

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ТА ВИКОРИСТАННЯ ПРОЄКТНОГО ПІДХОДУ
ДЛЯ СТВОРЕННЯ КОМП'ЮТЕРНОГО КЛУБУ

(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-2
Книш В. А.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Руденко Д. О.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____
(підпис)

Кобилін О. А.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Книшу Владиславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та використання проєктного підходу для створення комп'ютерного клубу

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 31 травня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд теоретичних основ проєктного підходу та інструментів проєктного управління.

2. Аналіз процесу створення комп'ютерного клубу як проєкту.

3. Розробка проєкту комп'ютерного клубу з використанням Trello.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми, постановка задачі, Trello-дошка.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-16.04.25	
4	Аналіз технічних засобів	17.04.25-19.04.25	
5	Розробка методу	20.04.25-27.04.25	
6	Програмна реалізація	28.04.25-14.05.25	
7	Оформлення пояснювальної записки	15.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Руденко Д. О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 61 с., 2 табл., 20 рис., 1 дод., 30 джерел.

КОМП'ЮТЕРНИЙ КЛУБ, ПРОЄКТ, ПРОЄКТНИЙ ПІДХІД, АВТОМАТИЗАЦІЯ, ПРОЄКТНЕ УПРАВЛІННЯ, PYTHON, ЗАПИТ, TRELLO, API.

Об'єктом роботи є комп'ютерний клуб як проєкт.

Метою роботи є розробка проєктного підходу до створення комп'ютерного клубу, його практичне застосування та автоматизація роботи. При розробці використано методи та принципи управління проєктами. Проведено дослідження ринку та SWOT-аналіз розробленого проєкту. Розроблено скрипти для автоматизації керування на основі взаємодії з прикладним програмним інтерфейсом системи управління проєктами.

У результаті роботи здійснена програмна реалізація процесу автоматизації роботи комп'ютерного клубу.

COMPUTER CLUB, PROJECT, PROJECT APPROACH, AUTOMATIZATION, PROJECT MANAGEMENT, PYTHON, REQUEST, TRELLO, API.

The object of the work is a computer club as a project.

The purpose of the work is the development of a project approach to the creation of a computer club, its practical application, and work automatization. Methods and principles of project management were used in the development. A market study and SWOT analysis of the developed project were conducted. Scripts for management automatization were developed based on interaction with the application programming interface of the project management system.

As a result of the work, a software implementation of the automatization.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Теоретичні основи проєктного підходу та інструментів управління проєктами.....	9
1.1 Поняття проєктного підходу	9
1.2 Принципи управління проєктами	13
1.3 Інструменти та методи управління проєктами.....	15
1.3.1 Gantt	17
1.3.2 Kanban	19
1.3.3 Scrum	20
1.4 Постановка задачі.....	22
2 Аналіз процесу створення комп’ютерного клубу як проєкту	23
2.1 Визначення мети створення комп’ютерного клубу.....	23
2.2 Планування етапів реалізації проєкту.....	26
2.3 Ресурсне забезпечення.....	29
2.4 Вибір системи проєктного управління на основі їх аналізу	32
2.4.1 Asana.....	33
2.4.2 Jira	35
2.4.3 Trello	36
2.4.4 Monday.com.....	37
2.5 Аналіз можливостей обраної системи керування проєктами	38
3 Розробка проєкту комп’ютерного клубу з використання Trello	42
3.1 Побудова Trello–дошки для управління проєктом.....	42

	6
3.2 Програмна автоматизація проєкту	45
3.3 Інструкція користувача.....	51
3.4 Заходи з покращення застосунку	55
Висновки	58
Перелік джерел посилання	59
Додаток А Результати виконання функцій.....	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PMI – Project Management Institute (Інститут управління проєктами, міжнародна некомерційна організація, що займається розробкою стандартів, проведенням дослідів, освітню діяльність та акредитацію у сфері управління проєктами)

UI – User interface (користувацький інтерфейс)

JIT – Just in time (підхід у менеджменті, що полягає у прагненні до вдосконалення систем коштом зменшення в процесі виробництва простоїв, вартості зберігання продукції тощо)

Гц – Герц (одиниця вимірювання частоти коливань в системі SI)

