться до своїх функціональних менеджерів або отримають призначення на новий проєкт. Необхідно тримати їх керівників, або інших менеджерів проєктів, на зв'язку в міру наближення до завершення проєкту, щоб вони встигли адекватно спланувати повернення своїх співробітників. Дайте їм знати на кілька місяців раніше часу, як виглядає графік і як скоро вони можуть планувати використання своїх співробітників на нових проєктах. Це дає іншим менеджерам можливість розпочати планування діяльності та планувати дати діяльності.

Команда даного проєкту була зібрана задля розвитку власних здібностей та отримання досвіду роботи. Було проведено заключну співбесіду із керівником проєкту, під час якої було вказано сильні і слабкі сторони нашої команди. Наприклад команда завжди швидко реагувала на зміни в технічному завданні та робила правки, що вносив замовник, проте не було враховано змін в навантаженні на учасників команди, через що було змінено терміни виконання завдань та перевищено бюджет проєкту.

3.4 Висновки

На прикладі розглянуто процес управління проєктом розробки реляційних баз даних для висококонкурентних додатків в сфері медицини. На основі обраних тенденцій, методологій та інструментів управління проєктами було розроблено план та цілі проєкту. Розглянуто та описано можливі ризики, розраховано бюджет та надано реалізацію проєкту.

Досліджено ризики що виникли під час проєкту, та надано шлях їх вирішення, враховуючи специфіку проєкту.

Визначено основні кроки завершення проєкту, такі як закриття контрактів та робота над помилками, що дозволило команді проєкту виправити свій підхід до виконання завдань та реалізувати зміни в наступних проєктах.

ЗАГАЛЬНІ ВИСНОВКИ

Під час написання кваліфікаційної роботи магістра було розглянуто процес управління проектом розробки реляційної бази даних для висококонкурентних додатків в області медицини.

Досліджено особливості моделей та архітектури баз даних. Відповідно до технічного завдання проєкту обрано клієнт-серверну модель, що була затверджена організацією та замовником проєкту. Серед архітектур баз даних була обрана реляційна, як найбільш поширена в сфері медицини завдяки можливості пов'язувати між собою значну кількість таблиць з даними.

Розглянуто основні принципи та етапи управління проектом, визначено необхідні для проєктного менеджера компетенції та знання, що сприятимуть покращенню процесу управління проєктами.

Визначено основні тенденції управління проектом, на основі яких буде сформований проєкт управління розробкою реляційних баз даних для висококонкурентних додатків.

За результатами досліджень для управління даним проектом було обрано філософію Agile, завдяки її гнучкості. Враховуючи досвід команди проєкту та організації, на яку вона працює, можливість швидко вносити і реагувати на зміни в технічному завданні проєкту стала ключовим аспектом, через який було обрано саме Agile.

Слід зазначити що в проєкті використовувалася не безпосередньо Agile, а методологія що поєднує два її фреймворки Scrum і Kanban – Scrumban. Ця методологія була обрана через свою зрозумілість і легкість реалізації. Подібність до Kanban-дошок було реалізовано завдяки MS Project, а щотижневі спринти та зв'язок між усіма зацікавленими сторонами відбувався через MS Teams.

При виборі інструменту для управління проєктами менеджер проєкту надав перевагу MS Project, так як вище керівництво організації користується середою Microsoft. Окрім самого MS Project також використовувалася платформа MS Teams, де проводилися щотижневі спринти та проходила основна комунікація

команди проєкту, а також дозволяла імпортувати таблиці з MS Project, що дозволяло робити швидкі правки в проєкті.

На прикладі розглянуто процес управління проєктом розробки реляційних баз даних для висококонкурентних додатків в сфері медицини. На основі обраних тенденцій, методологій та інструментів управління проєктами було розроблено план та цілі проєкту. Розглянуто та описано можливі ризики, розраховано бюджет та надано реалізацію проєкту.

Досліджено ризики що виникли під час проєкту, та надано шлях їх вирішення, враховуючи специфіку проєкту.

