Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель програмного забезпечення,
що орієнтовано на управління проектами

(тема)

Виконав:
студент II курсу, групи КСМм-21–1
Судаков В.О.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо–професійна або освітньо–наукова)

Освітня програма
Комп'ютерні системи та мережі
(повна назва освітньої програми)

Керівник: доц. Філімончук Т.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо–професійна або освітньо–наукова)

Освітня програма Комп'ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(Підпис)
"____" _____ 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Судакову Валерію Олександровичу
(Прізвище, ім'я, по батькові)

1. Тема роботи Модель програмного забезпечення, що орієнтовано на управління проектами.

затверджена наказом по університету від " 07 " листопада 2022 р. № 1453 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2022 р.

3. Вхідні дані до роботи
Перелік інструментів управління проектами, модель керування проектами Agile, Scrum тех-
Kanban board, project management tools, діаграми UML, загальний опис та структура
моделі "AS IS" управління проектами на етапі R&D, загальний опис та структура
моделі бізнес-процесів на етапі R&D.

.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ.
Аналіз літератури та особливості побудови мереж спеціального призначення.
Методи та алгоритми управління проектами
Проектування автоматизованої системи управління проектами у компанії
Модель автоматизованої системи управління проектами
Висновки. Додаток.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Презентація Powerpoint 17 слайдів.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначку консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№.	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літератури за темою роботи	08.11.22–10.11.22	
2	Постановка мети та задач	11.10.22–12.11.22	
3	Методи управління проектами	13.11.22–12.11.22	
4	Розробка алгоритму управління проектами	13.11.22–18.11.22	
5	Проектування АСУ проектами компанії	19.11.22–24.11.22	
6	Експериментальна частина	25.11.22–29.11.22	
7	Розрахункова частина	30.11.22–02.12.22	
8	Підготовка пояснювальної записки	03.12.22–06.12.22	
9	Розробка презентації та доповіді	07.12.22–09.12.22	
10	Подача роботи у ЕК	10.12.21	

Дата видачі завдання 07.11. 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Філімончук Т.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 102 с., 32 рис., 7 табл., 16 джерел.

ПРОЦЕСИ РОЗРОБКИ, ІТ-ПРОЕКТ, СЕРЕДОВИЩА ДЛЯ УПРАВЛІННЯ ПРОЕКТАМИ, JIRA, PMBOK, ОНТОЛОГІЯ, PROJECT MANAGEMENT.

Мета кваліфікаційної роботи є вивчення сучасних методів та інструментів управління проектами.

У ході виконання кваліфікаційної роботи було з'ясовано переваги та недоліки сучасних методів та інструментів управління проектами, особливості застосування у різних ІТ-компаніях; проаналізовано існуючі інформаційні системи з управління проектами; проаналізовано зміст діяльності проектної команди компанії у процесі створення програмного продукту, який використовує штучний інтелект.

ABSTRACT

Master's thesis: 102 pages, 32 figures, 7 tables, 16 sources.

DEVELOPMENT PROCESSES, IT PROJECT, PROJECT MANAGEMENT ENVIRONMENTS, JIRA, PMBOK, ONTOLOGY, PROJECT MANAGEMENT.

The goal of qualifying work is to study modern methods and tools of project management.

During the qualifying work the advantages and disadvantages of modern project management methods and tools, features of their application in various IT companies were clarified; existing project management information systems were analyzed; the content of the activity of the company's project team in the process of creating a software product that uses artificial intelligence is analyzed.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PMBOK – Project Management Body of Knowledge

ШІ – штучний інтелект

PM – Project Management

PMO – Project Management Office

PMI – Project Management Institute

R&D – Reserch and Development

CRISP-DM – Cross-Industry Standard Process for Data Mining

ПМ – проектний менеджмент

IPMA – International Project Management Association)

OGC – Office of Government Commerce

ISO – International Standartization Organization

KPI – key performance indicators

ВСТУП

Суспільні та технологічні умови сучасного світового розвитку характеризуються постійним зростанням обсягу цифрових технологій, що ґрунтуються на використанні штучного інтелекту (ШІ), інноваційними процесами у суспільстві загалом, а також у сферах виробництва, науки, бізнесу. Зміни, що відбуваються в соціумі та в ІТ-області, породжують нові вимоги до якості управління проектами, що втілюють у життя «розумні» програми: розпізнавання осіб, постановка діагнозів, безпілотне керування транспортом тощо.

Робота з розробки нейромереж (як математичної основи штучного інтелекту) за своїм змістом є науково-дослідною, тобто має творчий характер, і отже, структура виробничих відносин у процесі виготовлення програмного продукту не така жорстка, менш ієрархічна, причому ієрархія протягом часу створення програми може порушуватись. Це зумовлює необхідність вибору більш гнучких та стійких до змін методологій управління проектами у компаніях, що спеціалізуються на розробці ІІ-продуктів. На сьогоднішній день найбільш поширеними та популярними методологіями є Agile-сімейство, що включає Scrum, Kanban та інші методи. Але вони не забезпечують повною мірою запити управління ШІ-проектами, як наукомісткими та великими масивами даних, що оперують.

Сучасний ринок інформаційних технологій пропонує великий вибір платформ для автоматизованого керування проектами. Асортимент таких систем дуже великий. Налічується близько ста автоматизованих систем керування процесом створення програмних продуктів. Найпопулярнішими з них є Trello, Jira та Asana. Кожен із запропонованих продуктів має свої переваги та недоліки. Так, Trello орієнтується на малі та короткострокові проекти, добре зарекомендувала себе у роботі з легко формалізованими проектами; Jira зручна у використанні у великих командах, але за умови попереднього налаштування та розширення базового набору функцій за допомогою плагінів та над-

будов. Цей процес є досить трудомістким, потребує багато зусиль. Сама Jira має складну структуру та важка у розумінні для нових користувачів.

Велика різноманітність у клієнтських запитах на ринку IT-послуг, спектр застосування ШІ, що постійно розширюється, зумовлює існування різних видів компаній, що відрізняються як за кількістю співробітників, так і виробленого продукту. Тому актуальними є питання адаптації існуючих моделей управління проектами до конкретної компанії.

З погляду менеджменту, дослідницька робота є одним із найскладніших об'єктів управління [6,8,13]. Це обумовлено особливостями творчого процесу мислення, невизначеністю та важкою передбачуваність кінцевих результатів, неможливість повної алгоритмізації процесу пошуку нових рішень.

Таким чином, виявляється суперечність: з одного боку, об'єктивно зростає частка дослідницької роботи у сфері IT-бізнесу, посилюється необхідність постійного вдосконалення системи управління інноваційними та дослідницькими проектами в галузі ШІ; з іншого боку, методи системи управління є недостатньо розробленими та часто не адекватними об'єкту управління. Така невідповідність показує те, що витрати, вкладені в проект, не окупаються і робота виявляється нерентабельною.

Таким чином, такі фактори: постійні та досить швидкі зміни в інформаційних технологіях; збільшення кількості точок застосування штучного інтелекту в усіх сферах соціуму.

1 ОГЛЯД ЛІТЕРАТУРИ І АНАЛІЗ ПРОБЛЕМИ ДОСЛІДЖЕННЯ

1.1 Основні поняття, що використовуються в управлінні проектами

Поняття «проект» поширене у різних галузях науки, виробництва, навіть у побуті. Розглянемо існуючі у науковій літературі визначення поняття «проект»:

- організована діяльність, спрямована на створення унікальних продуктів, послуг чи результатів (відповідно до PMBoK);
- окреме підприємство з певними цілями, що часто включають вимоги щодо часу, вартості та якості досягнених результатів (згідно з англійською Асоціацією проект-менеджерів);
- підприємство (намір), яке значною мірою характеризується неповторністю умов у їх сукупності, наприклад: завдання мети; тимчасові, фінансові, людські та інші обмеження; розмежування з інших намірів; специфічна для проекту організація його здійснення (відповідно до німецького стандарту DIN 6990) [9].

За Бегьюлі Ф., проект - це послідовність взаємопов'язаних подій, які відбуваються протягом встановленого обмеженого періоду часу і спрямовані на досягнення неповторного, але водночас певного результату [4].

Івасенко О.Г. дає таке визначення: «Проект – це обмежена за часом цілеспрямована зміна окремої системи з спочатку чітко визначеними цілями, досягнення яких визначає завершення проекту, із встановленими вимогами до термінів, результатів, ризику, рамок витрачання коштів та ресурсів та до організаційної структури» [10] .

За Фунтов В.М., проектом називається «цілеспрямована, обмежена у часі діяльність, здійснювана задоволення конкретних потреб за наявності зовнішніх і внутрішніх обмежень і використання обмежених ресурсів» [5].

Як бачимо, наведені визначення відрізняються досить загальним підходом до терміну «проект», проте дозволяють виділити істотні особливості даного поняття. До таких відносяться:

- наявність чітко сформульованих цілей, а також ряду технічних, економічних та інших цільових показників;
- системний характер будь-якого проекту, тобто наявність внутрішніх та зовнішніх зв'язків між усіма елементами системи, а саме цілями, завданнями,
- операціями, ресурсами (включаючи людські), шуканим результатом. Це дає можливість алгоритмізації проекту, тобто уявлення його у вигляді комплексу взаємозалежних процесів;
- наявність попередньо позначених часових інтервалів (терміни початку та кінця проекту);
- обмежені ресурси;
- певний ступінь унікальності цілей проекту та умов його здійснення.

Таким чином, проект – динамічна система дій, спрямованих на отримання заданих результатів у багатокритеріальному полі протягом встановленого терміну та в рамках виділених ресурсів із залученням виконавців, які мають необхідні навички та знання.

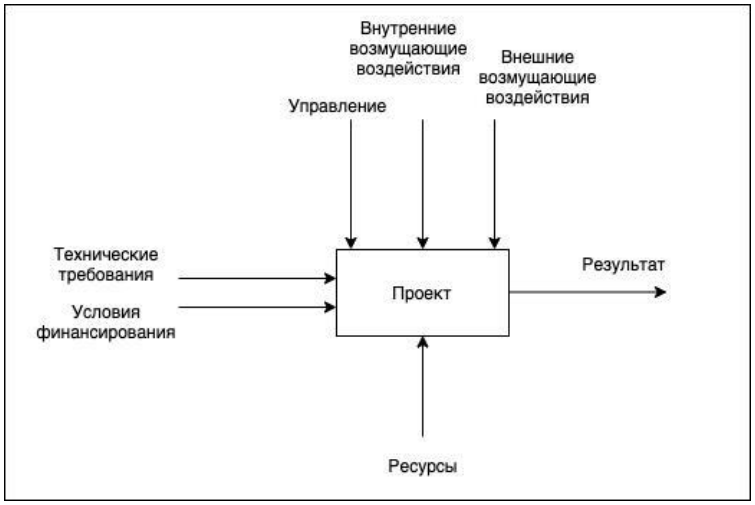

Рисунок 1.1 – Проект як система

Контекстна діаграма (рис. 1.1) представляє проект як систему. Вхідними даними є технічні вимоги та умови фінансування; метою є досягнення необхідного результату. У таблиці 1.1 розкривається зміст складових елементів проекту.

Таблиця 1.1 – Опис елементів системи проект

Елемент	Зміст
Ціль (результат)	Описуються нові продукти або послуги, які отримає замовник результаті реалізації проекту
Вартість проекту	Фінансові витрати, необхідні для виконання робіт проекту
Технічні вимоги	обсяги робіт (кількісні показники обсягу робіт проекту); термін виконання; якість (відповідність характеристик проекту та його продукції встановленим заздалегідь параметрам якості)
Ресурси	Устаткування, матеріали, персонал, програмне забезпечення, інформаційні системи; виробничі площі; фахівці та організації, залучені до виконання робіт проекту, їх кваліфікація
Внутрішні обурюючі впливу	Стиль керівництва проектом; організація проекту з погляду комунікації між основними учасниками проекту, розподілу прав, відповідальності та обов'язків; методи та засоби взаємодії між співробітниками всіх рівнів на проекті; умови праці та техніки безпеки, страхування та соціальне забезпечення та т.п.
Зовнішні обурюючі впливу	Взаємодія із замовником та конкурентами; ситуація на ринку та пов'язані з цим ризики; непередбачені обставини

Наявність ресурсів як матеріалів, фінансів, людського ресурсу забезпечує виконання робіт. Ефективність визначається управлінням процесом реалізації проекту. У функції управління входить розподіл ресурсів, координація

виконуваної послідовності операцій, компенсація внутрішніх і зовнішніх впливів, що обурюють [11, 12]

Існують також різні варіанти визначення терміну управління проектами. Розглянемо деякі з них.

Відповідно до PMBoK, управління проектами – це процес застосування знань, навичок, методів, засобів та технологій до проектної діяльності з метою втілення задумів учасників проекту [1].

Ентоні Вокер визначає управління проектами як планування, координацію та контроль проекту з позицій його завершення (і введення в дію) від імені замовника та з урахуванням його цілей в одиницях корисності, призначення, якості, строків реалізації та витрат; встановлення взаємозв'язку між ресурсами, координацію та контроль учасників проекту, їх персонального внеску у загальний результат, а також оцінку та вибір альтернатив задля найбільшого задоволення потреб замовника [5].

Гарольд Оберлендер вважає управління проектами мистецтвом та вмінням скоординувати матеріальні та нематеріальні ресурси, організувати послідовність робіт з реалізації проекту у часі та в рамках затвердженої вартості [4].

На думку Шапіро В. Д., управління проектами - це синтетична дисципліна, що поєднує спеціальні та надпрофесійні знання [4].

Отже, існує різноманітність у визначенні поняття «управління проектами».

Проте відмінністю будь-якого проекту є організація взаємодії між учасниками проекту, здійснювана через project manager'a. У процесі комунікації позначається коло проблем, які під час реалізації проекту та його можливі рішення.

Вочевидь, що – найбільш значний актив під час управління проектами, оскільки вони – джерело ідей та його реалізатори.

У науковій літературі є визначення різних типів проектів з різних підстав.

Внаслідок узагальнення існуючих у джерелах [1,9,10] класифікацій, збудуємо таблицю 1.2, у лівому стовпці якої зазначено основу класифікації, у правому – відповідний вид проекту.

Таблиця 1.2 – Види проектів

Критерій	Вид проекту
1	2
за складом та структурою проекту	монопроект, мультипроект, мегапроект
за основними сферами діяльності, в яких здійснюється проект	технічний, організаційний, економічний, соціальний, змішаний
за характером предметної галузі проекту	інвестиційний, інноваційний, науково-дослідницький, навчально-освітній, змішаний
за тривалістю періоду здійснення проекту	короткострокові (до 2-х років), середньострокові (до 5-ти років), довгострокові (понад 5 років)
за масштабом (за розмірами бюджету, кількості учасників та ступеня впливу на навколишній світ)	дрібні, середні, великі
за ступенем охоплення етапів інноваційного процесу	повні інноваційні проекти, що включає НДР, ДКР, освоєння нововведення та його комерціалізацію, неповні інноваційні проекти, що включають окремі етапи інноваційного процесу

Продовження таблиці 1.2

1	2
за складністю	прості, складні, дуже складні
по галузях економіки та соціальної сфери	промисловість, будівництво, транспорт, охорона здоров'я, туризм
за результуючим продуктом	продукт (частина іншого виробу, удосконалення виробу або кінцеве виріб); послуга (бізнес-функція, оптимізує виробничі процеси); поліпшення (існуючої лінійки продуктів або послуг); нематеріальний продукт (наприклад, дослідницький проект приносить нові знання)

Класифікація проектів дозволяє досить чітко ранжувати перспективні та реалізовані проекти, і як наслідок, ставити здійсненні цілі, задавати реальні терміни досягнення цілей, та залучати оптимально необхідні ресурси для їх успішної реалізації [2, 9].

Поняття життєвого циклу проекту передбачає певну послідовність етапів реалізації тієї чи іншої ідеї щодо виробничого чи управлінського процесу. Важливість даного поняття зумовлюється тим, що воно фіксує тривалість проекту, чітко означаючи термін його виконання; дозволяє деталізувати процес реалізації задуму, розбиваючи його за конкретні фази; дає можливість чітко визначити кількість задіяного персоналу та необхідні ресурси; полегшує процедуру контролю.

Життєвий цикл проекту – це сукупність фаз, якими реалізується початковий задум. Такий поділ важливий не тільки з теоретичної, але також і з практичної точки зору, адже він дає можливість краще контролювати процес виробництва програмного продукту. Так, прийнято виділяти такі етапи життєвого циклу ІТ-проекту: ініціація, планування, виконання, завершення (рисунок 1.2).

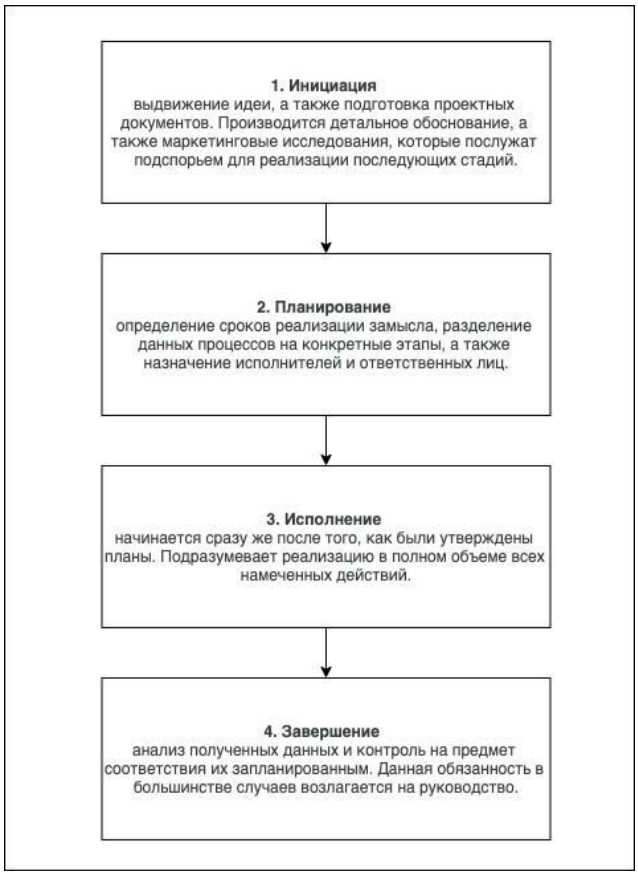

Рисунок 1.2 – Етапи життєвого циклу проекту

При вивченні життєвих циклів проектів було помічено, що, незважаючи на їхню різну спрямованість, усім проектам притаманні такі загальні властивості:

- найвищі показники витрат та кількість одиниць персоналу, присутній на проекті, властиві середині циклу; початок та кінець даного процесу характеризуються невисокими показниками;
- рівень ризику найвищий на початку життєвого циклу;
- саме на початку життєвого циклу проекту співробітники мають велику свободу щодо внесення змін та вдосконалення технології реалізації проекту, з часом це стає зробити все складніше [8,11].

Як свідчить Султанов І.А., «кожен project manager, набираючи досвід, дедалі більше розуміє значимість життєвого циклу у тому, щоб проектна реалізація щоразу проводилася дедалі безпечніше і з більш прогнозованим ре-

зультатом. У цьому допомагає як система оцінки ризиків. Велике значення має планування проекту з фаз його ЖЦ. Після кожного етапу намічаються вехи. У ці моменти керівники повинні зупинитися, оцінити досягнутий результат, здійснити прогнозний аналіз та вирішити подальшу долю унікального завдання. Досвід, знання та управлінська інтуїція одного з лідерів бізнесу дозволяють довірити йому такі відповідальні рішення»[12].

Методологічна цінність моделі життєвого циклу у цьому, що вона допомагає зрозуміти особливості виконання роботи, вчасно прийняти необхідні управлінські рішення, оскільки модель – це докладний опис послідовності необхідних дій, що дозволяє втрачати контроль над розробкою проекту на жодній стадії життєвого циклу.