API – application programming interface (прикладний програмний інтерфейс, набір визначень підпрограм, протоколів взаємодії та засобів призначених для полегшення створення програмного забезпечення. API може використовуватись для взаємодії з операційними системами, веб-базованими системами, базами даних, апаратним забезпеченням та програмними бібліотеками)

HTTP – HyperText Transfer Protocol (протокол передачі гіпертекстових документів, протокол передачі даних, що використовується в комп'ютерних мережах)

ВСТУП

В сучасному світі комп'ютерні клуби знову набувають популярності як простір для IT-ентузіастів, геймерів та людей, які цінують атмосферу спільної гри. Створення комп'ютерного клубу – це не просто відкриття фізичного простору з обладнанням, а перш за все повноцінний проєкт, який потребує стратегічного підходу, детального аналізу та планування, а також використання сучасних інструментів управління проєктами. Успішна реалізація такого проєкту вимагає чіткої мети, злагодженої команди, а крім того ефективної організації кожного етапу – від задуму до щоденної експлуатації.

Метою цієї роботи визначено розробку та впровадження проєктного підходу до створення комп'ютерного клубу із застосуванням методів управління проєктами, сучасних цифрових інструментів, зокрема системи Trello, а також частковою автоматизацією процесів. Особливу увагу приділено плануванню ресурсів, етапів реалізації, аналізу ринку, а також створенню технічної та організаційної бази, що забезпечить сталу й ефективну роботу клубу.

Актуальність даної роботи полягає у зростанні популярності комп'ютерних клубів з точки зору форми малого бізнесу, як серед клієнтської бази, так і потенційних бізнесменів та інвесторів. Це створює потребу в систематизації підходу до розробки подібних проєктів.

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТНОГО ПІДХОДУ ТА ІНСТРУМЕНТІВ УПРАВЛІННЯ ПРОЄКТАМИ

1.1 Поняття проєктного підходу

Проєкт – це тимчасова ініціатива, яка розробляється для створення унікального продукту, виду послуг чи результату. У своїй книзі «Управління високотехнологічними програмами та проєктами» Рассел Д. Арчибальд, який вважається одним з класиків проєктного управління, наводить інше, не менш влучне твердження. Він писав: «Проєкт – це комплекс зусиль, що робляться з метою отримання конкретних унікальних результатів у межах відведеного часу та в межах затвердженого бюджету, що виділяється на оплату ресурсів, що використовуються або споживаються у ході проєкту» [1].

На відміну від процесу, проєкт має чітко виражені часові границі. Оскільки він має визначений початок і кінець у часі, він також має визначений обсяг і ресурси [2]. Іншою важливою рисою проєкту є чітко визначена мета. Вона має бути унікальною, не рутинною, а головне – не абстрактною. Досягнення мети проєкту вимагає отримання чітких результатів, що відповідають певним, заздалегідь визначеним вимогам, таким як час, гроші або ресурси, які можуть бути використані під час розробки. Вигода від реалізації проєкту – це сукупна вартість усіх товарів, послуг й інших вигід, які є результатом грошових вкладень у цей проєкт.

Що стосується класифікації – проєкти можна розділити на групи за багатьма ознаками. Наприклад, залежно від сфери діяльності їх розділяють на технічні, організаційні та соціальні проєкти. В залежності від масштабу проєкту, виділяють наступні їх види:

- малі, що мають бюджет до 0,5 млн євро (при їх розробці зазвичай застосовуються спрощені процедури керування та формування команди проєкту);

- середні, що мають бюджет до 0,5 млрд євро;

– мегапроекти, які мають бюджет більше 0,5 млрд євро.

Окрім цього їх поділяють на короткострокові та довгострокові (залежно від часу, необхідного на реалізацію), на національні та міжнародні (за країнами учасниками), тощо. На додаток проекти можуть бути об'єднані в програму, або портфель.