Визначено основні кроки завершення проєкту, такі як закриття контрактів та робота над помилками, що дозволило команді проєкту виправити свій підхід до виконання завдань та реалізувати зміни в наступних проєктах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Блага Н. В. Управління проєктами : навч. посібник. Львів : Львівський державний університет внутрішніх справ, 2021. 152 с
2. Анісімов А.В. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Анісімов А.В., Кулябко П.П. – Київ. – 2017. – 110 с.
3. Методичні вказівки до виконання курсової роботи по дисципліни "Організація баз даних і баз знань". / Укладач В.О. Нелюбов. – Ужгород: Видавничий центр ЗакДУ, 2012. – 58 с.
4. Система управління базами даних Access. Навчальний посібник з курсу «Організація баз даних і баз знань»/ Укладач В.О. Нелюбов. – Ужгород: Редакційно-видавничій відділ ЗакДУ, 2011. – 62 с.
5. Гайна Г.А. Основи проєктування баз даних: Навчальний посібник. – К.: КНУБА, 2005. – 204 с.
6. Андрєєва Т. Є. Проектний менеджмент як засіб досягнення мети підприємства / Т.Є. Андрєєва, Т.Е. Петровська, Т.С. Титар // Вісник економіки транспорту і промисловості. – 2011. – № 34. – С. 364-370.
7. Батенко Л. П. Управління проєктами: навч. посіб. / Л. П. Батенко, О. А. Загородніх, В. В. Ліщинська. – К.: КНЕУ, 2003. – 231 с.
8. Безверхнюк Т. М. Технології проектного менеджменту в регіональному управлінні програмами і проєктами: розробка та впровадження / Т.М. Безверхнюк // Сучасна регіональна політика: освіта, наука практика. Матеріали підсумкової науково-практичної конференції за міжнародною участю. – 28 жовтня 2011. – Том II. – С. 140-142.
9. Бушуев С. Д. Управление проєктами: Основы профессиональных знаний и система оценки компетентности проектных менеджеров / С. Д. Бушуев, Н. С. Бушуева // National Competence Baseline, NCB UA Version 3.0. – К.: ІРІДУМ, 2006. – 208 с.

10. ДеМарко Т. Deadline. Роман об управлении проектами [Текст] / Том ДеМарко. – М. : Вершина, 2006. – 288 с.
11. Креативные технологии управления проектами и программами: Монография / С.Д. Бушуев, Н.С. Бушуева, И.А. Бабаева [и др.]. – К.: СаммитКнига, 2010. – 768 с.
12. Мазур И. И. Управление проектами: учебное пособие / И.И. Мазур, В.Д. Шапиро, Н.Г. Ольдерогге. – 2-е изд. – М.: Омега-Л, 2004. – 664 с.
13. Морозов В. В. Інформаційні системи і технології в управлінні проектами. Планування проектів у MS Project: навчальний посібник / В.В. Морозов, О.Б. Данченко, О.І. Шаров. – К. : Університет економіки та права "КРОК", 2011. – 167 с.
14. Морозов В. В. Компоненти управління проектами: навчальний посібник для самостійної роботи студентів магістратури по спеціальності 8.000003 "Управління проектами" / В.В. Морозов. – К.: Університет економіки та права "КРОК", 2005. – 62 с.
15. Trello – это бесплатный, гибкий и наглядный способ организовать что угодно с кем угодно [Электронный ресурс] // Офіційний сайт. – Режим доступу: <https://trello.com/>
16. Move work forward: Asana is the easiest way for teams to track their work and get results [Электронный ресурс] // Офіційний сайт. – Режим доступу: <https://asana.com/>
17. Atlassian JIRA Software [Электронный ресурс] // Офіційний сайт. – Режим доступу: <https://ru.atlassian.com/software/jira>
18. Microsoft Office Project Portfolio Server [Электронный ресурс] // Офіційний сайт. – Режим доступу: <https://products.office.com/ukua/project/project-and-portfolio-management-software>
19. Project Management Institute (PMI) [Электронный ресурс] // Офіційний сайт. – Режим доступу: <https://www.pmi.org/>

20. Top 10 best project management software [Электронный ресурс] // Creative Bloq.
– 2016. – Режим доступа: <http://www.creativebloq.com/software/best-project-management-71515632>

ДОДАТКИ

Управління проектом розробки реляційних баз даних для висококонкурентних додатків

Керівник: д.т.н., професор Веренич О.В.
Виконавець: Солодей Н.І.