1.2 Моделі життєвих циклів проектів та методи управління проектами

Управління проектами як вид професійної діяльності та як об'єкт наукових досліджень отримує суттєвий розвиток у 1980-х роках, коли світова економіка виходить із кризи, зростає насичення ринку та виникає необхідність вирішення нових, великих завдань. Саме тоді виходить перша значна робота PMBoK (A Guide to the Project Management Body of Knowledge), виконана в Project Management Institute (PMI). Ця організація на сьогоднішній день є найбільш авторитетною професійною асоціацією, яка розробляє стандарти у сфері управління проектами. Серед інших інститутів, які займаються стандартизацією проектного менеджменту, слід назвати IPMA (International Project Management Association), OGC (The Office of Government Commerce, PRINCE2), ISO (International Standartization Organization),

Розвиток IT-сфери, інтенсивна робота в галузі програмного забезпечення призвели до накопичення (і цей процес не зупиняється) великого практичного досвіду ("best practice"). Комплекс таких «найкращих практик», що реалізуються на різних стадіях життєвого циклу проекту та базуються на спільній ідеології, стандарт SWEBOK [14] називає «методологія розробки про-

грамного забезпечення». Методології, або методи розробки програмних продуктів на сьогоднішній день є областю інформаційних технологій, що швидко розвивається, оскільки спираються на реальні практичні знання.

Методи управління проектами тісно пов'язані із життєвим циклом проекту. Можна сказати, що вони взаємно обумовлюють одне одного. Фази життєвого циклу проекту представлені на рис. 1.2, є дуже узагальненими і однаково властивими проектам, що виконуються, наприклад, будівельної сфері, де всі етапи чітко позначені і зрозуміла послідовність різних видів робіт, або, наприклад, в авіаційній або космічній області, де можливе розпаралелювання операцій, або повторення (якщо є необхідність) одних і тих же видів робіт [5, 7, 12 , 13].

Тому розробки конкретного програмного продукту необхідна докладна деталізація його життєвого циклу і адекватна (змісту проекту) модель.

«У загальному випадку, життєвий цикл визначається моделлю та описується формі методології (методу). Модель чи парадигма життєвого циклу визначає концептуальний погляд на організацію життєвого циклу та, часто, основні фази життєвого циклу та принципи переходу між ними. Методологія (метод) задає комплекс робіт, їх детальний зміст та рольову відповідальність фахівців на всіх етапах обраної моделі життєвого циклу, зазвичай визначає і саму модель, а також рекомендує практики (best practices), що дозволяють максимально ефективно скористатися відповідною методологією та її моделлю» [13].

Традиційними та хронологічно першими були розроблені каскадна (водоспадна, waterfall) модель, вперше описана у 1970 році у роботі Уінстона Ройса [12] та спіральна (Spiral), запропонована Баррі Боемом у 1986 р. [10]. Пізніше, з ускладненням інформаційних технологій та наростаючим функціоналом програмних продуктів виникли так звані гнучкі (Agile) моделі [13].

Нині вони набули найбільшого поширення. Розглянемо сучасні методи та алгоритми управління проектами та проведемо їх порівняльний аналіз за найбільш суттєвими параметрами.

Каскадна модель (Waterfall).

Найстарішою та найвідомішою моделлю побудови багаторівневого процесу розробки є каскадна (або водоспадна) модель: у ній кожен етап розробки, що відповідає стадії життєвого циклу ПЗ, продовжує попередній. Тобто новий етап починається лише після повного завершення цього.

Каскадна модель проста і зрозуміла, але не така практична, як раніше. В умовах вимог, що динамічно змінюються, строго структурований процес може з переваги перетворитися на перешкоду на шляху успішного завершення розробки системи. Тому сьогодні водоспадна модель застосовується переважно великими компаніями для великих та складних проектів, які передбачають всеосяжний контроль ризиків [7].

Рисунок 1.3 – Каскадна модель життєвого циклу

Переваги каскадної моделі:

- зрозуміла та чітка схема робочого процесу;
- можливість прорахунку точної кількості витрачених на проект ресурсів;
- не потребує витрат на налагодження комунікацій між усіма членами команди.

Недоліки каскадної моделі:

- пріоритет формального підходу до послідовності процесу роботи;

- неможливість внесення змін замовником до закінчення розробки продукту;

- у разі нестачі ресурсів страждає на якість проекту через скорочення етапу тестування.

Незважаючи на те, що каскадна модель все ще використовується, вона вже втратила колишні позиції. Сьогодні їй на зміну приходять просунутіші моделі та методології розробки програмного забезпечення.

Спіральна модель.

Спіральна модель використовує розбиття проекту на ітерацію. Підвищена увага приділяється початковим етапам розробки – аналізу та проектування. Основна функція початкового етапу – обґрунтування можливості реалізації програмного рішення; при цьому створюється прототип майбутнього продукту [8].

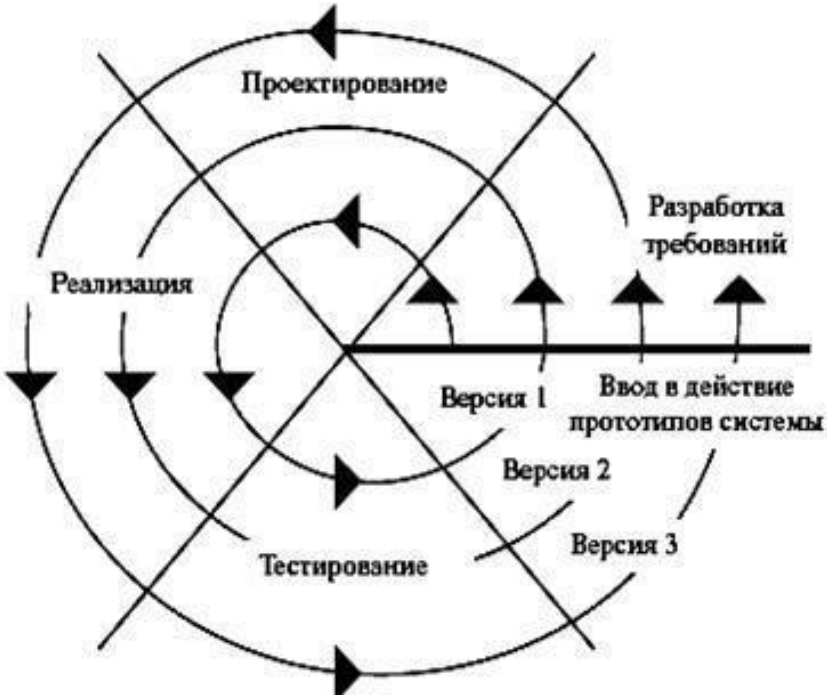

Рисунок 1.4 – Спіральна модель життєвого циклу

Свою назву спіральна модель отримала внаслідок того, що життєвий цикл проекту в цій моделі завдяки ітераційному (тобто поступово наближ-

чому) підходу створює образ спіралі – з подоланням кожної ітерації, або окремого циклу, майбутній програмний виріб набуває необхідних форм та змісту.

Такий підхід дозволяє на кожній ітерації приділяти час уточненню цілей і параметрів проекту, з'ясування труднощів, що виникають, планування робіт на наступному витку.

Завдяки такому підходу до розробки час виконання всіх поставлених завдань зменшується.

Переваги спірального підходу:

- подолання жорсткого, регламентованого процесу розробки у каскадній моделі; створення додаткових можливостей;

- полегшення процесу внесення коректив до проекту, якщо вимоги замовника змінилися;

- постійний зв'язок користувача майбутнього продукту з розробником, що дозволяє оперативно відслідковувати помилки, що виникають, або слабкі місця проекту;

- більш ефективне керування проектом завдяки наявності ітерацій;

- зниження рівня ризиків.

Проблеми використання спіральної моделі:

- відсутність обмежень часу на кожному з етапів життєвого циклу;

- то перевага, що є можливість постійно вносити зміни до вимог до проекту, обертається тим, що кількість циклів може зростати, отже подовжуються терміни розробки;

- досить складна структура моделі, що створює складнощі для розробників і менеджерів при її застосуванні.

Гнучкі (ітеративні) моделі (Agile).

Крім послідовних моделей, є ітеративні (інкрементальні). Це ціла родина методологій, об'єднаних загальним підходом в управлінні проектами:

- люди та взаємодія важливіші за процеси та інструменти;

- працюючий продукт важливіший за вичерпну документацію;

- співробітництво із замовником важливіше за погодження умов контракту;

- готовність до змін важливіша за проходження початкового плану [6,8].

У таких моделях життєвий цикл проекту розбитий на кілька міні-циклів, кожен з яких складається з базових стадій моделі життєвого циклу. Ці міні-цикли називаються ітераціями.

Розробка окремого компонента систем відбувається всередині ітерації, потім цей компонент додається раніше розробленому функціоналу.

Ітеративна модель не потребує повного обсягу вимог для початку робіт над продуктом. Розробка програми може починатися з вимог до частини функціоналу, причому згодом вимоги можуть змінюватись і доповнюватися. Шляхом повторення процесу відбувається оновлення версії продукту кожному циклі.

Рішення про використання результатів попередньої ітерації як вхідні для наступної приймається після закінчення попередньої (т.зв. інкрементальне прототипування). Зрештою, досягається точка, у якій усі вимоги були втілені у продукті – відбувається реліз.

Використовуючи математичну термінологію, ітеративна модель реалізує послідовну апроксимацію, тобто наближення до заданих параметрів, тобто. готового продукту.

Успіх використання цієї моделі визначається строгою валідацією вимог і верифікацією функціональності, що розробляється, на кожній ітерації.

Основні стадії процесу розробки в ітеративної моделі практично повторюють модель водоспаду. У кожній ітерації створюється програмне забезпечення, яке вимагає тестування всіх рівнях.

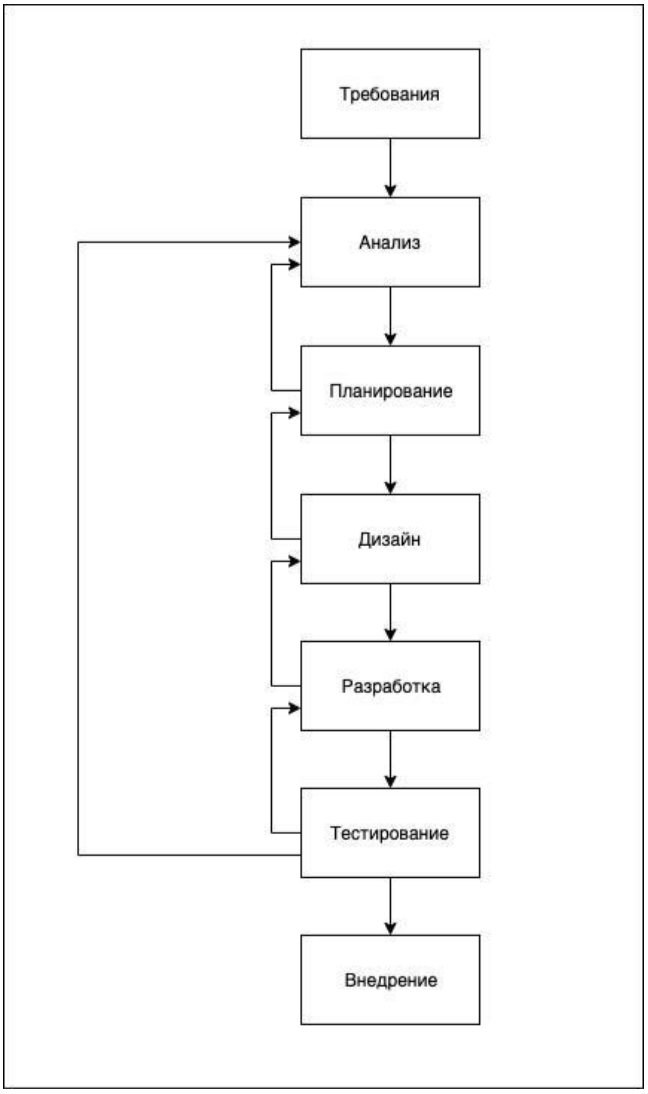

Рисунок 1.5 – Ітеративна модель життєвого циклу

Переваги ітеративної моделі:

- високий рівень взаємодії між членами команди проекту;
- швидкий результат (робочий код) у результаті «спринтів»;
- стимулювання зміни та покращень продукту під час його розробки;
- безпосереднє залучення замовника до робочого процесу;
- висока здатність до адаптивності під різні процеси та умови;
- швидкий відгук на зміни зовнішніх та внутрішніх умов проекту;
- пристосований для розробки інноваційних продуктів з високим ступенем невизначеності та недостатньою інформативністю;

Недоліки:

- ризик нескінченних змін продукту;
- велика залежність від рівня кваліфікації та досвіду команди;
- практично неможливо точно підрахувати підсумкову вартість проекту;
- ефективність застосування Agile-методик суттєво залежить від кваліфікації менеджменту (для полегшення застосування підходу прийнято використовувати методи Scrum, Kanban та інші).

Найбільш популярними представниками сімейства Agile-методології є Scrum, Lean, Canban, Six Sigma, PRINCE2.

Розглянемо докладніше Scrum та Canban.

Scrum- найбільш структурований у групі Agile-методів.

Спрощена схема роботи з Scrum полягає в наступному. Згідно технології Scrum, проект розбивається на частини - завдання, які будуть створюватися протягом проекту. Після цього отриманим частинам надається свій пріоритет.

Хронологічно проект розбивається на ітерації – певні проміжки часу, що встановлюються командою та погоджені із замовником для поточного контролю та управління проектом.

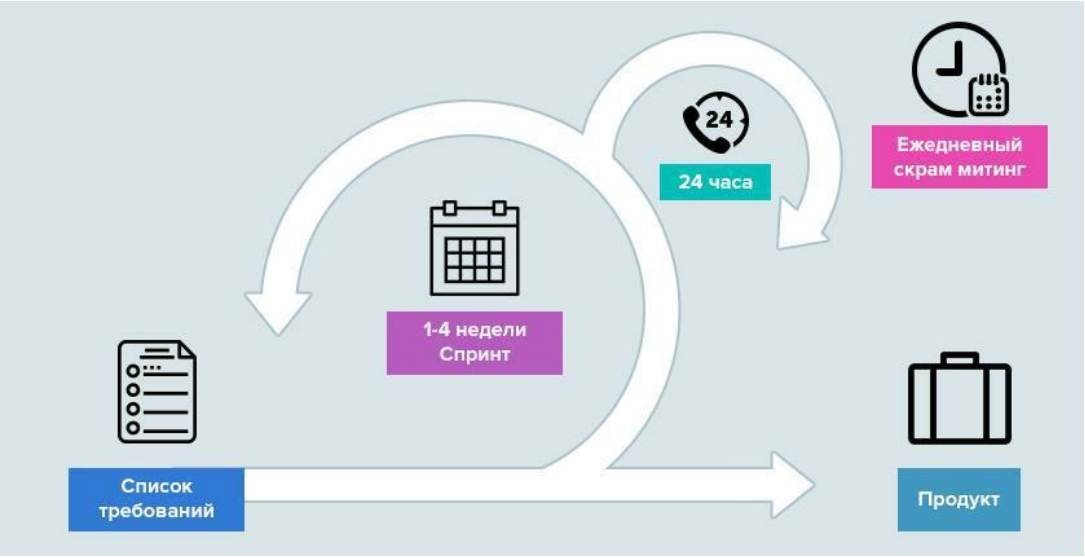

Рисунок 1.6 – Scrum модель життєвого циклу

Найважливіші завдання першими відбираються до виконання в ітерації.

За підсумками ітерації замовник отримує робочий проміжний варіант товару. Щоб переконатися відповідно до проекту вимог замовника, перед початком будь-якого спринту потрібно виконувати переоцінку ще не виконаного змісту проекту та вносити до нього зміни.

- Участь у цьому процесі бере керівник проекту, команда та ініціатор. Відповідальність розподіляється усім учасників.

Процес роботи у середовищі Scrum спирається п'ять виробничих нарад:

- нарада-упорядкування; у перший день нової ітерації відбувається обговорення того, що вже зроблено за проектом, що ще потрібно виконати;

- нарада-планування; команда вибирає зі списку пріоритетних завдань ті, які виконуватиме протягом поточної ітерації;

- наради-летучки; відбуваються щодня, оперативно (до 15 хвилин) обмінюватись даними про поточний стан виконання індивідуальних завдань кожного члена команди;

- підсумкові наради; команда демонструє свій результат усім заінтересованим особам; з'ясовується, наскільки продукт відповідає цілям та очікуванням;

нарада-ретроспектива. Проводиться аналіз того, наскільки злагоджено працювала команда, обговорюються проблеми методів і комунікації, робляться певні висновки.

Переваги Scrum:

- зручний для проектів, у яких можливий швидкий, оформлений та придатний для первісного використання результат;

- тісна та постійна комунікація співробітників допускає присутність у команді малодосвідченого розробника;

- швидке виправлення помилок завдяки миттєвому зворотному зв'язку між усіма учасниками проекту.

Недоліки Scrum:

- підходить не кожному темпераменту; наявність постійної та дуже тісної комунікації всіх учасників проекту висуває вимоги до психологічних властивостей та соціальних навичок учасників проекту;

- не підходить для проектів із строго регламентованих та нормованих областей (наприклад, юриспруденція).

Kanban-метод із групи Agile.

На відміну від Scrum, Kanban-метод передбачає наявність чітких тимчасових ітерацій. Процес створення продукту не поділяється на ітерації, а передбачає обмеження кількості одночасно виконуваних завдань.

Особливості методу в тому, що немає обмежених за часом ітерацій, регламентованих виробничих нарад, кожен співробітник команди може вирішувати кілька завдань одночасно. Відсутність часових рамок кожного етапу дозволяє призупинити виконання поточного завдання, якщо змінився пріоритет або з'явилися інші, більш важливі завдання.

Переваги Kanban:

- відсутність дедлайнів;

- економічно ефективний внаслідок розумного розподілу кількості завдань кожного члена команди.

Недоліки Kanban:

- найбільш ефективний у командах, члени яких взаємозамінні;

- не придатний для проектів, терміни виконання яких суворо обумовлено.

Crisp DM.

З розвитком методів створення штучного інтелекту постало питання перегляду життєвої моделі циклу проектів. Програмні продукти, що розробляються на основі штучного інтелекту, суттєво відрізняються від стандартних програмних продуктів (тобто таких, що не містять ІІ) в організації структури життєвого циклу. Крім того, такі продукти ґрунтуються на використанні великих масивів даних, а це потребує інших підходів до управління проектми.

Платформа CRISP-DM спочатку була створена для Data Mining - глибинного аналізу даних, згодом виявилося, що вона добре моделює життєвий цикл II-вмісних програмних продуктів.

Зміст моделі CRISP-DM життєвого циклу дослідження даних полягає в інтерпретації та обробці даних. Усередині процесу побудови Data Science-моделі немає накопичення результатів як із ітеративних, спіральних та інших моделях. Кожен новий цикл оновлює (або навіть обнуляє) ті результати, що вийшли у попередньому циклі.Розглянемо докладніше модель CRISP DM. Загалом вона представлена рисунку 1.4.

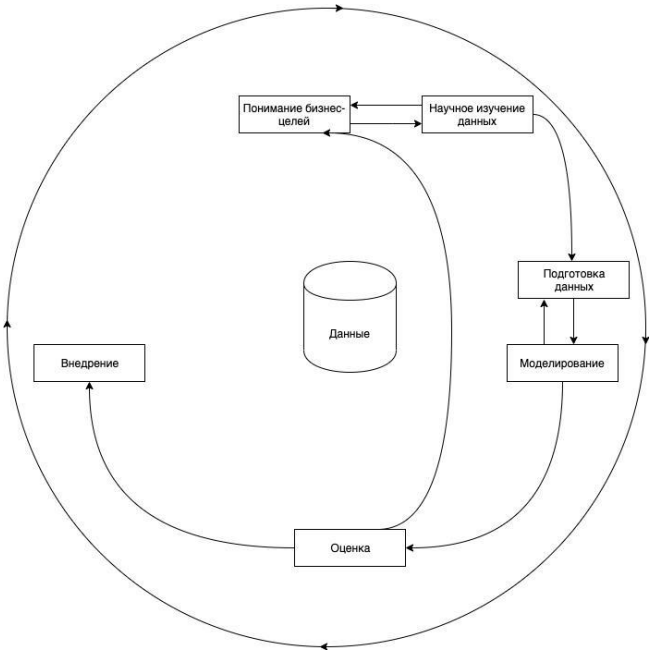

Рисунок 1.7 – Модель життєвого циклу дослідження даних CRISP-DM

Модель життєвого циклу CRISP-DM містить шість фаз:

- розуміння бізнес-цілей (Business Understanding). Змістом цієї фази є таке: дослідження бізнес-процесів компанії та висування пропозицій щодо аналізу даних виходячи з кінцевих цілей аналізу. В обговоренні бере участь максимальна кількість зацікавлених осіб, які зрештою виробляють план аналітичного проекту. Вирішується також питання щодо доцільності проекту;

- початкове вивчення даних (Data Understanding): докладне вивчення наявних даних з метою усунення проблем на стадії підготовки даних. В результаті організується доступ до даних, їх дослідження з використанням таблиць та графіків, оцінка якості даних, розробка необхідної документації;

- підготовка даних (Data Preparation): найбільш трудомісткий та відповідальний етап, що поглинає 50-70% часу та ресурсів. Підготовка даних - це їхня консолідація; формування вибірок; агрегування; збагачення; очищення; поділ даних на навчальні та тестові;

- моделювання (Modeling): побудова та використання аналітичних моделей, зазвичай протягом кількох ітерацій;

- оцінка (Evaluation) відповідності результатів проекту критеріям. Рішення про відповідність приймається найбільш відповідальними особами компанії, які формулюють бізнес-мети;

- впровадження (Deployment) як процес використання нових ідей та знань для підвищення ефективності діяльності підприємства.