Зазвичай, менеджер проекту чи керуюча організація можуть розділити його на підпроекти, або фази, для забезпечення якіснішого управління. Саме сукупність цих фаз – складає життєвий цикл проекту [3]. Ступінь деталізації й термінологія опису окремих процесів залежать від різних чинників, наприклад, від характеру проекту, його предметної області, поставлених задач, наявних ресурсів, уподобань та поглядів проектного аналітика тощо. Головне в процесі виділення фаз, стадій та етапів проекту – це позначити певні контрольні точки, під час проходження яких використовується додаткова інформація та визначаються або оцінюються можливі напрямки розвитку. В решті решт, затверджений поділ – це взаємодія проекту з середовищем. Багато організацій у своїх проектах використовують певний «універсальний» набір життєвих циклів. Базовий життєвий цикл включає в себе п'ять фаз: ініціювання, планування (або розробка), виконання (впровадження), моніторинг і завершення [4]. На рисунку 1.1 представлена типова схема етапів розробки, яка складається з п'яти процесів. З неї розуміємо, що етапи не обов'язково мають слідувати один за одним, та можуть повторюватися в межах одного проекту.

Етап ініціювання по своїй суті – це розробка плану, що має висвітлювати наступний перелік питань:

- аналіз попиту на ринку;
- огляд існуючих процесів або проектів-конкурентів;
- аналіз бізнес-потреб чи вимог у вимірюваних показниках;
- фінансовий аналіз витрат та доходів;
- розробка статуту проекту як підсумкового документу ініціації.

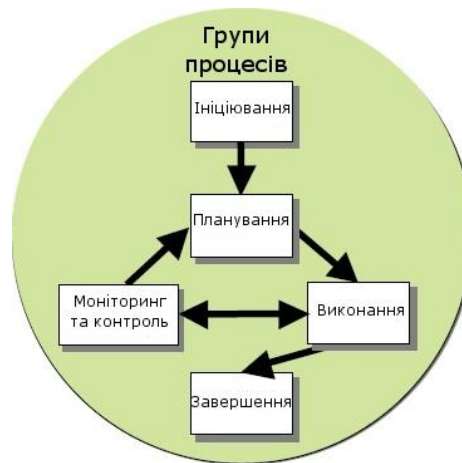


Рисунок 1.1 – Схема процесів розробки проєкту

Наступний після ініціювання етап проєкту – це планування. Головне його завдання – спланувати час, витрати та ресурси з метою коректної оцінки роботи, яку необхідно виконати, та задля ефективного управління ризиками під час впровадження проєкту. Аналогічно до групи процесів ініціювання, недостатньо розроблений план – ставить весь проєкт під удар, та значно знижує його шанси успішно завершити встановлені завдання [5].

Основними процесам на етапі планування проєкту є:

- розробка документу, який визначає межі проєкту та його зміст;
- призначення групи відповідальних за планування осіб;
- постановка бажаних результатів проєкту, та задач, які необхідно розв’язати для їх досягнення;
- оцінка часу та витрат, що підуть на виконання поставлених задач;
- планування ризиків, з якими проєкт може стикнутися на етапі розробки або впровадження;
- визначення вимог до ресурсів, що будуть задіяні для забезпечення реалізації завдань проєкту;
- розробка календарного плану проєкта;
- отримання формального погодження на початок роботи.

Окрім всіх вище перерахованих процесів, на цьому етапі розробки проєкту можуть бути додані й інші, наприклад, визначення ролей та рівня

відповідальності, планування комунікацій, а також проведення зустрічі всіх учасників проєкту [6].

Також для проєктів, що розробляють новий продукт, розробка його концепту може відбуватися паралельно з процесами планування. Причиною цього є висока вірогідність того, що це може допомогти групі з планування у визначенні результатів проєкту та запланованих завдань.

Наступний етап – виконання. Він складається з процесів, які необхідні для завершення робіт, визначених у плані проєкту, з метою досягнення його цілей. Процес виконання включає в себе координування людей та ресурсів, та одночасну інтеграцію та виконання завдань проєкту відповідно до плану управління проєктом [7]. Під час цієї фази розробники мають забезпечити якість кінцевого продукту, та повністю підготувати його до впровадження, або виходу на ринок. Проте слід зазначити, що у багатьох випадках контрольна група може знайти недоліки проєкту та відправити його назад на доопрацювання.

Ще одна фаза життєвого циклу проєкту – це моніторинг та контроль. Головна ідея цієї фази полягає в регулярному огляді та оцінці процесу виконання проєкту з метою визначення відхилень від плану управління проєктом. Вона складається з процесів, які спрямовані на оцінку стану виконання проєкту, щоб потенційні проблеми, які можуть виникнути під час розробки, були визначені, а також для того, щоб були вчасно виконані коригуючі дії для контролю його виконання.