2024

[1]

Об'єкт, предмет, методи та мета дослідження

Об'єкт дослідження: підхід до управління проектом розробки баз даних для автоматизованої обробки персоналізованої медичної інформації.

Предмет дослідження: методи та засоби управління проектами в сфері ІТ, архітектура баз даних в додатках із високим навантаженням на бази даних.

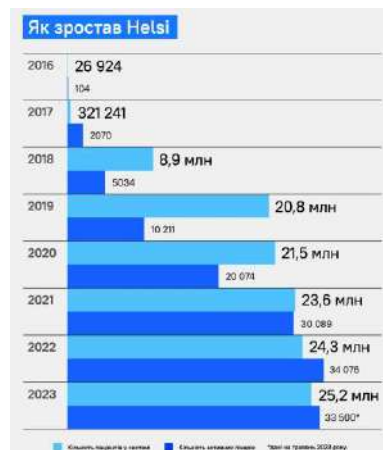
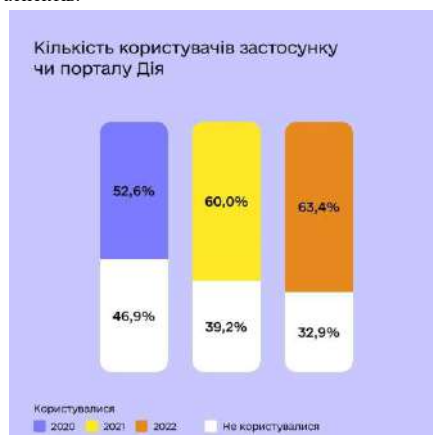
Методи дослідження: аналіз літературних джерел (для визначення відомих рішень в даній сфері), метод порівняння для виділення переваг та недоліків розробленого рішення порівняно з наявними

Мета і задачі дослідження: метою даної роботи є аналіз існуючих методологій управління проектами, та дослідження використання сучасних методів управління проектів на прикладі продукту, що розробляється.

[2]

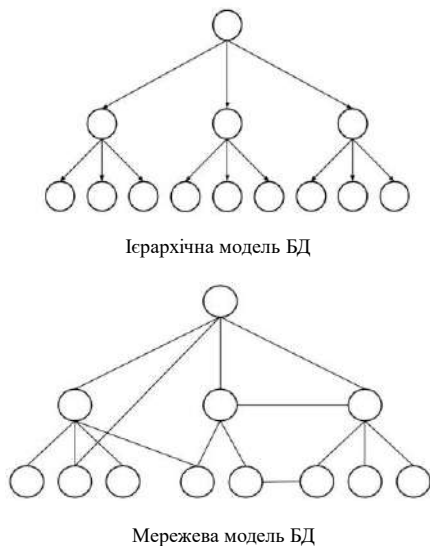
Актуальність дослідження

В час цифровізації суспільства на ринок виходять додатки з високим навантаженням на бази даних, наприклад Дія, Helsi, Резерв+ тощо. Успіх впровадження подібних додатків багато в чому залежить від процесу управління проектом: постановки цілей, планування ресурсів, аналізу ризиків та багатьох інших аспектів.



[3]

Аналіз предметної області



Принципи реляційних баз даних

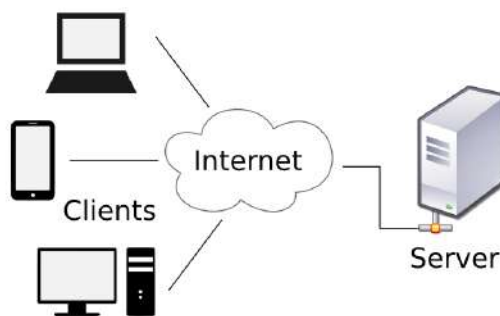
- Реляційні таблиці
- Рядки (кортежі)
- Стовпці (атрибути)
- Ключі
- Зв'язки
- SQL (Structured Query Language)
- Нормалізації
- Транзакції

[4]

Архітектура додатків з використанням баз даних

Файл-серверна архітектура

Файл-серверна архітектура централізованих БД з мережевим доступом передбачає функціонування одного з комп'ютерів мережі в якості виділеного сервера, на якому зберігаються файли централізованої БД



Клієнт-серверна архітектура

Сучасні промислові БД реалізовані на платформі клієнт-сервер, що дає можливість зберігати на сервері саме базу даних, а засоби представлення даних та алгоритми їх обробки (бізнес-логіку) перенести до клієнтської частини системи.