На рисунку 1.8. показані завдання кожної фази життєвого циклу проекту, що розробляється на основі нейронної мережі.

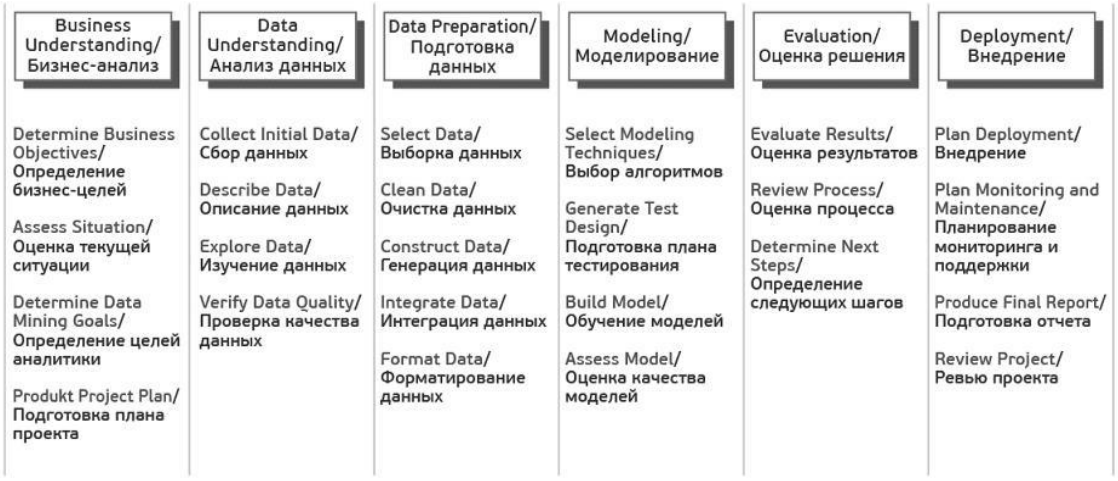

Рисунок 1.8 – Завдання фаз Crisp DM

До переваги цієї моделі можна віднести нейтральність щодо предметних областей; зручність використання на проектах, що використовують

ІІІМО; одна з небагатьох моделей, що включає інтелектуальний аналіз даних; є одним із найважливіших понять для технологій великих даних (Big Data).

Недоліки моделі такі: відсутність інструментів управління проектами; відсутній критерій часу кожного з циклів; цикли можуть бути різною тривалістю.

1.3 Аналіз інструментів управління проектами

Інструменти, які використовуються менеджером проекту, можуть застосовуватися як протягом усього проекту, так і на певному етапі (фазі) життєвого циклу.

Під інструментами управління ми розумітимемо закінчені формалізовані методики, процедури, а також шаблони необхідних проектних документів.

Програмне забезпечення Trello для управління Agile проектами. Trello хмарна програма для управління проектами невеликих груп, розроблена Fog Creek Software.

Це безкоштовна багатоплатформова система керування проектами (рис. 1.9, 1.10).

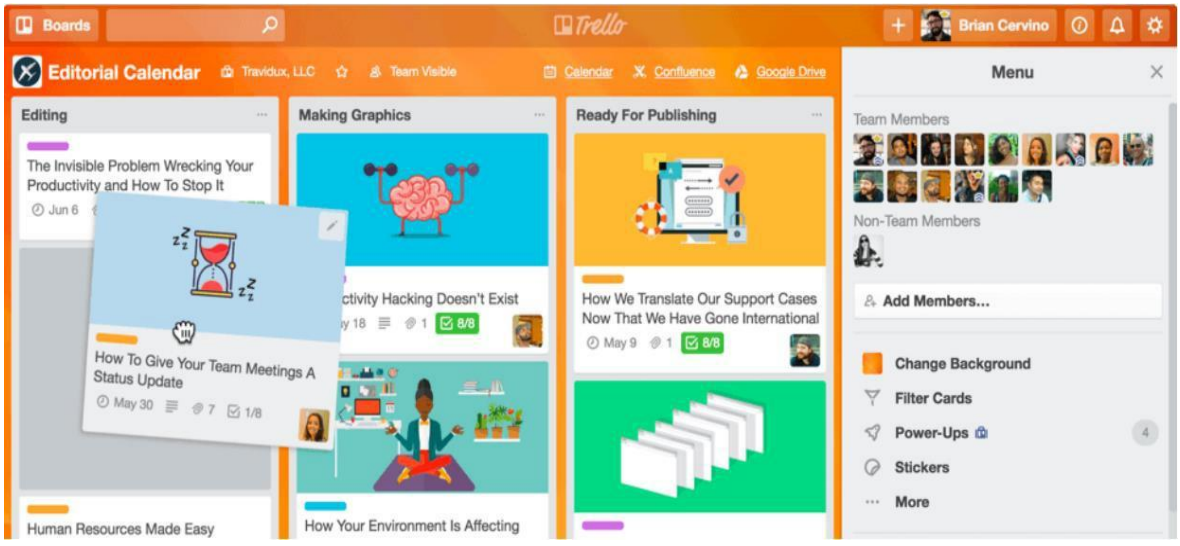

Рисунок 1.9 – Trello Board

Вона використовує парадигму управління проектами, відому як канбан.

Проекти зображуються дошками, що містять список. Списки містять карти, якими зображуються завдання.

Картки повинні переходити з попереднього списку до наступного (за допомогою перетягування), таким чином зображуючи рух якоїсь функції від ідеї до тестування.

Картці може бути надано ім'я відповідальних за неї користувачів.

Користувачі та дошки можуть об'єднуватись у команди. Trello має обмежену підтримку тегів у вигляді шести кольорових міток. Картки можуть містити коментарі, вкладення, дату завершення та переліки (списки підзавдань). Форматуються картки розміткою Markdown [11].

Переваги роботи з Trello полягають у наступному:

- наявність функціональних можливостей, що спрощують процес управління проектом, а саме: обмін файлами (включаючи фотографії та відео) зі своїми членами команди; коментарі до карти; стеження за to-do list; встановлення кольорових міток відповідно до пріоритету; необмежене створення карток під проект;

-

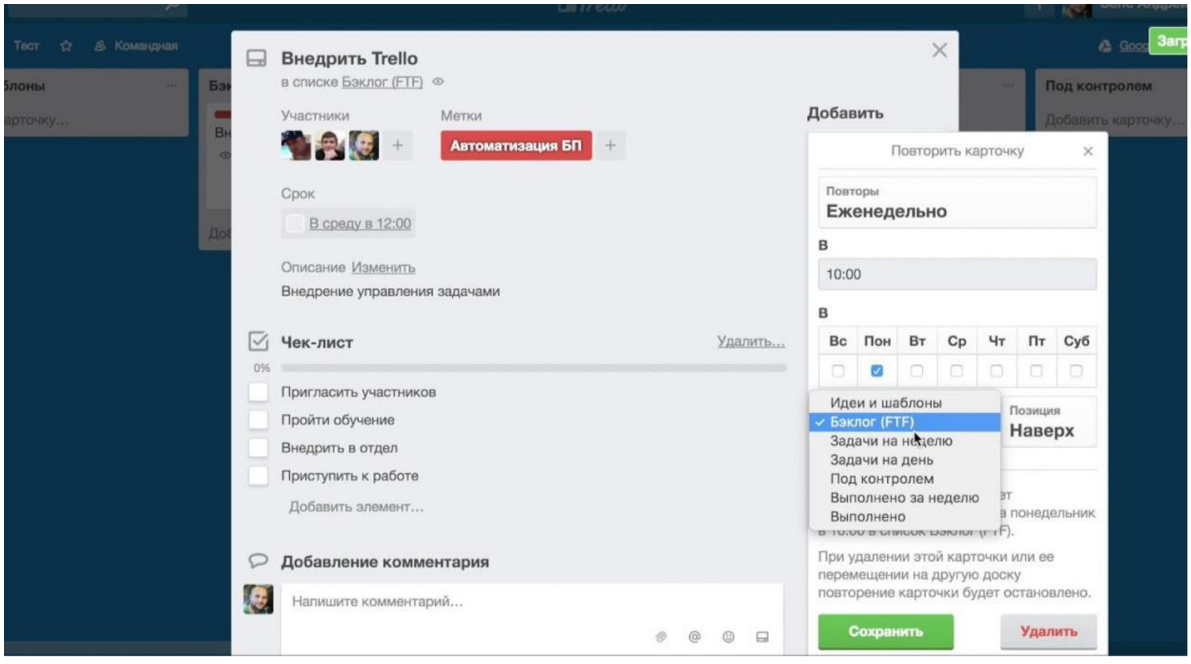

Рисунок 1.10 – Картки завдання Trello

- наявність функціональних можливостей, що спрощують процес управління проектом, а саме: обмін файлами (включаючи фотографії та відео) зі своїми членами команди; коментарі до карти; стеження за to-do list; встановлення кольорових міток відповідно до пріоритету; необмежене створення карток під проект;

- вартість. У порівнянні з іншими інструментами управління проектами, Trello має менш складну структуру ціноутворення. Простіший варіант версії дозволяє запросити необмежену кількість учасників, створити дошки, карти та списки. Версія бізнес-класу, вартість якої 25 доларів США на місяць, надає цілу низку функцій, таких, як інтеграція до Служб Google, легкий об'ємний експорт, а також можливість доступу та управління всіма дошками [14];

- зрозумілий час моніторингу. Ви ніколи не пропустите дату завершення проекту або важливу зустріч із Trello Борд. При створенні Cards можна додати відповідні дати (deadline). Картка стане жовтою за 1-2 дні до закінчення проміжку часу, відведеного на виконання; картка стане повністю червоною в день здачі проекту;

- mobile friendly. Trello працює на кожній платформі. Незалежно від того, чи він на комп'ютері, планшеті або телефоні, інструмент може адаптуватися під розмір екрану;

- instant notifications. Ви завжди будете в курсі новин, оскільки Trello снащена функцією миттєвого оповіщення під час оновлення, коментування або видалення завдання. Також можна отримувати сповіщення електронною поштою.

Недолікироботи з Trello наступні:

- немає offline доступу. Trello не працюватиме, якщо немає wifi. Якщо ви подорожуєте або ваш електронний пристрій переключено на режим "в літаку", ви не отримаєте доступу до Trello. Доступ до сайту Trello вимагає підключення до Interwebs Ці особливості, властиві Trello, створюють певний дискомфорт для користувача, що знаходиться в автономному режимі;

Таблиця 1.3 – Огляд Trello

Призначення продукту	Відстеження завдань проекту та їх статусів. Управління невеликими проектами.
Users	Software developers Project managers SCRUM masters
Use Cases	Управління проектами. Менеджмент продукту. Управління завданнями. Розробка програмного забезпечення. Agile розробка програмного забезпечення
Інтеграція	Confluence. JIRA. Slack. GitHub
Hosting options	Cloud

- ця платформа не призначена для роботи над великими проектами. Оптимальне її застосування – стартапи, домашні проекти чи деяка ідея, яка потребує перевірки. Trello – незручний інструмент управління масштабними проектами з кількома командами з усього світу;

- обмежений обсяг файлосховища. Кількість вкладених файлів на карті Trello може бути будь-яким, але для кожного вкладеного файлу існує ліміт завантаження файлу до 10 МБ. Члени бізнес-класу та Trello Gold можуть використовувати обмеження завантаження файлів розміром до 250 Мб. Ліміту зберігання даних облікового запису немає;

- незручність коментування. Після опублікування та збереження коментаря на карті редагувати його у разі потреби неможливо, доведеться писати новий коментар;

- недостатній рівень візуалізації процесу роботи всіх членів команди одночасно;

- неможливо створити довгострокові плани. Дошка проста і зручна для розробки програмного забезпечення невеликих проектів, але для проектів

тривалістю навіть два тижні створити дорожню карту виявляється неможливим;

- немає можливості переглядати ітерації. Ефективність роботи підвищується за умови регулярного спостереження за виконанням поточних завдань, продуманого плану контролю майбутніх завдань, а також можливості аналізу та коригування вже виконаної роботи. Trello не має вбудованих інструментів для перегляду та аналізу реалізованих завдань для успішного проведення ретроспектив;
- немає можливості вносити дані про витрачений завдання часу.

Відсутність можливості для менеджерів проекту та його учасників відстежувати кількість часу, витрачену на виконання кожного завдання. Це ускладнює складання щомісячної звітності, а також ускладнює контроль за виконанням завдання.

Програмне забезпечення ASANA для керування Agile-проектами. Asana (Асана) – одна з провідних мобільних та веб-додатків для управління проектами в невеликих командах.

Це універсальне рішення для управління проектами та завданнями, яке дозволить автоматизувати деякі з найскладніших завдань, таких як комунікація та зручна співпраця. Багато компаній хвалять Asana за точність та ефективне використання часу, особливо відзначаючи зручність відстеження завдань та обговорення їх у реальному часі (рис. 1.9).

Основний акцент творці сервісу роблять на тому, що тепер управляти проектами можна і без використання електронної пошти. Розробники Asana надали користувачам можливість обговорювати робочі процеси на тій же платформі, де вони були створені, та вдосконалили програму додатковим функціоналом: notes, groups, combined tasks, and followers [15].

Asana миттєво повідомляє про кожну зміну, яка може вплинути на проект. Програмне забезпечення допомагає зробити якісну оцінку проекту та керувати ризиками до моменту підписання контракту з клієнтом.

Сервіс пропонує синхронізацію у режимі реального часу для різних пристроїв; перевірку вашої поштової скриньки; створення плану на наступний день; перегляд, редагування та створення завдань та проектів з пріоритетами; структурування проектів з допомогою розділів; лайки до завдань; обговорення щодо проекту та завдань; дешборди, що настроюються; календарі; пошук завдань, проектів, людей та тегів; прикріплення файлів із Dropbox, Box або Google Диску; канбан-дошки.

Основні характеристики Asana представлені у таблиці 1.4.

Розглянемо переваги Asana:

- розширений перелік функцій управління;
- існує безкоштовна пробна версія. Сервіс також є безкоштовним для команди до 15 осіб;
- позначення кольором найважливіших завдань. Зручний функціонал для making priority for task (рисунок 1.11);

Таблиця 1.4 – Огляд Asana

Призначення продукту	Планування, документування, відстеження та випуск програмного забезпечення.
Users	Software developers Project managers SCRUM masters
Use Cases	Управління проектами Менеджмент продукту
	Управління завданнями Розробка програмного забезпечення Agile розробка програмного забезпечення
Інтеграція	JIRA Slack GitHub
Hosting options	Cloud

- необмежений функціонал додавання більш ніж одного тега до кожного елемента (рис. 1.12);

- підтримка мобільної версії для різних платформ;

- одна історія / елемент / завдання може мати посилання більш ніж на один проект і тег;

- є підтримка гарячих кнопок;

- додавання нових завдань без необхідності заповнення багатьох полів;

- інтеграція електронної пошти;

- чітке дерево завдань, які потрібно виконати. Можливість побачити всі завдання у одному місці.

- платформа «Преміум» коштує 10 доларів США за кожного члена/на місяць (у разі, якщо ви хочете мати додатковий та розширений функціонал) [7].

Недоліки Asana наступні:

- можна призначити історію/завдання лише одному з членів вашої команди;

- графічні деталі перевантажують інтерфейс, роблячи його повільним під час переміщення;

- важко працювати із завданнями у завданні (sub-tasks). Є можливість втратити зв'язок під час переміщення елементів;

- надмірна кількість функцій, що відображаються на екрані, ускладнюють та уповільнюють процес навчання.

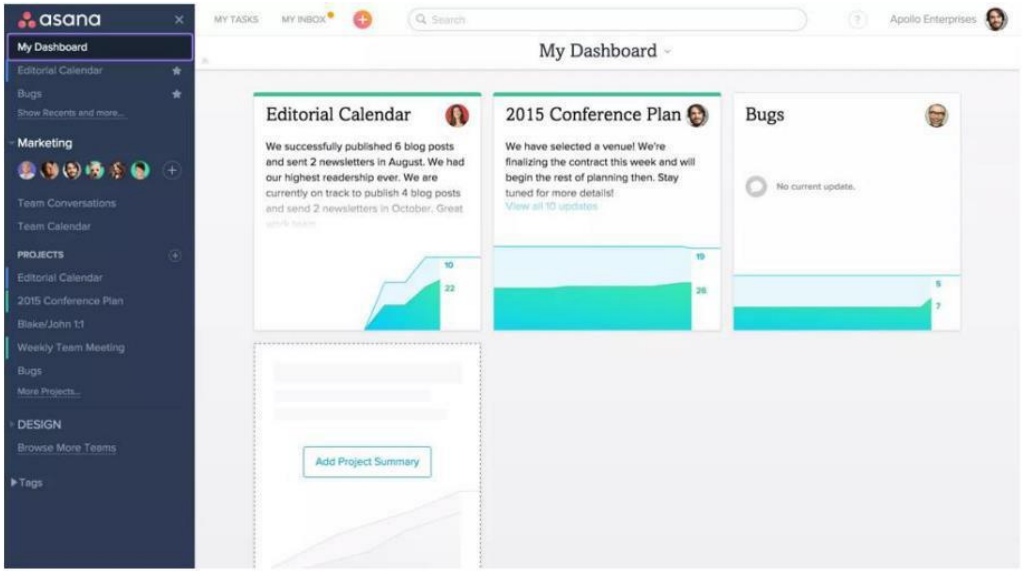

Рисунок 1.11 – Asana Dashboard

Повідомлення через програму зашифровані, але, на жаль, немає двоетапної перевірки. Відсутність 2-хвилинного керування робить програмне забезпечення небезпечним для ваших особистих даних [7].

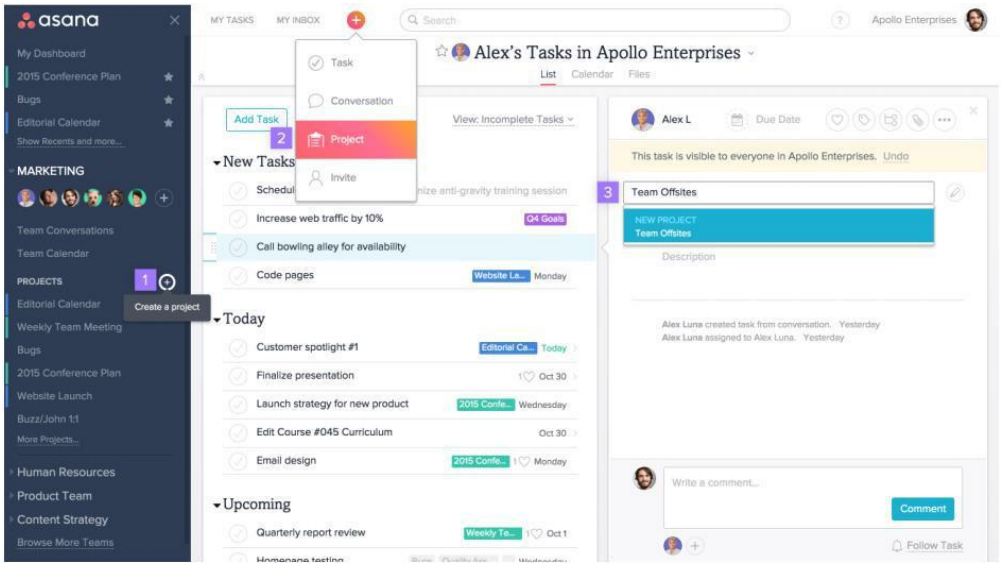

Рисунок 1.12 – Відкрите завдання в Asana

Програмне забезпечення JIRA для управління Agile проектами. JIRA – система відстеження помилок. Даний продукт розроблений для організації

взаємодії з користувачами, також її можна використовувати для управління проектами. Розробник зазначеної системи – компанія Atlassian.