Процес моніторингу та контролю включає:

- визначення поточного статусу виконання завдань проєкту;
- моніторинг змін складових проєкту в порівнянні з планом;
- визначення коригуючих дій, направлених на правильне вирішення питань, що з'явилися, та ризиків, які виникли;
- забезпечення інтеграції лише запланованих змін;

Крім того, часто процеси моніторингу та контролю продовжують свою роботу навіть після впровадження продукту, забезпечуючи зворотний зв'язок

між командою розробників та клієнтами. Такий процес називається підтримкою проєкту. На цьому етапі найбільше уваги слід приділити ефективності та швидкості вирішення проблем, які виникли у користувачів.

Фінальний етап будь-якого проєкту – це завершення. Під час нього виконується формальне прийняття проєкту та його закриття. Саме під час цієї фаз проводяться адміністративні активності, наприклад, передача до архіву робочої документації, підбиття підсумків у формі звіту [8].

Цей етап складається з:

- закриття проєкту, тобто фактичного завершення робіт в усіх групах процесів, включаючи вирішення будь-яких відкритих питань;
- закриття договору, тобто завершення та оплата кожного контракту, та закриття кожного договору, що стосується проєкту в цілому або певного етапу проєкту.

1.2 Принципи управління проєктами

Одна з частини розробки проєкту, якій слід приділити багато уваги – це комплектація персоналу компанії. Як правило команда розробки складається з наступних спеціалістів:

- аналітики, тобто ті, хто відповідають за збір та аналіз даних;
- менеджери (куратори), хто керують процесами та іншими відділами, відповідають за планування;
- розробники, що власне розробляють продукт;
- дизайнери;
- тестувальники, хто тестують продукт на наявність недоліків;
- галузеві спеціалісти, які консультують основний персонал при виникненні питань, для вирішення яких потрібні поглиблені знання в певній галузі;

– власник проекту – людина, що керує фінансами, які виділяються на проєкт. Він же отримує основний прибуток від реалізації проєкту.

Поширеною практикою є призначення завдань ролям, а не конкретним особам (наприклад завдання програмістам, завдання аналітику тощо). Головна перевага такого принципу – чітке розподілення обов'язків та можливість ротування персоналу без втрати продуктивності [9].

Основа ефективного управління проєктами – це принципи, які описують необхідні керівнику та виконавцям навички, а також способи забезпечення високого рівня задоволеності зацікавлених сторін [10]. Всі вони були описані Інститутом управління проєктами (Project Management Institute, PMI) [11].

Перший принцип полягає у чіткості цілі. Поставлені задачі мають бути конкретними, такими, які можна виміряти, такими, що мають значення, а головне – досяжними. Також вони мають бути узгодженими з усіма зацікавленими сторонами проєкту. Причина таких вимог проста – цілі, що відповідають описаним вище критеріям, легко оцінювати на предмет їх досягнення.

Наступним принципом є визначеність результатів. Важливо одразу окреслити те, яким буде фінальний продукт та якомога детальніше описати його для користувачів. Не слід забувати і про якісне планування. Слід одразу визначити обсяг робіт, терміни їх виконання та бюджет, розподілити ролі, обов'язки та ресурси.

Попередня оцінка ризиків також є важливою. Необхідно ще на початковому етапі виявити потенційні проблеми та питання, а також розробити плани їх врегулювання [12]. У цьому можуть допомогти попередні дослідження та досвід експертів у галузі.

Налагодження комунікації – одне з ключових положень, на яке слід звернути увагу, за твердженням PMI. Варто забезпечити постійний обмін інформацією між усіма учасниками. В сучасному світі дистанційна форма роботи є дуже поширеною, тому необхідно упевнитися в доступності даних для тих, хто працює дистанційно та тих, хто працює в офісі.

Ще одним з принципів є підвищення рівня відповідальності. Перед початком безпосередньої роботи над проєктом потрібно пояснити персоналу, як їх задачі пов'язані між собою та що станеться у разі затримки на кожному з етапів.