[5]

Визначення управління проектами



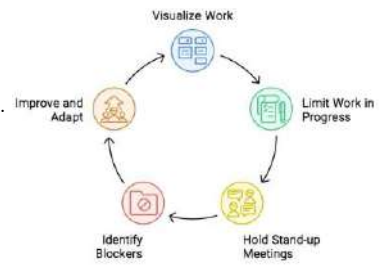
[6]

Опис проблеми та шляхи вирішення Методології та інструменти управління проектами



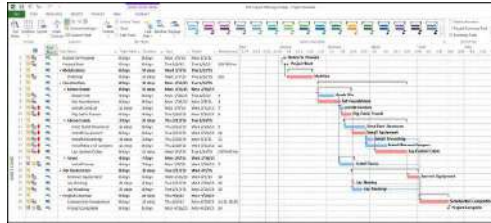
Методологія Agile

Для проекту було обрано методологію Agile із використанням фреймворку, що поєднує в собі підходи Scrum та Kanban. Основна команда організації працює із MS Project, тому серед інших інструментів для управління проектами було обрано саме його



Фреймворк Scrumban

Інтерфейс MS Project



[7]

Автоматизована обробка персоналізованої медичної інформації

Постановка цілей

Specific (Конкретна): Розробити базу даних для зберігання та управління інформацією про клієнтів, включаючи їхні контакти, історію хвороб та статусу платежів.

Measurable (Вимірювана): База даних повинна включати таблиці з відповідними зв'язками та підтримувати обробку до 10000 записів щодня. Один лікар може обслуговувати максимум 2000 пацієнтів. Така кількість пацієнтів рекомендує Міністерство охорони здоров'я.

Achievable (Досяжна): Команда з семи розробників завершить розробку протягом чотирьох місяців.

Relevant (Актуальна): Проект спрямований на покращення клієнтського обслуговування та оптимізацію внутрішніх бізнес-процесів.

Time-bound (Обмежена в часі): База даних має бути завершена до 17 липня 2024 року.

SWOT-аналіз проекту

Strengths (Сильні сторони)

- Технічна експертиза: Досвідчена команда розробників із знаннями в SQL, NoSQL, та сучасних технологіях баз даних.
- Гнучкість в адитації: Можливість швидкого внесення змін у базу даних відповідно до змін бізнес-процесів.
- Висока продуктивність: Використання оптимізованих алгоритмів для швидкої обробки великих обсягів даних.
- Інтеграція з іншими системами: Простота інтеграції бази даних з існуючими веб-додатками, ERP та CRM системами.

Weaknesses (Слабкі сторони)

- Високі початкові витрати: Витрати на розробку, налаштування інфраструктури та навчання персоналу.
- Залежність від технічної команди: Обмежена кількість людей, які мають глибокі знання про архітектуру бази даних.
- Ризик затримок: Можливі затримки через технічні проблеми або нечіткі вимоги на початковому етапі.
- Складність у масштабуванні: У випадку швидкого зростання бізнесу можуть виникнути труднощі з масштабуванням бази даних.

Opportunities (Можливості)

- Автоматизація бізнес-процесів: База даних дозволить автоматизувати рутинні завдання, підвищуючи ефективність роботи.
- Покращення клієнтського обслуговування: Зодакш швидкому доступу до актуальної інформації про клієнтів.
- Вихід на нові ринки: Розширення бази даних під нові регіони або додаткові бізнес-линії.
- Використання сучасних технологій: Інтеграція з хмарними сервісами, аналітичними платформами, AI/ML для прогнозування та аналізу даних.