Jira є одним із двох її основних продуктів (поряд з вікі-системою Confluence). Однією з переваг Jira є наявність веб-інтерфейсу [12].

JIRA - це комплекс систем, що дозволяє координувати роботу всіх команд, зайнятих розробкою нового продукту. JIRA пропонує кілька продуктів та варіантів розгортання, призначених спеціально для програмного забезпечення, IT, бізнесу, команд операторів тощо.

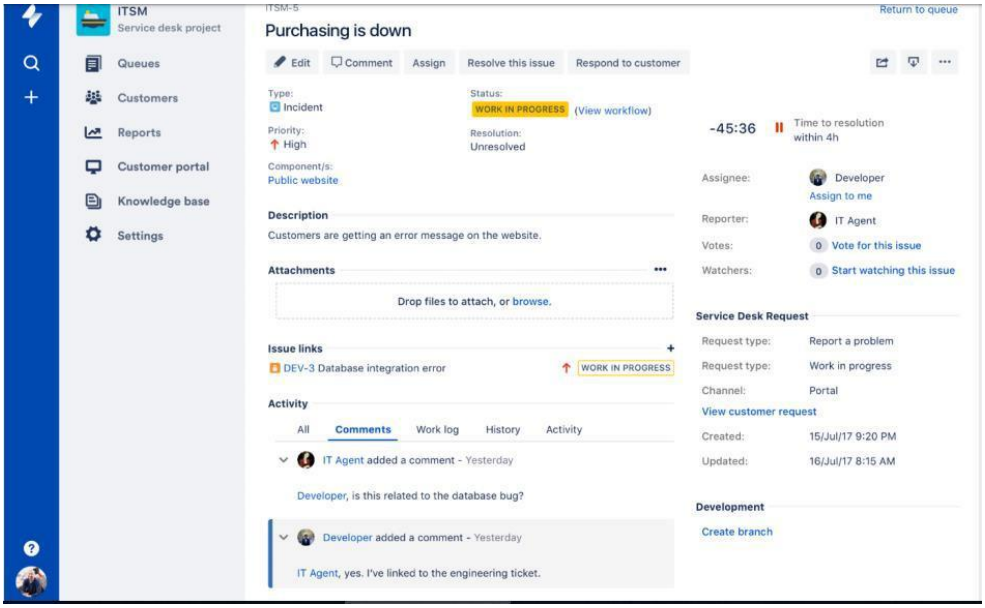

Рисунок 1.13 – Відкрите завдання у JIRA

Продукти та програми, побудовані на платформі JIRA, допомагають командам планувати, призначати, відстежувати, звітувати та керувати роботою. Платформа JIRA поєднує команди у всьому: від гнучкої розробки програмного забезпечення та підтримки клієнтів до управління списками покупок та сімейними справами (рис.1.13).

Чотири продукти побудовані на платформі JIRA: JIRA Software, JIRA Service Desk, JIRA Ops та JIRA Core. Кожен продукт поставляється зі вбудованими шаблонами для різних випадків використання та легко інтегрується, тому команди з різних організацій можуть працювати краще разом.

Розглянемо основні характеристики JIRA Software. Робочі процеси є послідовний шлях від зародження ідеї до її реалізації.

Схема базового робочого процесу показано на рис. 1.14.

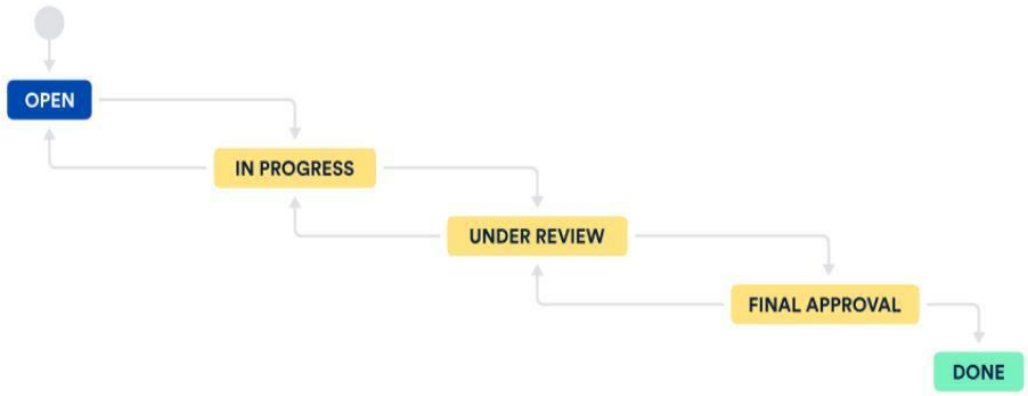

Рисунок 1.14 – Jira Workflow

У цьому випадку Open, Done та мітки між ними відображають статус, в якому може виникнути проблема, а стрілки – потенційні переходи від одного стану до іншого.

Робочі процеси можуть бути простими або складними, з умовами, тригерами, валідаторами та функціями повідомлення.

Рекомендується для адміністраторів, які починають використовувати JIRA Software, зберігати максимально простий варіант своїх робочих процесів до тих пір, поки при реалізації проекту не виникне потреба у ускладненні конфігурації робочого процесу [8].

Таблиця 1.5 містить відомості про важливі характеристики продукту Jira Software, а саме:

- призначення;
- потенційних користувачів;
- сфери застосування;
- можливості інтегрування коїться з іншими системами;
- доступні опції для хостингу.

Таблиця 1.5 – Огляд JIRA Software

Призначення продукту	Планування, відстеження та випуск програмного забезпечення світового рівня.
Users	Software developers Project managers SCRUM masters
Use Cases	Відстеження помилок Управління проектами Менеджмент продукту Управління процесами Управління завданнями Розробка програмного забезпечення Agile розробка програмного забезпечення
Інтеграція	Confluence Bitbucket Slack GitHub
Hosting options	Cloud, Server, Data Center

Розглянемо позитивні сторони використання JIRA Software:

- хороша видимість (visibility). Одним із факторів, що уповільнюють здійснення будь-якого проекту (не тільки в галузі розробки програмного забезпечення, а й в інших професійних сферах, та й просто у житті), є відсутність чіткої постановки завдань та вибудовування ієрархії їх виконання. JIRA усуває цю проблему, оскільки вона об'єднує команди таким чином, що всі члени команди отримують можливість бачити просування у виконанні завдань іншими співробітниками в режимі реального часу, оскільки завдання мають теги "started" та "completed". Це допомагає всім членам команди знати на якій стадії знаходиться проект, що прискорює роботу над програмним продуктом;

- зручне визначення пріоритетів. Ще одна перевага використання JIRA полягає в тому, що вона дозволяє краще визначити порядок пріоритетів завдань, які стоять перед усіма членами команди, внаслідок чого можна побачити, які завдання необхідно виконати негайно та які можуть бути вирішені пізніше. Це дуже корисно з точки зору дотримання термінів, особливо під час роботи над різними проектами з різними датами завершення;

- підвищення продуктивності. Використовуючи JIRA, члени команди отримують можливість у будь-який момент часу бачити послідовність завдань у списку (backlog), внаслідок чого зменшується час простою, витрачене на обговорення поточних проблем, та підвищується продуктивність роботи. Хоча час простоїв може здатися незначним, проте загальна їх кількість може призвести до перевищення часу, відведеного виконання завдання. Таким чином, JIRA сприяє усуненню цієї проблеми та підвищенню продуктивності праці;

- забезпечення безперервності взаємодії між членами команди, де б вони не були. Ще однією перевагою програмного забезпечення JIRA є те, що JIRA поставляється з доступними мобільними програмами. Це означає, що всі члени команди можуть залишатися на зв'язку не тільки в офісі або вдома через ноутбук, але й використовуючи мобільні телефони та планшети;

- понад 1000 плагінів. JIRA поставляється з більш ніж 1000 додатками, які допоможуть зробити програмне забезпечення ще більш корисним для команд гнучкої розробки, два найбільш популярні з них - GreenHopper та Bonfire [6].

Розглянемо недоліки використання JIRA:

- велика насиченість функціональними можливостями, що створює додаткові труднощі не тільки для користувачів-початківців, але й для просунутих, оскільки процес кастомізації продукту виявляється досить трудомістким через велику кількість настроюваних параметрів системи.

Таким чином, потрібно деякий час, щоб зрозуміти архітектуру програмного забезпечення;

- складний UX/UI. Велика кількість візуальної інформації, а також наявність не завжди інтуїтивно зрозумілих елементів інтерфейсу створює користувачу проблеми взаємодії з JIRA;

- вимога наявності навичок DevOps, знання Scrum, вміти будувати спринт до створення структури роботи;

- відсутність вбудованої функціональності керування календарем;

- висока вартість, яка становить для команди до 10 осіб 10 доларів США на місяць, до 100 осіб – 7 доларів США за кожного члена на місяць.

Для кращої організації спільної роботи разом із Jira в IT-компаніях зазвичай використовують Confluence, оскільки це продукти однієї компанії Atlassian і вони добре інтегровані. Розглянемо переваги та недоліки роботи з Confluence.

Confluence – платформа, що дозволяє публікувати web-сторінки документів у стилі wiki, а також обмінюватися контентом між учасниками команди та організувати обговорення.

Призначення Confluence:

- централізація інформаційних потоків;

- повне обслуговування файлової системи та документації (створення, зберігання, перегляд, обмін, редагування);

- функція пошуку документів або іншої робочої інформації, наприклад, листів;

- вбудований корпоративний календар;

- постановка завдань та контроль процесу виконання;

- корпоративні чати.

Серед недоліків програмного продукту користувачі вказують такі:

- незручна система коментарів;

- погано організована система одночасного редагування текстів;

- не налагоджено відстеження розвитку проекту;

- незручний тайм-менеджмент;
- відсутнє управління ресурсами;
- система ціноутворення; ціна на продукт зростає зі зростанням кількості користувачів [10].

Як видно, можливості Confluence досить великі. Розробники постаралися зважити на всі потреби роботи в офісі.

1.4 Постановка мети і задач дослідження

У якості об'єкту дослідження в нашій роботі виступає система управління проектами, що ґрунтуються на використанні штучного інтелекту та машинного навчання.

Предметом дослідження є методи та засоби моделювання автоматизованої системи управління проектами в компанії, орієнтованої на випуск ШІ продуктів, що містять.

Метою роботи є теоретичне обґрунтування та побудова моделі автоматизованої системи управління проектами на підприємстві.

Відповідно до об'єкту, предмету та мети роботи було сформульовано гіпотезу: для розробки моделі автоматизованої системи управління проектами на підприємстві можуть бути використані існуючі методи та засоби управління проектами.

Мета та гіпотеза дослідження зумовили вирішення таких завдань:

- вивчити сучасні методи та інструменти управління проектами, з'ясувати їх переваги та недоліки, особливості застосування у різних IT-компаніях;
- проаналізувати існуючі інформаційні системи з управління проектами;
- проаналізувати зміст діяльності проектної команди підприємства у процесі створення програмного продукту, який використовує штучний інтелект;

- з'ясувати особливості управління проектами у компанії, які застосовуються методології та інструменти управління проектами;

- проаналізувати організаційну структуру управління проектною командою у компанії, оцінити можливості удосконалення процесу управління проектами;

- виконати проектування та розробити модель АСУ проектами на підприємстві;

- виконати якісну оцінку розробленої моделі із застосуванням таких методик: метод експертних оцінок; SWOT-аналіз;

- порівняльний аналіз рівня автоматизації бізнес-процесів підприємства при використанні моделей управління As Is і To Be.

Проблеми управління проектами розглянуто у багатьох наукових працях – статтях, підручниках, доповідях на конференціях. На сьогоднішній день існує велика кількість досліджень, присвячених методам управління проектами; створено інститут управління проектами – Project Management Institute із дочірніми організаціями по всьому світу, Міжнародна асоціація управління проектами – International Project Management Association; розроблено міжнародні, національні, галузеві стандарти.

Дослідження питань управління проектами відрізняються різноманітністю напрямків. Однак питання моделювання АСУ проектами в ІТ-компаніях є затребуваними. Інтерес до цієї теми залишається постійним через динамічний розвиток інформаційно-комп'ютерної сфери.

У процесі виконання дослідження у подальшому у темі роботи треба розглянути такі підходи та методи:

- загальнонаукові прийоми та способи логічного пізнання: аналіз та синтез, абстрагування;

- системно-структурний, функціональний та формально-логічний підходи;

- аналіз літератури з питань теорії управління проектами, моделювання АСУ проектами, що існують стандартів в управлінні проектами;

- емпіричні методи (індивідуальні та групові інтерв'ю зі співробітниками компанії, спостереження за їх діяльністю протягом процесу створення програмного продукту, мозковий штурм, бенчмаркінг);

- методи структурного та об'єктно-орієнтованого аналізу моделювання інформаційної системи (процесне моделювання, ER-моделювання, опис прецедентів використання, контекстна діаграма) нотаціях IDEF1x, BPMN, UML;

- експертна оцінка; SWOT-аналіз; оцінка рівня автоматизації бізнес-процесів.

2 МЕТОДИ ТА АЛГОРИТМИ УПРАВЛІННЯ ПРОЕКТАМИ

2.1 Алгоритми процесу розробки програмних продуктів та методи управління проектами на підприємстві

Компанія ТОВ «Майстер Маїнд Інк» – молода продуктова компанія на ринку IT-послуг. Вона була утворена в Харкові у 2017 у зв'язку зі зростанням клієнтського попиту на інтелектуальний продукт, тобто платформи та сервіси, створені на основі штучного інтелекту (ШІ) та машинного навчання (МН, Machine Learning).

Портфоліо цієї компанії за порівняно невеликий час її існування склали такі розв'язані задачі:

- рекомендаційна система для купівлі супутніх товарів
- замовник – мережа українських аптек;
- аналітичний сервіс, що виконує прогноз потенційного відтоку клієнтів для підприємств, що працюють у сфері обслуговування (мережі фітнес-клубів, спа-салонів та інше);
- платформа для збільшення продажів у електронній комерції.

В даний час компанія працює над проектом для сфери банківського та біржового бізнесу.

Всі ці програмні продукти використовують методи штучного інтелекту та машинного навчання. Так, наприклад, згадана платформа для збільшення продажів в електронній комерції є самонавчається ІІ, який використовує дані про поведінку користувачів історію їх покупок, щоб стати розумнішими. Його основою є розроблена так звана система свідомих рекомендацій (CRS), побудована за допомогою оригінальних алгоритмів машинного навчання. CRS допомагає ідентифікувати покупки, об'єднані однією метою, і пропонує купити потрібний товар у потрібний час з урахуванням глибокого аналізу користувацьких переваг. Такі товари купуються максимально ймовірно.

Процес виготовлення прикладного програмного забезпечення, що використовує ШІ, передбачає більш складну структуру в порівнянні з процесом виготовлення типового програмного продукту.

Нижче представлено укрупнену блок-схему алгоритму створення будь-якого програмного продукту – як вмісту штучного інтелекту, так і стандартного (рисунок 2.1).

Даний процес включає п'ять стадій: від аналізу умов поставленого завдання до супроводу готового виробу в процесі його експлуатації замовником.

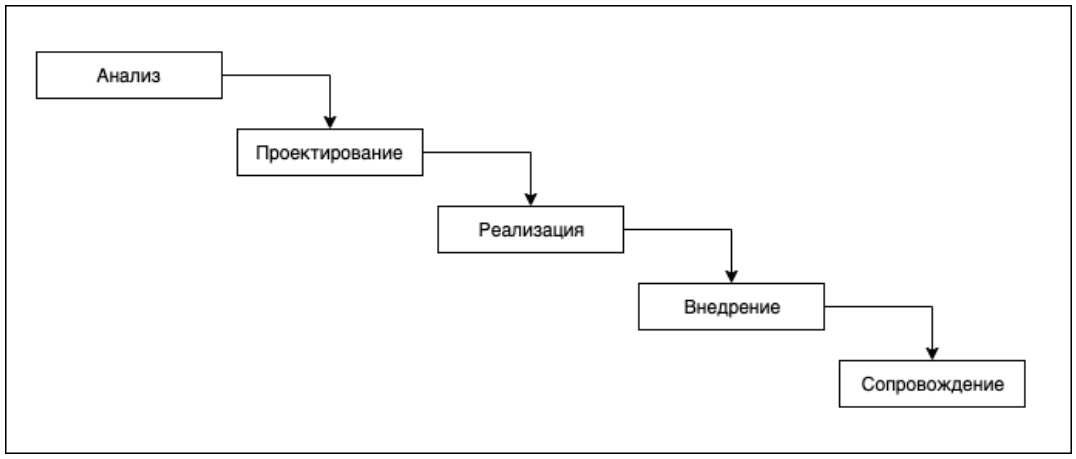

Рисунок 2.1 – Блок-схема робочого процесу розробки програмних продуктів

Однак більш докладна деталізація блок-схеми (рис. 2.2) у разі роботи над створенням сервісів, програм, платформ з використанням ШІ (наприклад, нейромереж) дозволяє виділити в робочому процесі цілий етап, що передує безпосередньо проектуванню програмного рішення – дослідницьку діяльність (Research and Development - R&D), рисунку 2.2 позначений червоним.

Такий тип діяльності притаманний Data Science-проектам та спрямований на дослідження можливих варіантів вирішення поставленого завдання, а також розробку та навчання моделі ШІ для кожного з варіантів.

Особливістю такої діяльності є те, що після етапу аналізу предметної області неможливо однозначно переходити до етапу проектування кінцевого

програмного продукту (як це відбувається при вирішенні стандартних завдань), оскільки вихідні дані, отримані після етапу аналізу, можуть бути переглянуті і уточнені.

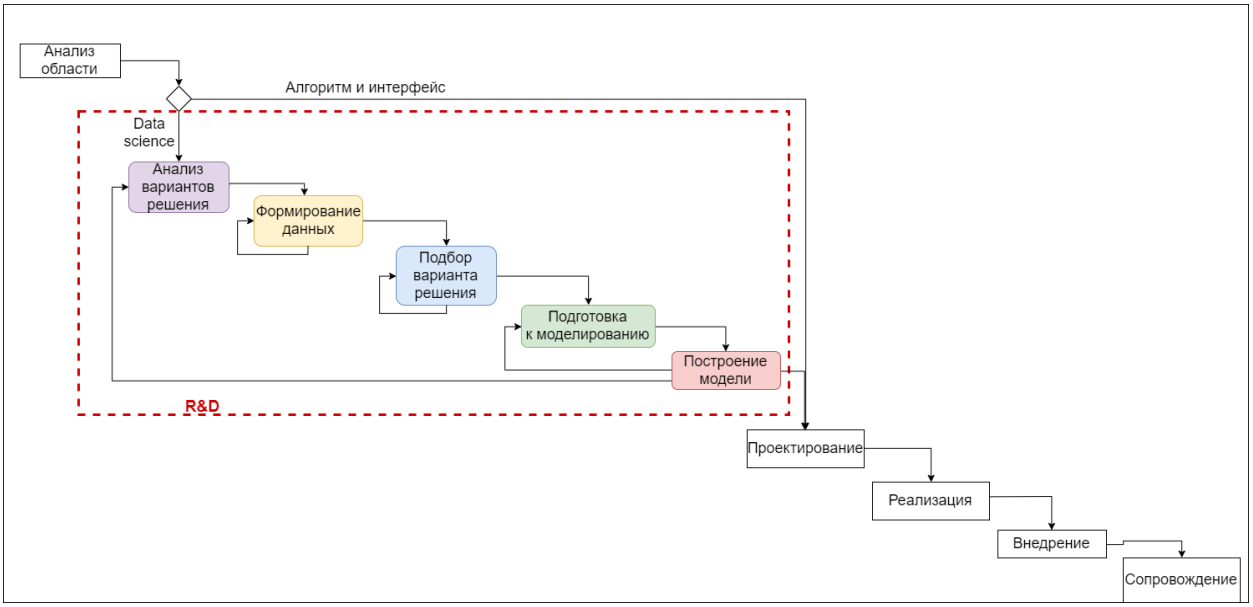

Рисунок 2.2 – Блок-схема робочого процесу розробки програмного продукту, що містить ШІ

Уточнений алгоритм етапу R&D представлений рисунку 2.3.