Наступне положення, що варто розглянути – це моніторинг прогресу. Регулярна перевірка керівником виконання плану й коригування дій у разі виникнення питань – вкрай важливі для досягнення мети проєкту. В цьому процесі вкрай необхідно відразу розробити критерії оцінювання прогресу, наприклад, бюджетні рамки, швидкість виконання роботи, співвідношення загальної кількості завдань до виконаних тощо.

Варто також не забувати про управління якістю. Менеджерам важливо забезпечити відповідність результатів до поставлених вимог та прозорість роботи [13].

На останок – оцінка результатів та висновки. Після офіційного закриття проєкту необхідно підбити підсумки та проаналізувати отриманий досвід. Це знадобиться при розробці нового проєкту.

Потрібно розуміти, що на практиці принципів управління проєктами набагато більше. Проте всі ці та інші принципи – це основа успішного управління проєктами.

1.3 Інструменти та методи управління проєктами

При розробці проєктного підходу важливим завданням є вибір інструментів для керування проєктами, які б мали широкий набір функцій, зручний користувацький інтерфейс (User Interface, UI) та допомагали б дотримуватися усіх принципів управління. Як правило, для таких завдань використовують спеціальні програми, або сервіси. Вони називаються таск-менеджери. Ці програми надають доступ до всіх потрібних функцій в межах одного продукту. Це в свою чергу дозволяє команді, що працює над проєктом,

не витратити зайві час та зусилля на пошуки кількох сервісів та постійні перемикання між ними [14].

Такі застосунки – це оптимальний інструмент, що забезпечує керуючому можливість розглядати проєкт з висоти пташиного польоту, тобто комплексно дивитися на роботу команди, а за необхідності – виокремити та проконтролювати роботу одного співробітника або відділу.

До інших переваг роботи із засобами для управління проєктами можна віднести наступні фактори:

- можливості автоматизації планування та керування;
- покращення взаємодії між учасниками проєкту;
- ефективне розподілення робочого часу;
- вбудована система сповіщень, що має попередити не виконання завдань вчасно і зсуву дедлайну проєкту, як наслідок;
- зрозумілість та наочність відображення зайнятості команди проєкту в цілому, або окремих її учасників;
- спрощення комунікації та підвищення ефективності обміну даними;
- оптимізація роботи з фінансами та спрощення створення звітів.

Для того, щоб обрати систему управління проєктами, перш за все треба врахувати загальну підготовку колективу до роботи з подібними сервісами. В разі потреби слід провести додаткову зустріч для ознайомлення колективу з системами управління в цілому, та конкретними, яким менеджер віддає перевагу у виборі.

Наступний важливий момент – це зрозумілість конкретного сервісу. Слід упевнитися, що обрана система керування доступна, а інструменти, які вона надає – інтуїтивно зрозумілі для використання. Необхідно також звернути увагу на ступінь задокументованості обраної системи та на наявність чітких, але водночас простих інструментів для ознайомлення з нею. Хоча майже завжди розробники таск-менеджерів самостійно розробляють інструкції для користувачів, керівнику слід пропрацювати їх заздалегідь для того, щоб спростити їх або внести зміни за потреби.

Далі слід звернути увагу на наявність та якість технічної підтримки, а саме на її оперативність та об'єм наданої допомоги у разі виникнення труднощів. Службу підтримки можна протестувати завчасно, для того щоб оцінити якість надання послуг.

Окрім всіх вище перерахованих якостей, слід також оцінити й вартість використання систем управління у відношенні з бюджетом проєкту. Використання обраного сервісу повинно окупитися принаймні у середньостроковій перспективі.

Сучасні таск-менеджери в основному використовують один з трьох підходів до управління проєктами: Gantt, Kanban, або Scrum. Для того, щоб обрати сервіс для менеджменту, слід спочатку описати ці методи для кращого їх розуміння.

1.3.1 Gantt

Діаграма Ганта, або стрічкова діаграма – це один із засобів планування та управління проєктами. Дана діаграма застосовується для ілюстрації плану або графіка робіт проєкту. Перший формат діаграми був розроблений Генрі Л. Гантом у 1910 році.

Діаграма Ганта зазвичай представлена у вигляді відрізків, розташованих на горизонтальній шкалі часу, як на рисунку 1.2, де кожен відрізок відповідає окремій задачі або підзадачі [15]. Початкова точка, кінець та довжина відрізка на цій шкалі – відповідають початку, завершенню та тривалості завдання у межах проєкту. На діаграмах Ганта можливо виразити залежність завдань між собою.