Threats (Загрози)

- Кіберзагрози: Ризик несанкціонованого доступу, витоку даних або атак на базу даних.
- Зміни у законодавстві: Необхідність відповідності нормативним актам (наприклад, GDPR).
- Конкуренція: Інші компанії можуть запровадити аналогічні або кращі системи.
- Залежність від постачальників: У разі використання сторонніх сервісів (хмарні рішення, аплети ПЗ) можуть виникнути ризики, пов'язані з їх надійністю та цінами.

[8]

Оцінка та аналіз ризиків

№	Назва ризику	Ймовірність виникнення	Вплив на хід проекту
1	Недостатня продуктивність системи	0.2	0.8
2	Неправильне моделювання даних	0.2	0.6
3	Невідповідність вибраної технології	0.1	0.9
4	Невірне визначення вимог	0.5	0.5
5	Недотримання терміна	0.4	0.6
6	Перевищення бюджету	0.3	0.2
7	Несанкціонований доступ	0.5	0.9
8	Відсутність резервного копіювання	0.2	0.7
9	Невідповідність нормативним вимогам	0.2	0.3
10	Недостатня кваліфікація персоналу	0.2	0.4

Одним з основних завдань, які розв'язують у межах управління проектами, є управління ризиками проєктної діяльності, або управління ризиками проєкту. Відомі чотири основних методи управління ризиками: **скасування, запобігання та контролювання, страхування та поглинання ризиків.**

[9]

Управління якістю

Одним із найважливіших параметрів проєкту, якими можливо і необхідно управляти, поряд з часом, вартістю і ресурсами є його якість. Якість в сучасних економічних умовах виступає невід'ємною цільовою характеристикою кожного проєктного продукту. В ході реалізації проєкту було застосовано стандарти якості із серії ISO 9000. Задля контролю якості керівник проєкту дотримувався процедури управління документацією, створення беклогів проєкту та загальної культури організації.

Створення команди проєкту

Керівник проєкту (Project Manager) – відповідає за загальну координацію проєкту: від складання плану, управління бюджетом та ресурсами до контролю термінів і якості виконання завдань.

Архітектор бази даних (Database Architect) – відповідальний за загальну архітектуру бази даних, включно з вибором структури, типом даних, їх нормалізацією та визначенням взаємозв'язків між сутностями.

Бізнес-аналітик (Business Analyst) – збирає вимоги від замовника або користувачів, документує їх та перетворює на технічні вимоги.

Розробники бази даних (Database Developers) – займаються безпосередньо розробкою та реалізацією структури бази даних, створюють таблиці, запити, процедури, функції тощо.

Адміністратор бази даних (Database Administrator, DBA) – забезпечує налаштування і підтримку бази даних у робочому стані: керує правами доступу, стежить за продуктивністю, налаштовує резервне копіювання і відновлення даних.

Інженер з тестування (QA Engineer) – відповідальний за тестування функціоналу бази даних.

Фахівець з безпеки (Security Specialist) – забезпечує захист бази даних від зовнішніх загроз і внутрішніх вразливостей.

DevOps-інженер – підтримує процеси CI/CD (безперервної інтеграції та розгортання) для розробки та впровадження бази даних.

Аналітик даних (Data Analyst) – залучається, якщо база даних містить значні обсяги інформації для аналізу та звітності.

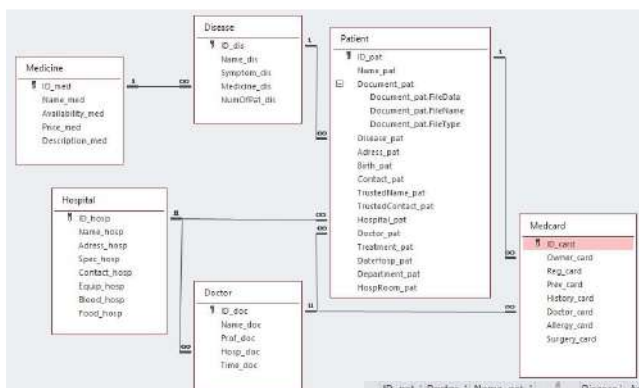


[10]