Як показано рисунку 2.3, на відміну процесу реалізації традиційних рішень, життєвий цикл ШІ-проектів ускладнюється необхідністю проведення низки підготовчих робіт, які мають циклічний характер. Кількість цих циклів наперед невідома і залежить як від ступеня складності поставленого завдання, так і від кваліфікації співробітників на проекті та розвитку їх творчого мислення.

Понад те, можливий результат, у якому детальне вивчення поставленого завдання показує, що для нього немає рішення.

Вузол прийняття рішення, позначений *1, відображає той момент дослідження, коли команда, ґрунтуючись на своєму досвіді та інтуїції, вибирає напрямок, пов'язаний або зі зміною параметрів моделі, або зміною «features», або з пошуком нового варіанта вирішення поставленого завдання.

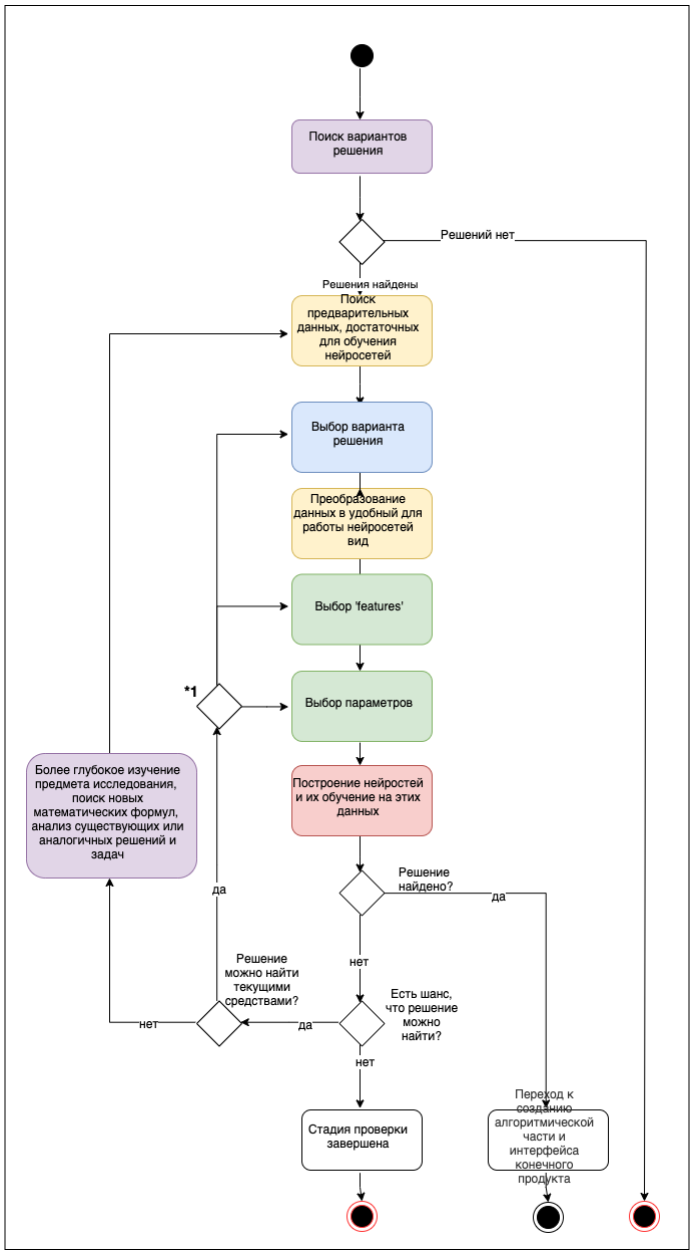

Рисунок 2.3 – Блок-схема алгоритму R&D

Циклічний характер процесу створення ШІ-продукту суттєво відрізняється від Agile-моделей (Scrum, Kanban та ін.), основу яких складають ітерації, на кожній з яких ми отримуємо деяку закінчену частину цільового програмного продукту.

У випадку з ШІ-проектами завершення кожного циклу не завжди означає створення частини або функції кінцевого продукту; навпаки, нерідко буває, що виконавці проекту змушені повернутися до точки нульового відліку та розпочати заново роботу з вихідною інформацією.

Крім того, діяльність R&D ґрунтується на роботі з великими масивами даних, причому протягом усього життєвого циклу проекту.

Ця ключова обставина змусила фахівців Data Science шукати спеціалізовані методи роботи з Big Data.

На даний момент у компаніях, що спеціалізуються на створенні Data Science-продуктів, найбільш поширеною є CRISP-DM – методологія дослідження даних.

Таким чином, діяльність команди в процесі створення ШІ-продукту, складається з дослідницьких дій, які завжди мають унікальний, оригінальний характер, та стандартних дій з проектування, реалізації та впровадження.

Протягом всього процесу менеджмент компанії ТОВ «Майстер Маінд Інк» для управління проектом використовує Jira, а ведення документації виконується за допомогою Confluence.

Ці інструменти містять весь необхідний функціонал для керування завданнями, персоналом та документацією на стандартних проектах, і ідеально підходять для керування етапами проектування та реалізації, але не достатні для керування дослідницькою діяльністю на етапі R&D. Як показало проведене автором вивчення особливостей управління проектами у компанії, на етапі R&D можливості Jira використовуються частково, фрагментарно та неефективно.

2.2 Організація процесу дослідницької діяльності на підприємстві

Розглянемо організаційну структуру компанії ТОВ «Майстер Маїнд Інк». У компанії працює 15 осіб, розподілених між такими відділами: фінансовий –1 чол., відділ DevOps – 2 чол., фахівці за даними (Data Engineering) – 2 чол., відділ R&D – 6 чол., технічний директор CTO, виконавчий директор CEO , Product Owner; розробник сайтів готового продукту – 1 чол.

Співробітники відділу R&D володіють такими знаннями та вміннями.

Вища освіта з прикладної математики, алгоритми та структури даних, Machine Learning, фінансово-економічні знання, мови програмування та фреймворки Python, Flask, Docker Swarm, Kubernetes, TenzorFlow, Cuda, C++, NodeJS, ClickHouse, MySQL, Redis, Kafka, R language, Bash script.

Фахівці Data Engineering мають такі знання та вміння: вища технічна освіта, алгоритми та структури даних, мови програмування та фреймворки C++, Cuda, MySQL, MSSQL, PostgreSQL, ClickHouse, MongoDB, RabbitMQ, Redis, Kafka, NodeJS, Bashscript.

Професійні знання та вміння співробітників відділу DevOps такі: вища математична освіта, Docker Swarm, Kubernetes, Kibana, Grafana, Git, NginX, Lua, Bash Script.

Директори компанії мають вищу технічну освіту, володіють деякими з перерахованих мов програмування та фреймворків.

Начальник R&D-відділу має ступінь PhD з математики.

Крім того, технічний директор виконує обов'язки архітектора проектів, а також керує відділами.

Виконавчий директор CEO поєднує обов'язки Project Manager.

Його завдання такі:

- постановка завдань для проектної команди;
- призначення виконавців для конкретних завдань;
- відстеження виконання;
- створення проектної документації тощо.

R&D-процес має на увазі велику залученість проектної команди завдання, які за Agile-методологіями (та іншими поширеними загальноприйнятими методологіями) виконує проектний менеджер.

Як видно з наведених відомостей, співробітники компанії є висококваліфікованими спеціалістами, і мають перехресні вміння, що є необхідною умовою роботи у сфері виробництва ШІ-продуктів.

У процесі роботи над програмним продуктом всі співробітники, включаючи менеджмент, тісно пов'язані між собою, і підтримують постійний зв'язок із замовником. Щоранку проводяться стендап-мітинги, на яких обговорюються результати минулого дня та плануються завдання на поточний день.

При цьому також приймаються рішення про відхилення проміжних завдань або зміну постановки задачі, якщо не вдалося знайти її розв'язання. Щотижня проводяться брейн-шторми, де на порядок денний виносяться завдання, виконання яких здається неможливим. В обговоренні пошуку можливих варіантів рішення бере участь весь склад компанії, в особливо скрутних випадках можуть залучатися сторонні фахівці.

Слід зазначити, що результати мітингів, що проводяться, практично не документуються.

Виникають у процесі обговорення думки, ідеї, висновки, пропозиції записуються на листочках, за допомогою стікерів на дошці, а часом взагалі не фіксуються. Поки компанія невелика, подібна практика не є катастрофічною, проте на сьогоднішній день існує великий недогляд у процесах управління проектами.

Схема організаційної структури підприємства у вигляді представлена на рисунку 2.4.

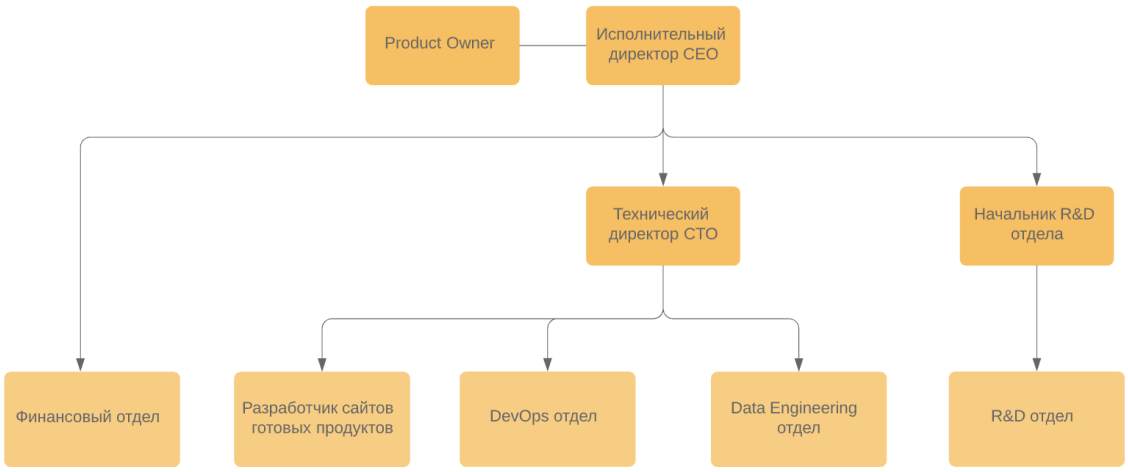

Рисунок 2.4 – Загальна організаційна структура підприємства ТОВ «Майстер Маїнд Інк»

Розглянемо, як відбувається створення програмного продукту у компанії з погляду поділу праці.

Аналіз організації процесу розробки проектів у компанії Майстер Маінд Інк на етапі R&D показав, що вона має складну нелінійну структуру (рис. 2.5).

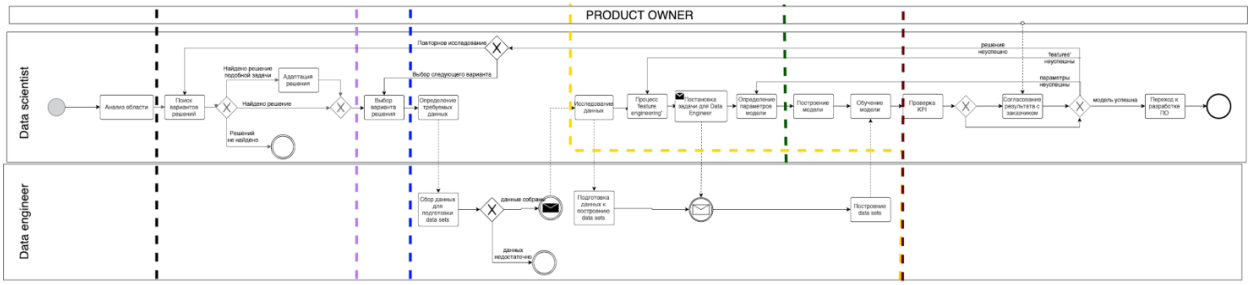

Рисунок 2.5 - Модель бізнес-процесів етапу R&D на підприємстві ТОВ «Майстер Маїнд Інк»

Подана модель бізнес-процесів відображає розгортання процесу розробки продукту в часі та розподіл робіт між виконавцями, а також зв'язки між ними.

Рисунок 2.5 є наступним рівнем декомпозиції блок-схем алгоритму R&D, виконаних на рисунках 2.2 і 2.3. Бо більше докладна деталізація ускладнює схему дослідницької діяльності, виділення чітких етапів (як це було зроблено в рисунку 2.2), стає дуже скрутним, оскільки реальний, фізичний процес створення програмного продукту є нелінійним, ситуативним, яке підпроцеси можуть бути паралельні чи перетинатися.

Процес створення та навчання нейромережі розподілено між трьома сторонами, що беруть участь: Product Owner (особа, яка приймає рішення з боку замовника), фахівці з Data Science та інженери за даними (Data Engineer). Як зазначалося, бувають випадки, коли виконавці змінюються ролями і тимчасово беруть він обов'язки іншого члена команди.

Простежимо організаційні зв'язки між відділами та співробітника на підприємстві ТОВ «Майстер Маінд Інк» на кожному етапі алгоритму R&D.

Як було показано рисунку 2.3, R&D-діяльності передує аналіз предметної області. Результатами цього аналізу мають бути:

- визначення мети товару;
- визначення бізнес-вимог;
- визначення KPI для кожної бізнес-вимоги;
- визначення можливих способів задоволення бізнес-вимог,

саме: визначення проектною командою стратегії вирішення кожної бізнес-вимоги – за допомогою побудови ШІ або алгоритмічних методів.

Попередня орієнтовна діяльність у проектах з ШІ має значення більше, ніж у будь-яких інших видах проектів. На цій стадії відбувається формулювання критеріїв, яким надалі відповідатиме готовий продукт.

Реалізація будь-якого проекту завжди починається з формулювання потреби замовника/ринку у ньому. Product Owner повинен мати уявлення про те, як цей продукт чи його нова функція повинні працювати зрештою. Наприклад, впровадження деякого сервісу запобігає відпливу відвідувачів або простимулює продажі інших продуктів.

Формулювання такої потреби не претендує на повний опис проекту, вона лише означає проблему: «наші клієнти повинні зрозуміти, як вони розпоряджаються своїм бюджетом», «якщо ми додамо функцію передбачення години пік в аеропорту, то клієнти охоче купуватимуть додаток».

Тут потрібно викласти мети товару, виходячи з яких формуються бізнес-вимоги, тобто. набір тієї функціональності, що визначає призначення.

На цьому етапі всі учасники проекту пропонують ідеї, які можна використовувати під час вирішення сформульованої проблеми, відбувається створення деяких первісних нарисів можливого рішення. Цим етапом зазвичай керує фахівець Data Science, але важливо, щоб брали участь абсолютно всі учасники роботи на проекті, включаючи менеджмент, оскільки можливі рішення та продуктивні ідеї можуть лежати в різних галузях знань.

В результаті цього етапу зазвичай стає ясно, наскільки глибоким буде процес дослідження даних.

Потім слід визначення кількісних показників, якими визначатиметься, чи досягнуто бізнес-вимоги. Цей етап полягає у спільному визначенні обсягу

та ключових показників ефективності проекту (KPI). Потім ці KPI повинні уточнені та переведені у вимірні. Можливо, вдасться отримати дуже чіткі показники, такі як «прогнозування очікуваного CTR оголошення з наближенням не менше X% у щонайменше Y% випадків для будь-якого оголошення, показаного не менше тижня, і для будь-якого клієнта з більш ніж двома місяцями тимчасових даних» . Однак у деяких випадках необхідно використовувати якісні показники, наприклад, «час, необхідний вивчення теми з використанням згенерованих розширених запитів, буде скорочено і/або поліпшиться якість результатів проти вихідними запитами».

Це особливо вірно, коли модель призначена для надання допомоги людині в якійсь складній діяльності.

Ключові показники можуть бути уточнені у зв'язку з наявними часовими рамками та ресурсами. Для їх уточнення потрібний подальший збір даних.

Зазвичай ці метрики (KPI) визначаються проектною командою спільно

Product Owner у точних кількісних показниках. Але залежно від ресурсів та тимчасових обмежень вони можуть бути переглянуті після консультацій із замовником. Оскільки будь-який зворотний зв'язок означає додаткову витрату часу, то проектна команда може спробувати знайти додаткові жорсткі метрики, на які можна спертися для продовження роботи над продуктом, і уникнути додаткових консультацій замовником.

Визначення обсягу робіт цьому етапі особливо важливо, оскільки дослідницькі проекти мають тенденцію подовжуватися і збільшуватися у розмірах і масштабах у міру появи нових можливостей під час проведення досліджень.

Процес визначення значень KPI є таким чином ітераційним. Він завершується тоді, коли знайдені значення задовольнять клієнта. Тут був помічений наступний факт, який відображає певний конфлікт бізнес-інтересів замовника та тих можливостей програмного продукту, як їх оцінює проектна команда. Бізнес завжди прагне мати максимальні показники KPI. Розробники стверджують, що програмний продукт об'єктивно не в змозі (як і будь-яка

модель) відобразити реальну ситуацію (в предметній галузі) на 100%. Наприклад, якщо система передбачає поведінку покупців з точністю менше 95%, то така система, на думку більшості замовників, є марною.

Однак у багатьох випадках ретельний аналіз та обговорення припущень про продукт можуть призвести до появи дуже цінних рішень, які можуть виявитися не такими технічно складними, як здавалося на першій ітерації вироблення KPI.

Визначити обсяг робіт на даному етапі також є важливим із психологічної точки зору. Дослідник, який починає вивчати отримане завдання, прагне вивчити якомога більшу кількість статей та книг. Тому відсутність чітко окреслених меж часу, ресурсів та цілей проекту може призвести до довгих тижнів або місяців, витрачених на розробку прекрасних моделей, які зрештою не відповідають реальній потребі, або призводять до неврахування деяких KPI, які могли бути визначені раніше.

Аналіз рішення (позначений бузковим кольором на рисунках 2.2, 2.3 та 2.5).

На цьому етапі перед проектною командою стоять завдання вивчення літератури та існуючих готових рішень.

Дослідник аналізує як академічну літературу, так і існуючі програмні рішення та інструменти.

Глибина занурення в літературу залежить від особливостей проекту: є фундаментальним, визначальним подальшу стратегію компанії чи майбутній програмний продукт призначений на вирішення разової проблеми. Також слід відповісти на запитання: чи планується публікація результатів дослідження з цього питання в академічних виданнях; чи планує дослідник стати експертом команди з цієї теми тощо.

Наприклад, проект призначений для того, щоб допомогти відділу продажів краще прогнозувати відтік клієнтів та збитки від відтоку. Для вирішення поставленої задачі потрібно добре знання теорії випадкових процесів, на якій будуються багато загальних рішень подібних проблем.

Потрібна глибина розуміння залежить від технічних аспектів проблеми. Деякі аспекти можуть бути передбачені заздалегідь, а деякі виявлені тільки пізніше.

Наприклад, якщо продукт, що розробляється, повинен бути інтегрований в готове програмне середовище, написане мовами Java і Scala, то доводиться використовувати ті технології, які застосовуються в готовому продукті. І якщо розробники не володіють цими мовами, їх доведеться вивчити.

При вивченні літератури фахівець Data Science повинен підготувати для обговорення в команді не лише обрані варіанти вирішення проблеми, але зробити короткий огляд усієї області та всіх розглянутих рішень, пояснюючи переваги та недоліки кожного рішення та обґрунтувати свій вибір.

Після того, як знайдено відповідний математичний апарат, знайдені напрямки рішень необхідно оцінити з точки зору способу реалізації та складності цього у виробництві.

Цілі продукту та бізнес-вимоги до нього, а також структура та характеристики запропонованих варіантів рішень є обґрунтуванням для вибору способу зберігання та обробки даних, здатність до масштабування по горизонталі та вертикалі (scalability) та приблизну оцінку вартості проекту.