У багатьох випадках діаграма Ганта використовується разом із таблицею списку робіт, в якій рядки відповідають окремо взятому завданню, зображеному на діаграмі. В свою чергу стовпці таблиці містять додаткову інформацію про відповідну задачу.

Лінійність та послідовність планування при використанні цього методу робить його ідеальним варіантом для довгострокового планування. В свою чергу часові рамки, що визначені для кожного завдання окремо – одночасно і допомагають, і створюють незручності в розробці.

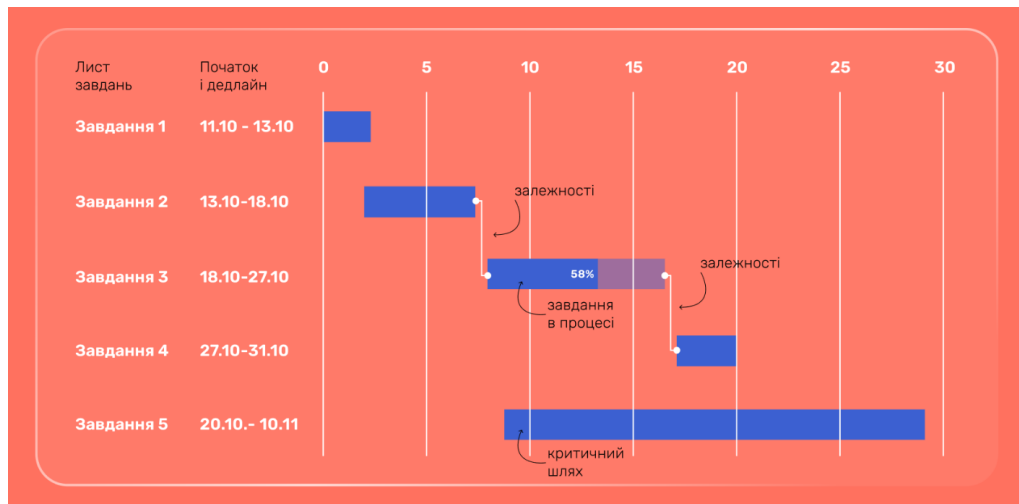


Рисунок 1.2 – Приклад використання діаграми Ганта

Головна проблема цього методу – слабе пристосування до змін, що можуть виникнути у ході виконання проєкту.

Отже основними перевагами використання цього методу є:

- чіткі терміни завдань;
- прозорість у відображенні прогресу;
- зручність довгострокового планування;
- детальний опис взаємозв'язку між задачами.

В той же час недоліки діаграм Ганта не такі чисельні. Проте вони роблять цей метод не оптимальним вибором при розробці проєкту в сучасному світі. Головними проблемами, з якими можна зіштовхнутися є низька гнучкість до змін у системі, та можлива перевантаженість окремих учасників проєкту.

1.3.2 Kanban

Канбан – це метод проєктного управління, який робить акцент на ІТ-підході та уникненні перевантаження членів команди. При використанні цього підходу процеси виконання проєкту, від опису задачі до представлення результатів виконання користувачу, наочно показується учасникам процесу, а учасники команди мають здатність витягувати роботу з черги.

Слово «канбан» походить з японської. Його можна грубо перекласти як «вивіска» чи «білборд». Девід Андерсен описав цей метод, як підхід до інкрементної, еволюційної зміни систем і процесів в організації[16]. За основу він бере обмеження завдання, яке знаходиться в процесі розробки, як ключовий механізм для виявлення проблем в роботі системи. За словами Андерсена цей метод також стимулює роботу в команді для постійного її вдосконалення. Основним інструментом, що використовується цим методом є, як не складно здогадатися, Канбан-дошка. Приклад такої Канбан-дошки представлено на рисунку 1.3.