Матриця відповідальності

Назва роботи	Керівник	Адміністратор БД	Архітектор БД	Розробник БД	Інженер з тестування	DevOps-інженер	Аналітик даних	Бізнес-аналітик	Фахівець з безпеки
Створення плану проєкту	R	I	I	I	I	I	I	I	I
Формування вимог	R					C		C	
Аналіз ринку	I							R	
Проектування бази даних	I	C	R/A	C		R	C		C
Розробка бази даних	I	I	C	R/A		R	C		C
Нормалізація бази даних	I	R	C				C		C
Тестування	I			C	R/A	C			C
Завершення проєкту	R	I	I	I	I	I	I	C	I

[11]



Довідники

ID	Name_med	A	Price_me	Description_m
1	Стрептоцид	⊞	450,00 ₪	Антибіотик
2	Солпаден-ані	⊞	300,00 ₪	Болезаспінійли
3	Парацетамол	⊞	200,00 ₪	Проти жару
4	Терафлю	⊞	250,00 ₪	Від застуди
5	Креон	⊞	300,00 ₪	Покращення тр

Розробка бази даних

Ім'я поля	Тип даних
ID_pat	Счетчик
Name_pat	Короткий текст
Document_pat	Включення
Disease_pat	Числовий
Address_pat	Короткий текст
Birth_pat	Дата і час
Contact_pat	Короткий текст
TrustedName_pat	Короткий текст
TrustedContact_pat	Короткий текст
Hospital_pat	Числовий
Doctor_pat	Числовий
Treatment_pat	Длинний текст
DateHosp_pat	Дата і час
Department_pat	Короткий текст
HospRoom_pat	Числовий

ID_pat	Doctor	Name_pat	A	Disease	Address_pat	Birth_pat	Contact	DateHosp	Department	Hosp	HospRoom
1	0	Микола	Ан	1	Бабенка, 12	13.05.1980	+38(259)	26.11.2022	Інфекційне	4	657
2	1	Іван	Ан	2	Заїрнецька, 2	12.04.1999	+38(283)	29.10.2022	Реабілітаційне	1	124
3	2	Анастасія	Ан	3	Луначарського	24.04.2000	+38(567)			3	
4	3	Валерія	Ан	4	Сосни, 4	12.08.1999	+38(457)			3	
5	4	Владимир	Ан	5		05.07.1999	+38(575)			3	
6	5	Олександр	Ан	1		09.12.1999	+38(527)	16.01.2022	Інфекційне	4	245
7	1	Ліана	Ан	2		07.11.1999	+38(658)	26.07.2022	Хірургічне	1	612
8	3	Євгеній	Ан	5		08.06.1999	+38(658)			3	
9	1	Мініта	Ан	2		04.07.1999	+38(464)	28.11.2022	Хірургічне	1	125
10	2	Андрій	Ан	4		09.12.1999	+38(246)			1	
11	2	Ворис	Ан	4		09.02.1999	+38(246)			1	
12	2	Сніжанна	Ан	2		27.04.1999	+38(246)	13.06.2023	Реабілітаційне	1	282
13	6	Ігор	Ан	3		17.08.1999	+38(247)			3	
14	8	Євген	Ан	5		12.12.1999	+38(246)			3	
15	9	Олена	Ан	1		31.07.1999	+38(248)	08.02.2021	Інфекційне	4	512

[12]