Це важлива перевірка, яку необхідно виконати на цьому етапі, оскільки обробка даних та розробка програмного забезпечення можуть починатися паралельно з розробкою моделі. Крім того, запропоноване рішення може виявитися неадекватним або занадто дорогим з технічної точки зору, і в цьому випадку це має бути виявлено та усунено якнайшвидше. Коли технічні питання розглядаються до початку розробки моделі, знання, отримані на етапі дослідження, можуть використовуватися для пропозиції альтернативного рішення, яке могло б краще відповідати технічним обмеженням. Це ще одна причина, через яку етап дослідження має закінчитися виробленням спектру рішень, а не обмежуватися одним

Підготовка даних та вибір рішення (синій та жовтий колір на рисунках 2.2, 2.3, 2.5);

На цьому етапі проектна команда визначає набір даних, які будуть потрібні надалі для навчання нейронних мереж; збирання первинних даних, обробка їх (очищення від сміттєвої інформації), перетворення на необхідний формат. Зберігаються підготовлені дані разом із запитами у базах даних компанії.

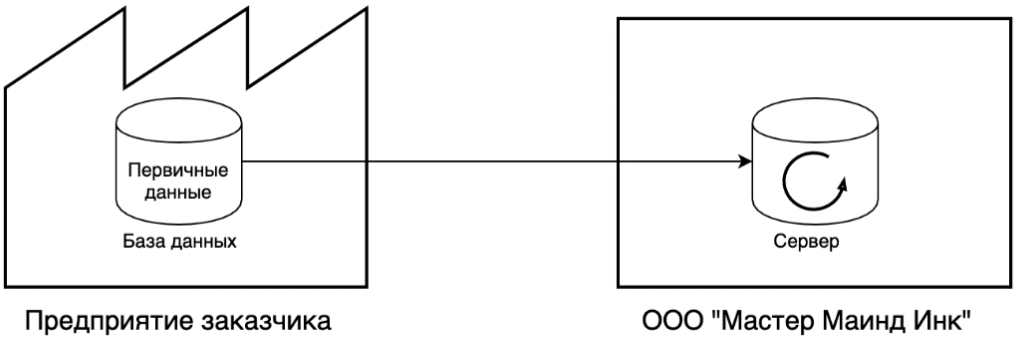

Рисунок 2.6 – Схема взаємодії баз даних

Процес підготовки даних часто супроводжується вивантаженням великих масивів даних (Big Data) із баз даних замовника на сервер компанії (якщо дозволяють умови конфіденційності). Надалі відбувається обробка Big Data у формат, зручний для швидких запитів та складних обчислень. Ці дії необхідні початку процесу дослідження даних і іноді займають більше часу, ніж очікувалося.

Далі розпочинається процес дослідження даних.

Тепер фахівець Data Science може оперувати фактичними жорсткими KPI та показниками моделі.

Однак і тут треба дотримуватися динамічної рівноваги між дослідженням та розробкою; навіть маючи на увазі чіткі KPI, корисно не втрачати на увазі деякі, здавалося б, невідповідні напрямки рішення.

Оброблені дані повинні мати формат зручний для роботи відділу Data Engineering. Однак можуть бути виявлені деякі недоліки сформованого робі-

тника масиву даних (наприклад, даних, наданих замовником, може виявитися недостатньо). Data Engineer даних має бути готовим до такої ситуації, і шукати додаткове джерело даних.

Слід зазначити, що хоча процес підготовки даних і виглядає відокремленим від процесу теоретичного дослідження та аналізу варіантів рішень, вони зазвичай або виконуються паралельно, або чергуються між собою.

Підготовка до моделювання (зелений колір на рисунках 2.2, 2.3, 2.5). Підготовка до початку розробки моделі значною мірою залежить від наявного технічного забезпечення та обсягу технічної підтримки, доступної Data Scientist. Можливо, Data Scientist створить новий репозиторій коду і запустить локальний сервер Jupyter Notebook або запросить потужнішу хмарну машину для виконання обчислень.

Також може бути, що співробітники відділу Data Science створять код для управління версіями даних та моделей або для відстеження та управління експериментом. Можливо, що така функціональність вже існує у використовуваних сервісах, тоді буде потрібно створення деяких налаштувань для розподілу ресурсів або налаштування пакетів користувача і т. д.

Цей етап пов'язаний не лише з технічною підготовкою, а й підготовкою 'features' та параметрів для майбутньої нейромережі.

Розробка та навчання моделі (червоний колір на рисунках 2.2, 2.3, 2.5). Завданнями цього етапу є розгортання моделі, навчання (Machine Learning) та безперервний моніторинг результатів навчання.

У підготовленому на попередньому етапі середовищі з вибраними параметрами створюється модель нейромережі. Фахівці відділу Data Engineering передають до відділу Data Science набори даних та створені до них запити, підготовлені відповідно до набору 'features'. Відділ Data Science завантажує ці дані у навчальне середовище, яке використовує ці дані для навчання моделі. Фахівці відділу Data science здійснюють моніторинг результати навчання, використовуючи функціональність сервісів. Після того, як модель вважається навченою, можна переходити до модельного тесту.

При розробці та навчанні моделі різні її версії (і обслуговуючий масив даних) повинні постійно перевірятись на відповідність заздалегідь встановленим KPI.

Це дає можливість оцінити рівень просування в навчанні, а також дозволяє відділу Data Science вирішити, коли модель працює досить добре, щоб гарантувати повну відповідність виробленим на початковому етапі KPI.

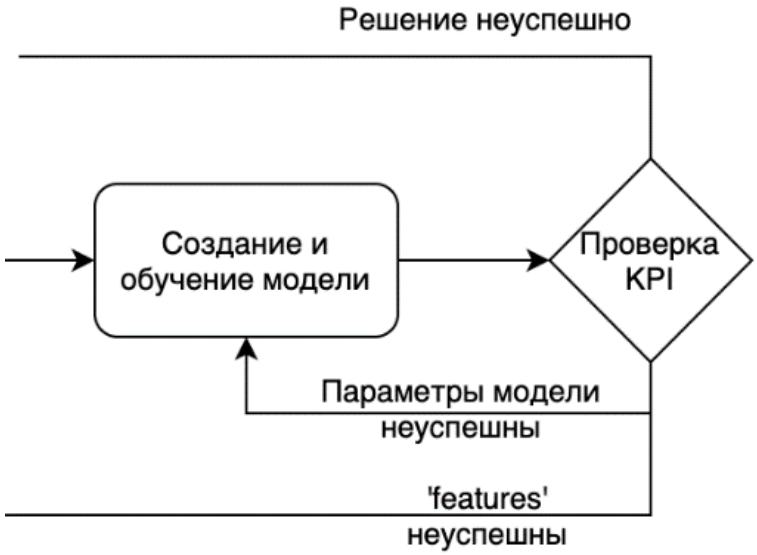

Рисунок 2.7 – Алгоритм ухвалення рішення про успішність моделі

Якщо результати навчання моделі не відповідають очікуваним показникам, існує кілька причин такого результату.

Спеціаліст відділу Data Science аналізує результати навчання нейронної мережі та приймає рішення про подальші дії: змінити параметри, змінити «features», або даний варіант моделі не придатний для вирішення задачі і треба переходити до вибору наступного варіанта моделі.

Іншим можливим результатом невдалого навчання є перегляд мети товару. У окремих випадках зміна мети тягне у себе незначні зміни у технічної реалізації проекту. Але загалом це означає, що ми повертаємося на фазу аналізу предметної області.

Не виключено найбільш екстремальний варіант – скасування проекту; якщо фахівець за даними впевнений, що всі напрямки досліджень були вивчено, а проектний менеджер упевнений, що продукт вичерпав свій потенціал, можливо, настав час перейти до іншого проекту.

Якщо навчання моделі пройшло успішно і вона реалізує задані KPI, то етап R&D можна вважати успішно закінченим і переходити до розробки алгоритмічної частини кінцевого продукту, а також створення інтерфейсу користувача.

У ситуаціях, коли розбіжність між отриманими та заданими значеннями KPI не є критичною, потрібно більш ретельний їх перегляд та узгодження з Product Owner.

Необхідно переконатися, що поточне значення показника перебуває у інтервалі прийнятного розкиду. При цьому необхідно враховувати вимоги до продукту та інтереси клієнта.

Коли Product Owner переконаний, що модель відповідає заявленим цілям проекту (задовільно), команда може приступити до його виробництва програмного продукту.

З виконаного аналізу організації процесу дослідницької діяльності у компанії ТОВ «Майстер Маінд Інк» можна дійти невтішного висновку у тому, що структура цього процесу перестав бути лінійної. Розгортка Agile-моделі життєвого циклу проекту має лінійну архітектуру, як представлено рисунку 2.8.

Рисунок 2.8 – Розгортка Agile-моделі життєвого циклу проекту

Моделі Crisp DM, на відміну від Agile-моделей, можна подати у вигляді дерева рішень, де кожна гілка відображає новий цикл розробки.

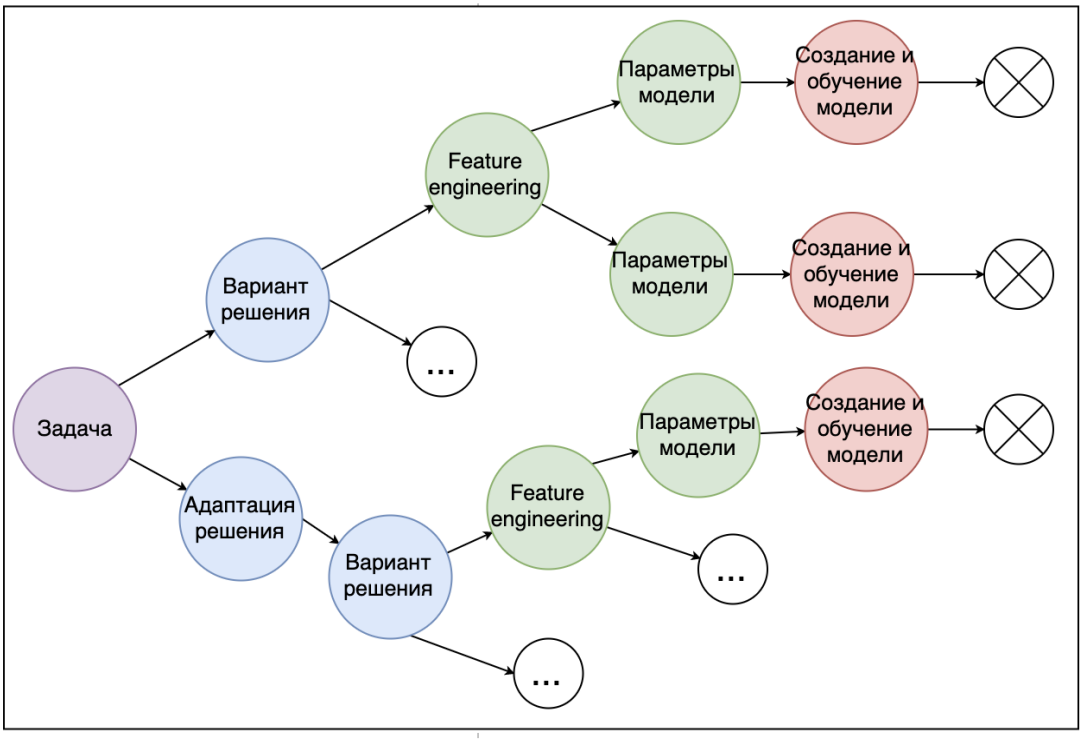

Рисунок 2.9 – Дерево рішень моделі Crisp DM

Кореневим вузлом дерева є певні межі проекту у вигляді цілей, бізнес-вимог та набору KPI. Вузлами другого рівня можна уявити знайдені чи адаптовані можливі рішення поставленого завдання. Далі кожен вузол має розгалуження на розроблені 'features'. З даних вузлів розходяться гілки вибраних параметрів. Кожен вузол належить одному з розглянутих раніше фаз етапу R&D.

2.3 Модель системи управління проектами As Is

Виконаний аналіз процесу створення програмного ШІ містить продукту, а також організаційної структури процесу в компанії ТОВ Майстер Маінд Інк дозволяє побудувати модель As Is (Як є) процесів управління в компанії.

Модель зображена на рисунку 2.10 додатка Б. Вона відображає управлінські дії проектного менеджера в інструментах Jira та Confluence, а також дії інших співробітників компанії, які приймають на себе функції Project

Manager відповідно до поточних потреб проекту. Побудована модель відображає процеси управління, що супроводжують усі фази R&D-діяльності.

Процеси розподілені між Project Manager, відділом Data Science та відділом Data Engineering.

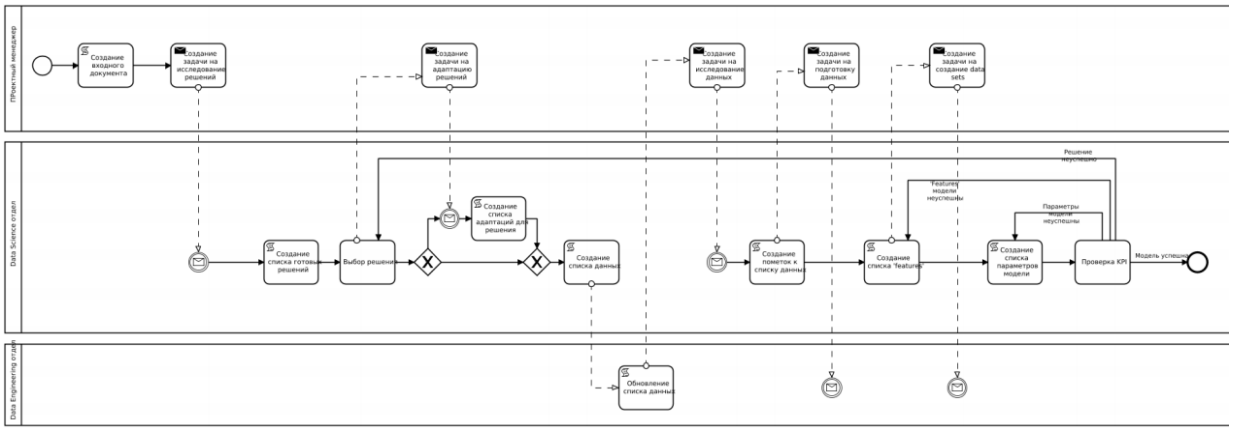

Рисунок 2.10 – Модель управління процесами As Is

З точки зору управління даний R&D етап можна представити так.

Розглянемо докладніше представлену модель процесу управління проектом, що функціонує у компанії ТОВ «Майстер Маїнд Інк»:

- створення вхідного документа. Цей процес належить фазі аналізу предметної галузі. Тут команда збирається на нараду і записує на листку або дошці всі прийняті на нараді рішення, а саме: список цілей, список бізнес-вимог, список KPI, а також позначок кожній бізнес-вимоги, за допомогою чого він вирішується (ШІ або алгоритм). Після наради PM робить фотографію дошки чи листочка і переносить усі записи в Confluence;

- створення завдання вивчення рішень. Після закінчення наради PM створює завдання Jira, і закріплює завдання за співробітником Data Science-відділу. Співробітник, який отримав завдання, розпочинає її виконання;

- створення переліку готових рішень. Після дослідження співробітник створює список знайдених рішень у вигляді презентації та робить доповідь для проектної команди;

- вибір рішення. Після прослуховування доповіді команда обговорює її та приймає рішення, які з представлених рішень перспективні, які треба відкинути, а які залишити «про запас»;

- створення завдання адаптацію решения. Якщо жодне з обраних рішень не можна застосувати в готовому вигляді, то PM створює завдання в Jira на адаптацію рішення та закріплює за співробітником Data Science-відділу;

- створення списку адаптацій на вирішення. Співробітник Data Science-відділу, який отримав та виконав завдання, створює список адаптацій і заводить його в Confluence;

- створення списку даних. Після вибору варіанта рішення проектна команда збирається на нараду та створюється список даних, необхідні роботи над проектом. Цей пункт також може виконати співробітник Data Science-відділу самостійно. Цей список створюється окремим документом Confluence;

- оновлення списку даних. Співробітник Data Engineering-відділу роздруковує документ, створений у п.7 та займається збором даних. На роздрукованому документі створюються позначки у тому, які дані вдалося зібрати у якому обсязі;

- створення завдання дослідження даних. Після успішного збору даних PM створює завдання Jira для начальника Data Science-відділу;

- створення позначок до списку даних. Начальник Data Science-відділу досліджує дані та готує документ із позначками до зібраного списку даних. Цей документ створюється в Confluence;

- створення завдання підготовку даних. PM створює завдання Jira для співробітника Data Engineering-відділу на підготовку даних. Співробітник Data Engineering-відділу отримує завдання разом із документом з п.10;

- створення списку "Features". Після створення позначок до списку даних співробітники Data Science-відділу проводять нараду з метою

визначення списку Features. Цей список заводиться в Confluence одним із співробітників Data Science-відділу;

- створення завдання підготовку наборів даних. Після виконання п. 12 PM створює завдання Jira для співробітників Data Engineering-відділу с з прикріпленим документом з п.12;

- створення списку параметрів моделі. Після виконання завдання, поставленої п.12, співробітники Data Science-відділу створюють список параметрів для поточної моделі;

- перевірка KPI. Після створення та навчання моделі з вибраними параметрами та вибраним списком Features співробітники Data Science-відділу перевіряють результати навчання моделі засобами Jupiter Notebook. Після цього показники поточного циклу не зберігаються як документа.

Як можна побачити, проектний менеджер створює завдання за допомогою Jira. Найбільшу кількість інформації, що підлягає документуванню, виконує робота відділу Data science.

Оскільки проектний менеджмент входить у прямі обов'язки дослідників і розробників, то документування виконується ними непрофесійно. Вони просто здебільшого нехтують питаннями безпеки потенційно корисної інформації.

В результаті проектного менеджера немає візуального відображення стану справ у будь-який момент часу. Він може дізнатися про статус виконання завдань та дані щодо поточної гілки тільки в усній формі.

Виконане дослідження організації управління у ТОВ «Майстер Маїнд Інк» дозволило зробити такі спостереження. Більшість документації немає чітку структуру і створюється на розсуд виконавця.

Не вся документація здійснюється за допомогою корпоративної системи Confluence. Деякі документи заводяться в особистих облікових записах співробітників усередині інших систем або створюються у паперовому вигляді.

Документи, створені поза Confluence, не завжди переносяться до Confluence;

Багато документів зовсім не створюються, обмін робочою інформацією відбувається або в усній формі, або у паперовому вигляді.

2.3 Висновки до розділу

Вивчення методів управління проектами, прийнятих у компанії ТОВ «Майстер Маінд Інк», а також алгоритму R&D-діяльності дозволили зробити такі спостереження та висновки.

Компанія стикається з такими проблемами, як: неповна документація.

Ця проблема виникає у зв'язку з тим, у зв'язку з тим, що створення документації зазвичай є коло обов'язків менеджерського складу та бізнес-аналітиків. У разі дослідницької діяльності документація є технічної і проектний менеджер неспроможна її коректно заповнити без допомоги (диктування) технічних фахівців. Тому розробники виробляють її самостійно. Але оскільки такого роду обов'язки не є традиційними для них, необхідна документація формується неякісно, і іноді навіть взагалі не створюється, оскільки виконавці не хочуть витрачати на неї час і покладаються на свою пам'ять.

Таким чином, цінні думки, ідеї, творчі рішення можуть бути загублені. Через незбережену документацію в деяких випадках проектній команді доводиться витрачати додатковий час на повторне обговорення завдань, неуніфікований формат.

У зв'язку з описаною вище, документація створюється в різних форматах. Це ускладнює процес її спільного використання тощо.

Слабка візуалізація. Дерево рішень не візуалізується та ніде не зберігається. Це може впливати на загальне уявлення про стадії проекту, а також ускладнює проектному менеджеру завдання звітності замовникам про виконану роботу.

Неповний облік часу. Завдання повторюються від гілки до гілки, деякі завдання є незначними (з точки зору часу), проектна команда до компанії невелика і кожен знає свої обов'язки, тому деякі завдання можуть не створюватися в інструменті управління проектами. Це ускладнює облік часу, оскільки в Jira зберігатиметься неповна інформація про завдання. Також це може спричинити проблему «втрати» деяких завдань.

Модель CRISP-DM є на даний момент єдиною прийнятною для Data Science проектів, але вона має певні недоліки, які впливають на процес управління R&D етапом. До них відносять відсутність засобів підтримки.