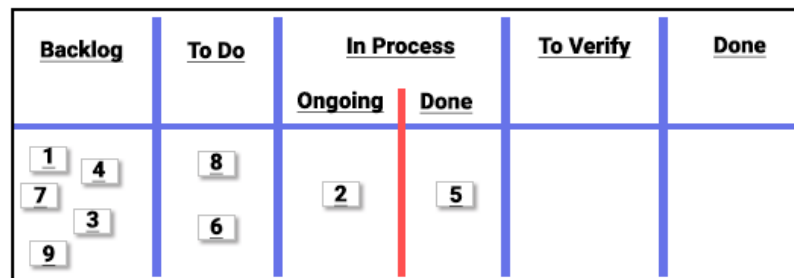


Рисунок 1.3 – Приклад простої Канбан-дошки.

По своїй суті метод «Канбан» ґрунтується на чотирьох принципах.

По-перше, слід починати з того, що є у даний момент. Метод Канбан не описує набір ролей чи кроків процесу. Він починається з ролями й процесами, які уже є в проєкті, та стимулює постійні інкрементні та еволюційні зміни в системі. Якщо коротко, то Канбан – це метод управління змінами у проєкті.

По-друге, потрібно домагатися інкрементних змін. Уся команда, що працює над проектом, має погодитись, що постійні, інкрементні зміни будуть покращувати систему, та забезпечити впровадження цих покращень. Звичайно глобальні зміни можуть здаватися більш ефективними, але вони також мають більший ризик провалу. Канбан працює через постійні, але незначні зміни системи.

По-третє, не варто відхиляти поточні ролі, відповідальності та посади. Зазвичай організація ще на початку роботи має деякі елементи, що працюють задовільно. Саме тому їх варто зберегти. Суть Канбану в тому, щоб поважати поточні ролі та посади для того, щоб отримати глибшу підтримку.

На останок, слід заохочувати розвиток лідерських якостей на всіх рівнях від працівників до менеджерів.

Основними перевагами описаного методу є:

- гнучкість до змін;
- прозорість та простота процесу розробки;
- легкість управління.

Проте Канбан не позбавлений недоліків. Головний з них – це відсутність чітких термінів виконання завдань. Це негативно впливає на процес розробки. Крім того, цьому методу не вистачає опису структури, через що у великих командах розробників можуть виникнути проблеми з організацією.

1.3.3 Scrum

Скрам – це підхід до управління проектами, призначений для гнучкої розробки програмного забезпечення. Найбільше уваги Скрам приділяє контролю якості процесу розробки. Хоча даний підхід і був вперше описаний ще на початку 1986 року, проте вперше застосовувати його безпосередньо в управлінні проектами стали на початку 1990-х років. Цей підхід включає в себе набір методів, церемоній (зустрічей) та попередньо визначених ролей.

Його суть, продемонстрована на рисунку 1.4, полягає у декларуванні завдань у «беклоги», тобто у списки вимог до функціональності, які упорядковані згідно зі ступенем важливості; зустрічах, на яких із «беклогів» продуктів вибирають «беклоги» на спринт; власне спринтів – 15 – 30 денних періодів, під час яких працівники здійснюють функціональний ріст програмного забезпечення; та зустрічі для демонстрації результатів спринту та підбиття його підсумків [17].

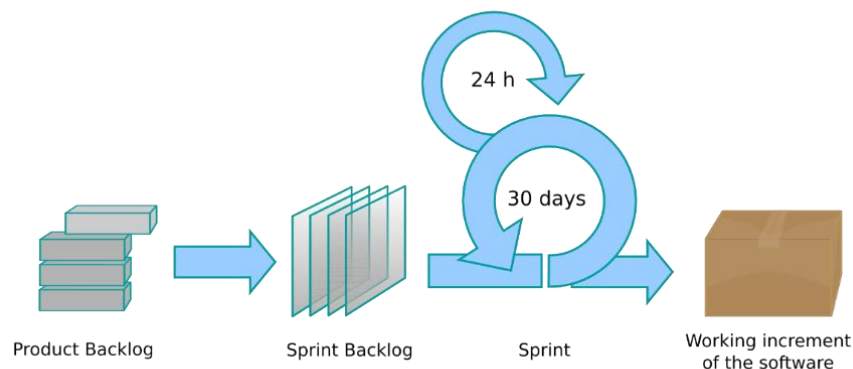


Рисунок 1.4 – Скрам процеси

За методикою Скрам у виробничому процесі є визначені ролі, що розбиті на дві групи – «свиней» та «курей». До «свиней» (тобто повністю задіяних) відносяться:

- «ScrumMaster» – людина, яка опікується процесами, веде їх і працює як керівник проекту;
- «Власник продукту» – людина, яка представляє інтереси усіх зацікавлених сторін, у тому числі кінцевих користувачів;
- «Команда розробки» (у Scrum вона є самокерованою).