Управління проектом

Назва етапу	Діагностика	Начало	Скінчено	Продовження	Назва ресурсу
Розробка ПЗ					
Початковий етап					
Визначення цілей	2 дні	Чт 12.10.23	Вт 02.04.24		
Створення команди	2 дні	Пн 10.10.23	Пн 13.10.23		Керівник проекту, Системний аналітик
Забезпечення робочого простору	2 дні	Пн 23.10.23	Ср 01.11.23		Менеджер по МТЗ [50%], Юрєнда [50%]
Технологічне забезпечення	8 днів	Пн 23.10.23	Ср 01.11.23		Ліцензія [1], Менеджер по МТЗ [50%]
Етап аналізу та планування					
Огляд ринку	8 днів	Чт 02.11.23	Пн 13.11.23	5,6	Керівник проекту, Системний аналітик
Опитування потенційних користувачів	8 днів	Вт 14.11.23	Вт 21.11.23	8	Керівник проекту [50%]
Аналіз конкурентів	14 днів	Вт 14.11.23	Пн 01.12.23	8	Керівник проекту [50%]
Визначення технологій	10 днів	Пн 04.12.23	Пт 15.12.23	10,9	Системний аналітик
Визначення бюджету	4 дні	Пн 18.12.23	Чт 21.12.23	11	Бухгалтер
Етап розробки					
Створення прототипу	28 днів	Пт 22.12.23	Вт 30.01.24	12	Електронщик, Програміст
Розробка інтерфейсу	12 днів	Ср 31.01.24	Чт 02.02.24	14	Програміст
Додавання функцій	16 днів	Ср 31.01.24	Ср 21.02.24	14	Програміст
Тестування	8 днів	Чт 22.02.24	Пн 04.03.24	16	Програміст
Фінальне випробування	5 днів	Вт 05.03.24	Пн 11.03.24	17	Програміст
Завершальний етап					
Остаточні тести та в	10 днів	Вт 12.03.24	Пн 25.03.24	18	Програміст
Рекламна кампанія	5 днів	Вт 26.03.24	Пн 01.04.24	20	Керівник проекту, Маркетинг [1]
Випуск ПЗ на ринок	1 день	Вт 02.04.24	Вт 02.04.24	21	Керівник проекту

Назва етапу	Діагностика	Начало	Скінчено	Продовження	Назва ресурсу
Розробка ПЗ					
Початковий етап					
Визначення цілей	2 дні	Чт 12.10.23	Вт 02.04.24		
Створення команди	2 дні	Пн 10.10.23	Пн 13.10.23		Керівник проекту, Системний аналітик
Забезпечення робочого простору	2 дні	Пн 23.10.23	Ср 01.11.23		Менеджер по МТЗ [50%], Юрєнда [50%]
Технологічне забезпечення	8 днів	Пн 23.10.23	Ср 01.11.23		Ліцензія [1], Менеджер по МТЗ [50%]
Етап аналізу та планування					
Огляд ринку	8 днів	Чт 02.11.23	Пн 13.11.23	5,6	Керівник проекту, Системний аналітик
Опитування потенційних користувачів	8 днів	Вт 14.11.23	Вт 21.11.23	8	Керівник проекту [50%]
Аналіз конкурентів	14 днів	Вт 14.11.23	Пн 01.12.23	8	Керівник проекту [50%]
Визначення технологій	10 днів	Пн 04.12.23	Пт 15.12.23	10,9	Системний аналітик
Визначення бюджету	4 дні	Пн 18.12.23	Чт 21.12.23	11	Бухгалтер
Етап розробки					
Створення прототипу	28 днів	Пт 22.12.23	Вт 30.01.24	12	Електронщик, Програміст
Розробка інтерфейсу	12 днів	Ср 31.01.24	Чт 02.02.24	14	Програміст
Додавання функцій	16 днів	Ср 31.01.24	Ср 21.02.24	14	Програміст
Тестування	8 днів	Чт 22.02.24	Пн 04.03.24	16	Програміст
Фінальне випробування	5 днів	Вт 05.03.24	Пн 11.03.24	17	Програміст
Завершальний етап					
Остаточні тести та в	10 днів	Вт 12.03.24	Пн 25.03.24	18	Програміст
Рекламна кампанія	5 днів	Вт 26.03.24	Пн 01.04.24	20	Керівник проекту, Маркетинг [1]
Випуск ПЗ на ринок	1 день	Вт 02.04.24	Вт 02.04.24	21	Керівник проекту