Ця модель стала застосовуватися нещодавно до таких проектів. І самі Data Science проекти з'явилися нещодавно. Тому існуючі інструменти управління ще не встигли адаптуватися під потреби Data Science-проектів, а нові не з'явилися на ринку.

Відсутність критерію часу. Модель не передбачає врахування часу або обмеження часу кожного циклу.

3 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ПРОЕКТАМИ У КОМПАНІЇ

3.1 Модель автоматизованої системи управління проектами

Вивчення методу організації управління проектами у компанії ТОВ «Майстер Маінд ІНК» показало, що у підприємстві є суттєві змогу підвищення ефективності процесу управління проектами. Багато в чому ручна модель управління, яка встановилася в компанії, не цілком адекватна процесам створення проектів, які виконує команда.

Для роботи проектної команди необхідний єдиний інформаційний центр, з допомогою якого можна вирішувати такі завдання управління:

- зберігати проектну документацію;
- вести протоколи мітингів, стендапів, мозкових штурмів,
- виробничих нарад;
- зберігати презентації доповідей працівників;
- оповіщати співробітників про події та плани;
- створювати реєстри завдань, бізнес-процесів, власних розробок;
- створювати власну бібліотеку готових рішень щодо профілю компанії;
- створювати «запасники» цікавих ідей, методів, процедур, які можуть стати в нагоді в майбутньому;
- призначати завдання та роздавати доручення;
- здійснювати зворотний зв'язок за поставленими завданнями та дорученнями.

Мета даного етапу дослідження – розробити модель такого інформаційного центру у вигляді автоматизованої системи управління проектами. Ця АСУ проектами має бути інтегрована у існуючу систему управління проек-

тами та за рахунок синергії підвищити ефективність управління проектами, і, отже, роботи компанії загалом.

Аналіз методів та алгоритмів процесу управління проектами у компанії ТОВ «Майстер Маінд Інк» показав, що управління на етапі R&D істотно відрізняється від процесу управління на етапі виконання програмного продукту.

Система, що моделюється, повинна бути заснована на програмних продуктах, які на даний момент використовуються на підприємстві ТОВ «Майстер Маінд Інк»: Jira та Confluence. Цей продукт не замінюватиме поточні системи, а буде вбудований у неї з метою автоматизації рутинних процесів. Продукт не є самостійною та незалежною системою, а є плагіном для системи Jira, тому має бути адаптований до роботи з нею.

Існують такі методи адаптації програмного забезпечення:

- параметрична адаптація, пов'язана з налаштуванням параметрів програми. Це найпростіший спосіб адаптації; він передбачає зміну значень показників, регулюючих роботу програми. За допомогою параметричної адаптації можливо проводити необхідне налаштування функцій та компонентів програми, та відбирати стратегії поведінки із запропонованого набору стратегій;

- функціональна адаптація передбачає зміну функцій програмного забезпечення у встановлених межах. Функціональна адаптація може виконуватися спільно з параметрами. Структура та організація ПЗ у своїй залишаються постійними;

- організаційна адаптація - перенастроювання потоків та процесів у системі. Організаційна адаптація означає перерозподіл потоків та процесів системи як її внутрішніх ресурсів, без зміни її структури. Може поєднуватись з функціональною адаптацією;

- структурна адаптація – зміна структури системи. Використовуючи цей спосіб адаптації, ми робимо модифікацію або заміну одних структурні елементи системи, або алгоритмічні модулі, на інші. Програма, що

адаптується, стає більш адекватною розв'язуваним завданням. У цьому можливе використання всіх попередніх видів адаптації системи;

- розмноження – породження собі подібних нащадків. Адаптація розмноженням є ефективним методом адаптації. Система виробляє собі подібну, але з наявністю вільних ресурсів і здатністю змін.

- розвиток – спрямований процес еволюції систем. Адаптація програмного забезпечення, що відбувається через розвиток, подібна до процесів еволюції живих систем. Адаптоване ПЗ проходить процес еволюції через накопичення інформації про себе, зовнішнє середовище, розв'язувані завдання. Проходячи через етапи зародження, становлення певних якостей, стійкого функціонування, деградацію та загибель, ПЗ як система стає більш пристосованим для вирішення завдань [3].

Модельована система управління проектами повинна бути окремим функціональним блоком, що вбудовується в систему Jira і Confluence, а також взаємодіє з ними за допомогою їх функціональних можливостей. Таким чином, існуюча система управління проектами в компанії ТОВ «Майстер Маінд Інк», що поєднує ручне та автоматизоване, дещо змінює свою структуру, організацію та функції.

Розглянемо, як змінюються функції. У моделі As Is функції створення цілого ряду документів і завдань лежали на проектній команді і проектному менеджері. За допомогою R&D-плагіну ця робота виконуватиметься автоматично, оскільки в систему, що розробляється, буде закладено алгоритм R&D-діяльності, завдяки чому шаблони документів на кожному етапі процесу управління формуватимуться автоматично; завдання та доручення також генеруватимуться за допомогою нової системи завдяки функції відстеження дій проектної команди.

Таким чином, не порушуючи встановлених меж зміни функцій існуючої системи управління, система To Be розширює спектр функцій, що використовуються в команді.

Організаційна адаптація системи управління проектами у тому, що колишні потоки управлінської інформації змінюють свою траєкторію. Взаємодія між проектним менеджером та відділами компанії здійснюватиметься у новій моделі через єдиний інформаційний центр управління. При цьому точки входу та виходу організаційних потоків залишаються незмінними.

Також дещо змінюється структура існуючої системи управління проектами. Методи ручного управління перестають бути обов'язковими і в системі Jira з'являється новий елемент - R&D-плагін.

На основі структурно-функціонально-організаційного методу адаптації було розроблено модель АСУ проектами на підприємстві ТОВ «Майстер Маїнд Інк». Модель представлена такими діаграмами:

- верхньорівнева діаграма екосистеми "R&D-плагін";
- діаграма бізнес-процесів управління "To Be";
- діаграма прецедентів;
- ER-діаграма.

Верхньорівнева діаграма екосистеми "R&D-плагін" показана на рисунку 3.1.

Діаграма екосистеми показує, з якими вже існуючими підсистемами належить взаємодіяти новому плагіну, і відображає зв'язок між ними.

Рисунок 3.2 зображує модель взаємодії системи управління проектами To Be (Як буде) з учасниками етапу R&D.

Як ми можемо бачити, у цій моделі з'являється ще один учасник – R&D Plugin. Процеси, які були присутні в моделі As Is, перейшли з пулів учасників проекту в пул плагіна, що означає, що виконання цих процесів бере на себе плагін.

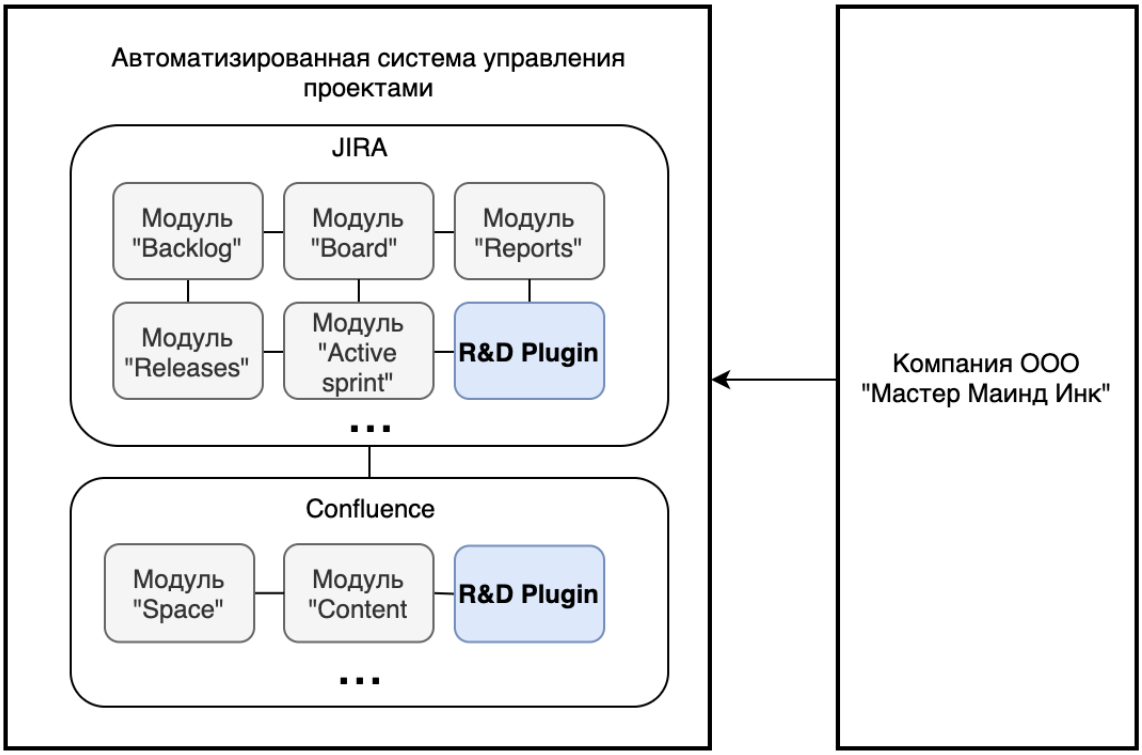

Рисунок 3.1 – Діаграма екосистеми плагіна, що вбудовується, для управління проектами на етапі R&D

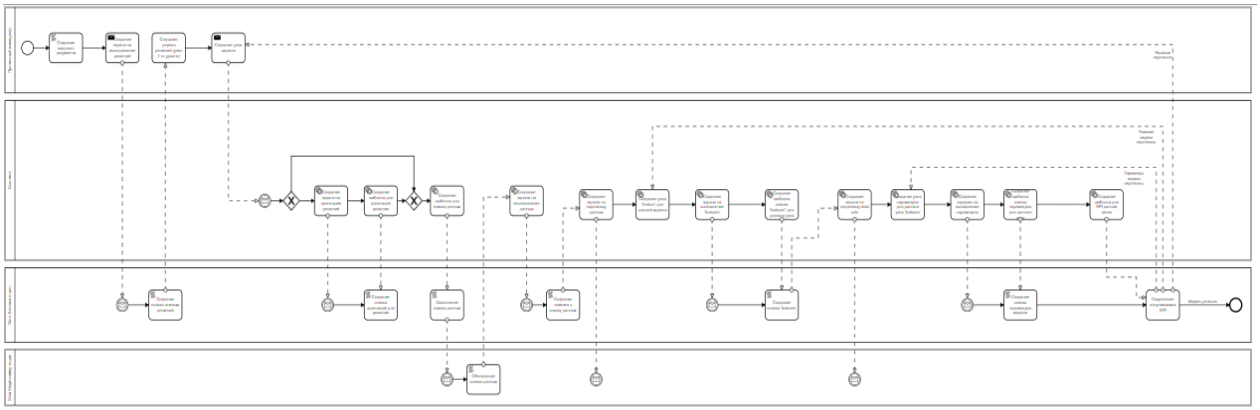

Рисунок 3.2 – Діаграма бізнес-процесів моделі To Be управління проектами на етапі R&D

Забираючи він частина обов'язків інших учасників процесу, новий плагін може істотно скоротити час, необхідне проектної команді створення завдань, документації і відстеження процесу розробки.

Розглянемо детальну пропоновану діаграму бізнес-процесів управління проектами на етапі R&D To Be:

- створення вхідного документа. Для ініціації створення нового проекту проектним менеджером плагін пропонує йому заповнити вхідний документ за встановленим шаблоном: заповнити список цілей проекту,заповнити список бізнес-вимог, до кожної бізнес-вимоги додати коментар - чи створюється проект за допомогою ШІ або алгоритмізованого підходу, а також створити набір KPI майбутнього продукту;

- створення завдання вивчення рішень. Після закінчення наради PM закінчує роботу над створенням документа, перевіряє заповнені дані та підтверджує створення нового проекту у системі Jira. p align="justify"> Плагін створює новий проект в Jira, створює завдання на дослідження рішень, а також зберігає протокол, заповнений PM в системі Confluence. Далі PM закріплює завдання за співробітником Data Science-відділу. Співробітник, який отримав завдання, розпочинає її виконання;

- створення шаблону списку можливих варіантів розв'язання. Плагін автоматично створює шаблон для знайдених варіантів рішення у системі Confluence. Даний шаблон як посилання буде прикріплено до завдання, створеного у п.2;

- створення переліку готових рішень. Після пункту 3 співробітник Data Science заповнює список знайдених рішень документ. Після наради, де проектна команда обговорюватиме варіанти рішення, творець списку зробить позначки, у яких розставить пріоритети кожного знайденого рішення. Також позначаються рішення, які потребують адаптації;

- створення дерева рішень (вузол 1-го рівня). На сторінці плагіна проектним менеджером створюється кореневий вузол майбутнього дерева рішення;

- створення вузла моделі. Проектний менеджер ініціює створення вузла рішення. АСУ створює вузол рішення відповідно до обраних у п.4 пріоритетів і переходить до наступного етапу;

- створіння завдання на адаптацію рішення. Якщо рішення, відповідне створеному в п. 6 вузлу не можна застосувати у готовому вигляді, то плагін створює завдання у Jira на адаптацію рішення. PM може закріпити її за конкретним співробітником Data Science-відділу;

- створення шаблону для адаптації рішення. Якщо було виконано, то плагін створює шаблон у системі Confluence для документа адаптацію рішення і прикріплює його посиланням до завдання, створеної в п.7;

- створення списку адаптацій на вирішення. Співробітник Data Science- відділу, який отримав та виконав завдання, заповнює список адаптацій і за створеним у Confluence шаблоном;

- Створення шаблону для списку даних. Після закінчення п.6, якщо рішення не вимагало адаптацій, або після п.9, якщо вимагало, плагін створює шаблон списку даних, необхідних для поточного вузла;

- створення списку даних. Після вибору варіанта рішення проектна команда збирається на нараду та створюється список даних, необхідних роботи над проектом. Цей пункт також може виконати співробітник Data Science-відділу самостійно. Цей список заповнюється за шаблоном, створеним Confluence;

- оновлення списку даних. Співробітник Data Engineering-відділу роздруковує документ, створений у п.7 або відкриває його в мобільному додатку Confluence та займається збором даних. У документі створюються коментарі про те, які дані вдалося зібрати та в якому обсязі. Після закінчення цього етапу СТО компанії зазначає на інтерфейсі плагіна, що збирання даних закінчено;

- створення завдання дослідження даних. Після виконання п.12 плагін автоматично створює завдання Jira з прикріпленим посиланням на сторінку Confluence зі списком даних і призначає її виконання начальнику Data Science-відділу;

- створення позначок до списку даних. Начальник Data Science-відділу досліджує дані та керує документом зі списком даних, роблячи позначки.

Після закінчення виконання завдання начальник Data Science-відділу зазначає на інтерфейсі плагіна, що завдання закінчено;

- створення завдання підготовку даних. АСУ створює завдання Jira на підготовку даних. PM може призначити виконавця цієї задачі. Співробітник Data Engineering-відділу отримує завдання разом із документом з п.10.;
- створення вузла "features" для даної моделі. Після того, як виконаний п.15, плагін створює вузол features;
- створення завдання складання «features». Після закінчення виконання п.16, плагін створює в Jira завдання складання списку «features».
- створення шаблону списку "features". Як тільки створено завдання, описана в п.17, АСУ створює шаблон списку features для даного вузла. Даний шаблон як посилання на сторінку в системі Confluence прикріплюється до завдання п.17.;
- створення списку "features". Після створення позначок до списку даних співробітники Data Science-відділу проводять нараду з визначення списку «features». Цей список заповнюється в Confluence одним із співробітників Data Science-відділу в заздалегідь створеному шаблоні. Після закінчення цього етапу начальник Data Science-відділу зазначає на інтерфейсі плагіна, що список "features" складений;
- створення завдання підготовку Data Sets. Після виконання п.19 плагін створює завдання у системі Jira на підготовку Data Sets. До завдання прикріплюється документ, створений у п.19. PM може призначити виконавця даної задачі;
- створення вузла параметрів даного вузла «features». Після створення завдання, описаного в п.20, плагін створює новий вузол у системі - вузол параметрів;
- створення завдання складання параметрів. Після створення вузла параметрів, АСУ створює завдання Jira на створення списку параметрів поточного варіанту моделі. PM може призначити виконавця даної задачі;

- створення шаблону списку параметрів для цього сайту. Плагін створює шаблон списку параметрів Confluence і прикріплює посилання на дану Confluence-сторінку до завдання, описаної в п.22.;

- створення списку параметрів моделі. Після виконання завдання, поставленої в п.22, співробітники Data Science-відділу заповнюють список параметрів для поточної моделі шаблону, створеному плагіном;

- проектна команда займається створенням моделі за заданими параметрами та навчання її на підготовлених Data Sets. Після того, як модель вважатиметься навченою, команда має проаналізувати її показники з погляду досягнення KPI продукту;

- створення шаблону для KPI цієї гілки. Після виконання п.24 плагін створює шаблон Confluence для списку показників KPI для цієї гілки;

- збереження отриманих KPI. Після закінчення навчання моделі співробітник Data Science-відділу заповнює створений у п.25 шаблон.

Після закінчення п.26 проектна команда приймає рішення про те, створене рішення є відповідним і можна вважати етап R&D завершеним, або KPI вважаються не досягнутими. У випадку, якщо KPI вважаються не досягнутими, перед проектною командою стоїть вибір про те, до якого з вузлів рішення повертатися:

- створити інші параметри моделі з цим набором «features» і повернутися в п.21;

- створити новий набір «features» та повернутися до п.16;

- повернутися до вибору нового варіанта рішення та повернутися до п.6.

Побудуємо діаграму прецедентів (Use Case діаграма), що відображає варіанти взаємодії учасників R&D етапу з плагіном, що моделюється.

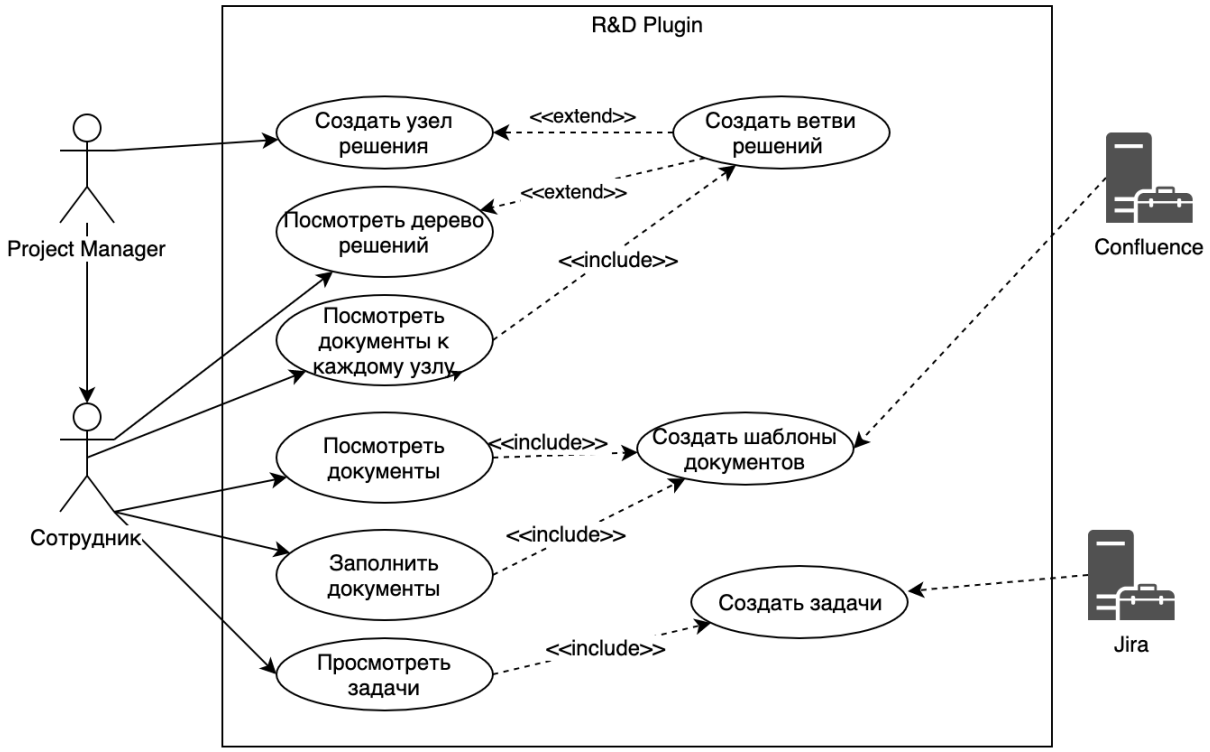

Рисунок 3.3 – Діаграма прецедентів для моделі «To Be» управління проектами на етапі R&D

Діаграма варіантів використання дозволяє описати функціональне призначення системи, а також показує межі системи. R&D Plugin є системою, що автоматизує процеси, які вимагають ручного управління. АСУ проектами може покрити виконання таких функцій як створення документів та його структури, створення завдань. Дані функції є автоматичними і виконуються у потрібний час на основі системи тригерів, яка корелюється із закладеним у систему алгоритмом R&D етапу.

Завдяки функціям відстеження виконання завдань та своєчасного створення шаблонів потрібної документації система спрощує процес створення документації до проекту, а також полегшує створення уніфікованого її формату. Також плагін дозволяє зручним способом візуалізувати фази етапу R&D.

Для проведення структурного аналізу системи, що моделюється, був використаний метод моделювання «сутність-зв'язок», або ER-діаграма. Даний метод дозволяє відобразити об'єкти, які використовуватимуться в системі, а також способи їх взаємодії. Діаграма ER АСУ ШІ-проектами на R&D-етапі наведено рисунку 3.3.

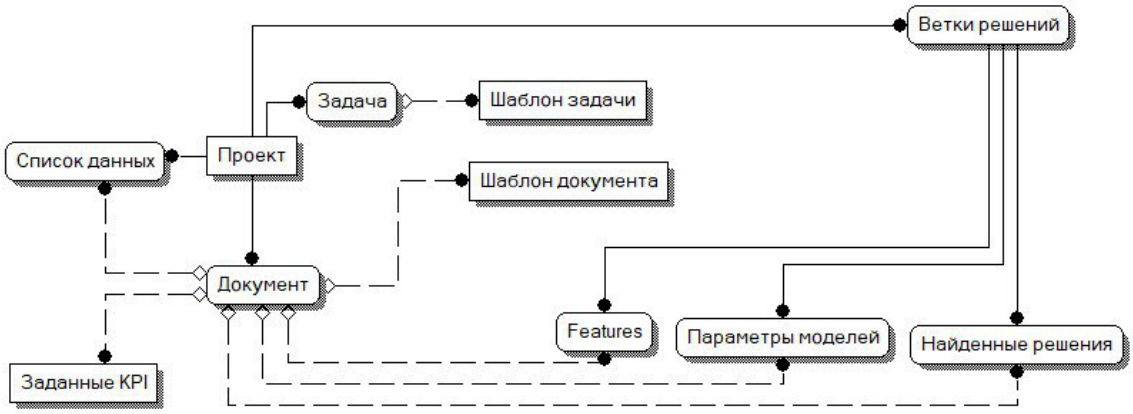

Рисунок 3.4 – ER-діаграма моделі To Be управління проектами на етапі R&D

ER-діаграма відображає взаємодію між ключовими сутностями даної системи (проект, документи, завдання та гілки рішень) у статиці.

Сукупність описаних діаграм (діаграма екосистеми, діаграма бізнес-процесів, діаграма прецедентів та ER-діаграма) дає повний опис моделі To Be інформаційної системи управління ШІ-проектами на етапі R&D на підприємстві ТОВ «Майстер Маінд Інк».

3.2 Якісна оцінка моделі автоматизованої системи управління проектами

З метою якісної оцінки запропонованої моделі автоматизованої системи управління проектами було застосовано такі методи:

- оцінка ступеня автоматизації бізнес-процесів у порівнянні моделей
- експертні оцінки.

Для виявлення слабких та сильних сторін розробленої моделі АСУ проектами, а також потенціалу її подальшого розвитку та визначення можливих ризиків було застосовано SWOT-аналіз.

Оцінка ступеня автоматизації проводилася за методикою, описаною Кораблевим І.Г. [11].

Для проведення оцінки ступеня автоматизації необхідно було підраховувати кількість рутинних або ручних дій, що виконуються в даний час в компанії (модель As Is), і кількість тих же дій після можливого впровадження плагіна (модель To Be) і автоматизації процесу управління.

Найбільше рутинних дій посідає два процесу: створення завдання у системі Jira і створення документа у системі Confluence. Створення завдань є рутинною дією, тому що на етапі R&D завдання в основному є короткостроковими, але їх досить багато, тому проектному менеджеру іноді є недоцільним витрачати час на створення та керування завданням у системі Jira.

Створення документа є ще й трудомістким саме собою, оскільки система Confluence не найзручніший інструмент.

Також незручністю є те, що документи в силу специфіки етапу R&D доводиться створювати технічним фахівцям.

Подібні обов'язки рідко покладаються не на проектних менеджерів, тому технічні фахівці часто замість повноцінного створення документа прикріплюють фотографію того ж документа, написаного від руки, а то й зовсім нехтують виконанням цього завдання.

Оцінка ступеня автоматизації проводилася на підставі порядкової шкали 10 рівнів автоматизації (LOA) Т. Шерідана та В. Вепланка [11] (таблиця3.1).

У таблиці 3.2 перераховані елементарні дії користувача у процесі створення завдання та створення документа As Is та дано оцінку ступеня їх автоматизації.

Таблиця 3.1 – Шкала рівнів автоматизації Шерідану та Вепланка

Рівень автоматизації (LOA)	Опис
1	Комп'ютер не пропонує допомоги: л юдина повинна приймати всерішення та виконувати всі дії сам
2	Комп'ютер пропонує людині повний набір рішень/дій, альтернативи (приклад – робота з електронним довідником)
3	Комп'ютер пропонує повний набір рішень/дій, альтернативи та звужує вибір до кількох варіантів
4	Комп'ютер пропонує одну альтернативу
5	Комп'ютер пропонує одну альтернативу та автоматично виконує це пропозиція, якщо людина погоджується
6	Комп'ютер пропонує одну альтернативу і виконує цю пропозицію, якщо людина протягом обмеженого часу не накладає вето на автоматичне виконання операції
7	Комп'ютер виконує операції автоматично, обов'язково інформуючи людини
8	Комп'ютер виконує операції автоматично інформує людину, тільки якщо комп'ютер запитають
9	Комп'ютер виконує операції автоматично інформує людину, тільки якщо він (комп'ютер) вирішить
10	Комп'ютер вирішує все і діє автономно, не зважаючи на людини

Розглянуті процеси (створення документа і завдання) неодноразово повторюються протягом усього етапу R&D.

Знайдемо загальну кількість операцій, що виконуються учасниками проектної команди на одній повній гілці рішень.

Також розділимо ці операції на 2 категорії – ручні (LOA<6) та автоматичні (LOA>=6) – і знайдемо відсоток змісту кожної з категорій операцій на етапі R&D.

Таблиця 3.2 – Ступінь автоматизації процесів створення завдання та створення документів As Is

Назва процесу	Опис операцій у цьому процесі	LOA
Створення документа	Відкрити Confluence	1
	Відкрити потрібну папку	1
	Створити документ	1
	Створити структуру документа	1
	Заповнити документ	1
	Опублікувати документ	1
Створення завдання	Відкрити Jira	1
	Відкрити Backlog	1
	Створити завдання	1
	Описати завдання	1
	Призначити виконавця	1
	Перенести до поточного спринту	1

Збільшення рівня автоматизації процесів призводить також до зростання деяких якісних показників управління, серед яких можна вказати покращення візуального представлення циклів рішення, потенційне збереження більшої кількості даних про створювані програмні рішення та багато інших.

Практична актуальність та доцільність розробленої моделі автоматизованої системи управління проектами на підприємстві «ММІ» підтверджується відгуками експертів – технічним директором та виконавчим директором компанії ТОВ «Майстер Маінд Інк» (додатки Г та Д).

Для оцінки ефективності розробленої моделі АСУ проектами застосовано SWOT-аналіз. Він дозволив виявити переваги та недоліки системи, на які слід звернути увагу при подальшій розробці програмного продукту. Також було виявлено потенційні напрями розвитку та способи монетизації програмного продукту. Останнім завданням SWOT-аналізу стало виявлення можливих ризиків, пов'язаних з розробкою програмного продукту, що моделюється.

Рисунок 3.5 – SWOT-аналіз АСУ проектами в компанії «Майстер Маїнд Інк»

SWOT-аналіз показав, що запропонована система має значну кількість сильних сторін та переваг перед старою моделлю управління. Модель To Be пропонує такі покращення як: упорядкування документообігу, збільшення ступеня візуального та функціонального комфорту роботи всіх учасників проекту, покращення комунікації між ними тощо.

Слабкі сторони в основному пов'язані з «прив'язкою» до системи Jira та їхньої залежності таким чином від цінової політики компанії Atlassian та від того, наскільки стабільним буде функціонал продуктів Atlassian. Також до

вразливостей запропонованого плагіна можна віднести звуження цільової аудиторії до користувачів продуктів Atlassian, оскільки не всі data science компанії використовують Jira та Confluence.

3.3 Висновки до розділу

Дослідження методів, засобів та існуючих інформаційних систем управління проектами; докладний аналіз діяльності учасників проектної команди на етапі R&D; розроблений алгоритм процесу проектної діяльності на етапі R&D; вивчена модель As Is існуючих процесів управління в компанії ТОВ Майстер Маінд Інк дозволили розробити модель To Be автоматизованої системи управління проектами, яка враховує недоліки управління, наявні компанії, наприклад, неналагоджений документообіг та наявність рутинних дій.

Модель To Be пропонує адаптувати існуючу в компанії систему управління проектами Jira і Confluence, вбудувавши в неї плагін управління етапом R&D. При цьому (у встановлених межах) дещо змінюється структура, функції та організація інформаційних потоків існуючої системи.

Модель «To Be » повністю описується чотирма медіаграмами: діаграма екосистеми вбудовуваного плагіна для управління проектами етапі R&D, діаграма бізнес-процесів моделі To Be управління проектами на етапі R&D, діаграма прецедентів для моделі To Be управління проектами на етапі R&D, ER-діаграма моделі To Be управління проектами на етапі R&D.

Запропонованого опису моделі To Be за допомогою зазначених діаграм достатньо для того, щоб приступити до написання SRS і розробки плагіна.

ВИСНОВКИ

На підставі розглянутих моделей життєвих циклів проекту, моделей та засобів управління проектами можна зробити такі висновки. Модель життєвого циклу проекту CRISP-DM найбільше адекватно відображає процеси розробки нейронної мережі. Для управління процесами розробки програмного продукту з використанням ШІ найбільш оптимальним буде використання комбінації гнучкої моделі та моделі CRISP-DM.

Гнучка модель є найбільш підходящою для IT-компаній, стартапів, проектів в інноваційних сферах, модель CRISP-DM дозволяє керувати діяльністю відділу R&D; каскадна модель зручна у проектах, де ключовим обмежувачем є термін реалізації проекту, а чи не фінанси.

Паралельне використання даних методологій дозволить оптимізувати процеси та синхронізувати принципово різну діяльність команд, які працюють над створенням одного продукту.

Основні наукові результати роботи полягають у наступному.

Відповідно до мети дослідження, було проведено теоретичне обґрунтування моделі автоматизованої системи управління проектами в компанії ТОВ «Майстер Маінд Інк», яка спеціалізується на розробці програмних продуктів, що містять штучний інтелект та використовують машинне навчання. Аналіз першоджерел на тему дослідження – наукових статей, навчальних посібників з теорії управління проектами, матеріалів мережі Інтернет, де міститься велика кількість значущої інформації про практичний досвід використання різних інформаційних систем управління проектами, дозволив теоретично обґрунтувати необхідність та можливість використання існуючих засобів та інструментів для побудови моделі ІС управління ШІ-проектами на підприємстві.

Для досягнення поставленої мети було вивчено сучасні методи, інструменти та інформаційні системи управління проектами, визначено їх переваги

та недоліки; шляхом спостереження за діяльністю проектної команди компанії «Майстер Маінд Інк» зі створення ШІ-продукту було проаналізовано її зміст та виявлено велику кількість рутинних дій, некерованих, дублюючих і тому марних робіт, а також недоліки документообігу. Оскільки управління проектами в компанії ведеться в системі Jira, було ухвалено рішення про розробку моделі АСУ проектами з метою подальшого створення на її основі вбудовуваного плагіна Jira.

Теоретичне обґрунтування та аналіз процесів управління на підприємстві «Майстер Маінд Інк» дозволили розробити модель автоматизованої системи управління проектами, повний опис якої складено з чотирьох діаграм: діаграми екосистеми плагіна, що вбудовується, для управління проектами на етапі R&D, діаграми бізнес-процесів моделі «To Be» управління проектами на етапі R&D, діаграма прецедентів для моделі To Be управління проектами на етапі R&D, ER-діаграма моделі To Be управління проектами на етапі R&D.

Була проведена якісна оцінка розробленої моделі із застосуванням таких методів експертних оцінок та порівняльного аналізу рівня автоматизації бізнес-процесів підприємства при використанні моделей управління «As Is» та «To Be». Було показано, що модель To Be дозволяє збільшити ступінь автоматизації створення завдань на 85,7%, а створення документа на 50%.

SWOT-аналіз моделі «To Be» показав сильні та слабкі сторони нової моделі, а також можливі ризики та потенціал подальшого розвитку.

У цілому, новизна дослідження та його теоретичну значимість полягає в тому, що запропонована модель системи управління проектами спрямована в першу чергу на управління етапом R&D, який найважче піддається управлінню через специфіку науково-дослідної діяльності; для створення адекватної моделі управління розроблено алгоритм процесу проектної діяльності на етапі R&D; запропонований спосіб представлення життєвого циклу проекту та фаз проектної діяльності у вигляді дерева рішень, явною перевагою якого

перед іншими життєвими моделями є візуальне відображення вихідних точок початку наступного циклу.

Рекомендації щодо практичного застосування розробленої моделі та подальший розвиток теми дисертаційної роботи полягають у наступному:

- втілення моделі "To Be" у вигляді плагіна R&D та впровадження його в систему управління проектами компанії ТОВ "Майстер Маінд Інк";

- керування часом. На даний момент запропонована модель ніде не враховує такого важливого чинника, як час. Ніде в моделі немає контроль часу, тобто не встановлений параметр і не розроблена функція, регулююча кількість часу, необхідна для: вивчення предметної галузі проекту,

- пошуку адекватних та доступних для реалізації рішень формулованих задач;

повторення циклу дослідження (якщо не пощастило знайти відразу необхідне рішення).

Також не визначено протяжність часового інтервалу навчання моделі.

Проблема регулювання часу в науково-дослідних проектах, які мають бізнес-мети, є дуже важливою, оскільки збільшення кількості часу на R&D-етапі є цілком природним. Дослідник прагне знайти найкраще рішення, як з погляду ефективності, і з естетичної і пізнавальної.

Складність полягає в тому, що кількість часу в таких дослідженнях дійсно важко регулювати. Планування робіт та прогноз того, скільки потрібно годин, днів або місяців на дослідницьку частину розробки проекту (estimate) залежить від багатьох факторів, як зовнішніх, так і внутрішніх: складності предметної галузі, її рівня вивченості та розробленості в науці; наявності математичних та прикладних процедур, необхідних для створення нейронної мережі; складності реалізації самого проекту; професійного рівня співробітників, і навіть психологічного особистісного профілю кожного; сформованих взаємовідносин всередині компанії та із замовником та ін.

Проте управління часом у компанії виконується – кожним співробітником, проектним менеджером та Product Owner-ом.

Отже, можна спробувати розробити деякі плаваючі часові коефіцієнти, що керують процесом створення проекту. У процесі накопичення досвіду виконання проектів компанією відбуватиметься налаштування тимчасового коефіцієнта та вдосконалюватиметься система управління проектами.

У процесі розширення «бібліотеки готових рішень» за профілем компанії можлива її інтеграція із зовнішніми подібними бібліотеками;

створення рекомендаційної підсистеми, яка спостерігатиме за роботою дослідника на етапі «аналіз рішень». Вивчаючи запити дослідника у конкретній області, така підсистема зможе допомагати співробітнику пошуку необхідного рішення, здійснюючи пошук джерел.

Економічність та результативність запропонованої моделі автоматизованої системи управління проектами забезпечується наступними її властивостями:

- автоматизація рутинної роботи для підприємства;
- автоматизація ручних методів керування;
- збереження всіх документів, що супроводжують процес створення проекту;
- напрями організаційних потоків через єдиний інформаційний центр;
- скорочення часу, необхідного для кожного комунікативного акта;
- незалежність від взаємної територіальної віддаленості працівників на проекті;
- підвищення комфорту та швидкості взаємодії між співробітниками.

Достовірність та обґрунтованість результатів забезпечується обраною теоретичною та методологічною основою дослідження; аргументованістю висновків, підтверджених оцінками експертів – директорів компанії «Майстер Маїнд Інк»; відповідністю методів дослідження мети, предмету та завданням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Філімончук Т.В., Іванісенко І.М., Судаков В.О. Застосування статистичних оцінок в управлінні проектами з розробки програмного забезпечення. Проблеми інформатизації: Тези доповідей деся-тої міжнародної науково-технічної конференції. Т.1: секція 2. Черкаси: ЧДТУ; Баку: ВАЗС АР; Бельсько-Бяла: УТіГН; Харків: НТУ «ХПІ»;. Харків: ХНУРЕ; Харків: ДП «ПД ПКНДІ АП». 2022. с. 26.

2. Parsons D. Agile software development methodology, an ontological analysis [Electronic resource] // Academia. - Electronic data. - 2011. - URL: http://www.academia.edu/1020810/Agile_software_development_methodology_an_ontological_analysis.

3. Арутюнова, Д.В. Инновационный менеджмент: [учебное пособие] – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 152 с.

4. Александрова, Т.В. Повышение эффективности проектного управления в организации на основе гибкой методологии Agile // Экономика и бизнес: теория и практика. 2019. №9.

5. Баканов, А. Б., Дрождин, В. В., Зинченко, Р. Е., Кузнецов, Р. Н. - Методы адаптации и поколения развития программного обеспечения // Известия ПГПУ им. В. Г. Белинского. 2009. № 13 (17). С. 66-69.

6. Rosito M.C. A model to integrate software project management with organizational workflows / M.C.Rosito, R.M.Bastos // 12th International Conference on Intelligent Systems Design and Applications (ISDA). - 2012. - pp.40 - 45.

7. Sheeba T. An Ontology in Project Management Knowledge Domain / T.Sheeba, R.Krishnan, M.J.Bernard // International Journal of Computer Applications. - 2012. - Volume 56. - pp. 1 - 7.

8. Бэгьюли, Ф. Управление проектом: пер. с англ. / Ф. Бэгьюли – М.: Гранд ФАИР-ПРЕСС, 2002. – 202 с.

9. Варенникова, О.В., Бобылева, А.А., Голубев, Д.В. Управление проектами в электроэнергетике/ Colloquium-journal. – 2019. - №13(37). – С.43-56

10. Вертакова, Ю. В. Управленческие решения: разработка и выбор: учеб. пособие / Ю. В. Вертакова, И. А. Козьева, Э. Н. Кузьбожев; под общ. Ред. Проф. Э. Н. Кузьбожева. — М.: КНОРУС, 2005. — 352 с.

11. Голубев, С.А. Управление венчурными проектами.-СПб, СпбГТУ, 2009

12. Заренков, В. А. Управление проектами: учебное пособие. - 2 изд. - СПб: АСВ, 2010. - 312 с.

13. Ивасенко, А. Г. Управление проектами / А. Г. Ивасенко, Я. И. Никонова, М. В. Каркавин – Ростов-на-Дону : Феникс, 2009. – 327 с.

14. Parsons D. An ontology of agile aspect oriented software development // Research Letters in the Information and Mathematical Sciences. - 2011 - Volume 15. - pp. 1-11.

15. Stephen R. Palmer A Practical Guide to Feature Driven Development/ Stephen R. Palmer, John M. Felsing // Pearson Education Limited

16. Porrawatpreyakorn N. A Prototype for Support of the Integrated Software Process Development and Improvement / N. Porrawatpreyakorn, G. Quirchmayr, W. Chutimaskul // Advances in Information Technology. - 2010. - Volume 114. - pp. 94 - 105.