Ресурс	Тип	Вартість	Відсоток	Всього	
Керівник проекту	Трудовий	100%	250,00 €	€	
Системний аналітик	Трудовий	С	200%	120,00 €	€/ч
Менеджер по МТЗ	Трудовий	М	100%	180,00 €	€/ч
Програміст	Трудовий	П	300%	200,00 €	€/ч
Електронщик	Трудовий	Е	100%	150,00 €	€/ч
Бухгалтер	Трудовий	Б	100%	190,00 €	€/ч
Ліцензія	Матеріальний	Л		5 000,00 €	
ПЗ	Матеріальний	П		3 800,00 €	
Техніка	Матеріальний	Т		10 000,00 €	

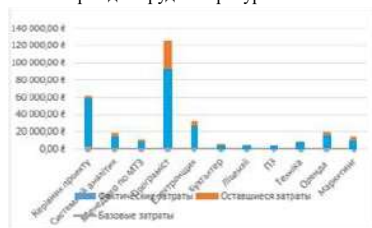
Ресурс	Заплановано	Фактично	Перевищення
Матеріальний	50000	50000	0
Трудовий	200000	190000	-10000
Бюджет на витрати	50000	64870	14870
Загальний бюджет	300000	304870	4870

[13]

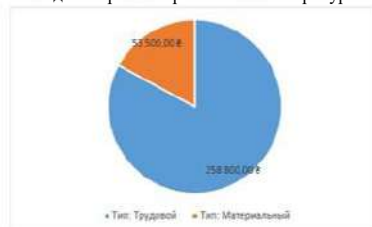
Контроль роботи та звітності

Огляд витрат ресурсів

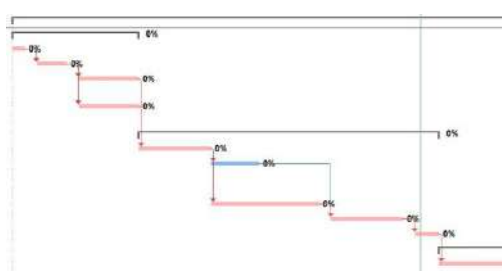
Стан витрат для трудових ресурсів



Розподіл витрат між різними типами ресурсів



Діаграма Ганта проекту, створена в Microsoft Project



Діаграма Ганта (також стрічкова діаграма, графік Ганта) — діаграма, яка використовується для ілюстрації плану, графіка робіт за будь-яким проектом, вона є одним з засобів планування та управління проектами. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщені по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. Створена діаграма дозволила оцінити терміни та порядок виконання робіт, а також швидко реагувати на зміни в проведеному проекті.

[14]

Висновки

Під час написання кваліфікаційної роботи магістра було розглянуто процес управління проектом розробки реляційної бази даних для висококонкурентних додатків в області медицини.

Досліджено особливості моделей та архітектури баз даних. Відповідно до технічного завдання проекту обрано клієнт-серверну модель, що була затверджена організацією та замовником проекту.

Розглянуто основні принципи та етапи управління проектом, визначено необхідні для проектного менеджера компетенції та знання, що сприятимуть покращенню процесу управління проектами.

Визначено основні тенденції управління проектом, на основі яких буде сформований проект управління розробкою реляційних баз даних для висококонкурентних додатків.

За результатами досліджень для управління даним проектом було обрано філософію Agile, завдяки її гнучкості. Слід зазначити що в проекті використовувалася не безпосередньо Agile, а методологія що поєднує два її фреймворки Scrum і Kanban – Scrumban. Ця методологія була обрана через свою зрозумілість і легкість реалізації.

При виборі інструменту для управління проектами менеджер проекту надав перевагу MS Project, так як вище керівництво організації користується середою Microsoft. Окрім самого MS Project також використовувалася платформа MS Teams, де проводилися щотижневі спринти та проходила основна комунікація команди проекту, а також дозволяла імпортувати таблиці з MS Project, що дозволяло робити швидкі правки в проекті.

На прикладі розглянуто процес управління проектом розробки реляційних баз даних для висококонкурентних додатків в сфері медицини. На основі обраних тенденцій, методологій та інструментів управління проектами було розроблено план та цілі проекту. Розглянуто та описано можливі ризики, розраховано бюджет та надано реалізацію проекту.

Визначено основні кроки завершення проекту, такі як закриття контрактів та робота над помилками що дозволило команді проекту виправити свій підхід до виконання завдань та реалізувати зміни в наступних проектах.

[15]