До «курей» же, тобто до частково залучених, відносяться користувачі, клієнти, продавці та експерти-консультанти.

Позитивними сторонами використання даного підходу є:

- висока гнучкість до змін;
- постійне вдосконалення процесу розробки та команди.

Проте не слід забувати і про його недоліки. До них входять потреба в хорошій організованості та високій дисципліні команди, а також погана сумісність із проектами, які мають фіксовані вимоги.

1.4 Постановка задачі

Отже, створення проекту комп'ютерного клубу є перспективним та актуальним завданням, для створення успішного бізнесу. При його розробці необхідно врахувати усі бізнес правила та правила управління проектами.

Об'єктом роботи є комп'ютерний клуб як проект.

Метою роботи є розробка проектного підходу до створення комп'ютерного клубу, його практичне застосування та автоматизація роботи.

Для досягнення мети необхідно вирішити такі завдання:

- розробити детальний план створення проекту комп'ютерного клубу з аналізом ринку;
- проаналізувати доступні інструменти керування проектами;
- створити проект в обраній системі управління;
- розробити методи автоматизації отримання інформації про статус роботи комп'ютерного клубу;
- програмно реалізувати розроблені методи.

2 АНАЛІЗ ПРОЦЕСУ СТВОРЕННЯ КОМП'ЮТЕРНОГО КЛУБУ ЯК ПРОЄКТУ

2.1 Визначення мети створення комп'ютерного клубу

Найпершими етапами розробки проєкту є планування та аналіз. Ці етапи визначають його напрямок. Від якості підготовки на них залежить майбутнє усього проєкту.

Визначення мети проєкту – це один з ключових етапів процесу планування. Він передбачає аналіз цільової аудиторії, попиту та ринку обраної сфери. Саме на цьому етапі формуються концептуальні засади проєкту, що в подальшому визначає рішення, що будуть прийматися стосовно вирішення питань та проблем.

Перш за все потрібно проаналізувати ринок. Перші інтернет-кафе з'явилися ще у середині 90-х років минулого століття [18]. Згодом ця сфера занепала через дві причини: підвищення доступності персональних комп'ютерів та початок пандемії COVID-19. Проте в Україні на початку 20-х років ХХІ сторіччя ця ніша почала стрімко розвиватись.

Першим фактором, що призвів до цього стало підвищення цін на комплектуючі персональних комп'ютерів (ПК). В останні роки ціни на найважливіші елементи ПК, такі як відеокарти та процесори, підвищилися у кілька разів. Саме по собі таке явище не перевернуло би ринок, бо ріст цін пропорційний росту продуктивності. Проте у поєднанні із двома наступними чинниками воно дало поштовх розвитку сфери комп'ютерних клубів в Україні.

Наступним чинником став технологічний «стрибок» програмного забезпечення та комп'ютерних ігор. В останні роки темпи підвищення технологічності комп'ютерних ігор та інших застосунків сильно пришвидшилися. Це, в свою чергу, призвело до зменшення часу «старіння» комплектуючих ПК. Ще недавно щойно випущена відеокарта могла залишатися актуальною більше п'яти років. Тепер же нові відеоадаптери

втрачають ефективність вже через кілька років. Тут також слід відмітити технологічний прогрес шкідливого програмного забезпечення та «майнінгу». В сучасному світі існує безліч «комп'ютерних вірусів», які дуже важко виявити. Вони сильно навантажують систему що призводить до її зношення. Теж саме стосується і так званого «майнінгу».

Великий вплив також здійснила фінансова криза, що почалася через глобальну пандемію COVID-19 [19]. Попри все, коли в усьому світі її наслідки почали згасати, в Україні почалася повномасштабна війна, що призвело до ще більшого погіршення фінансового становища в країні.

Останнім фактором, що не слід уникати, є місцеві умови. Окрім фінансового краху, війна в Україні призвела до іншого явища – масового переселення. Велика частка населення країни була вимушена залишити, або втратили свої домівки. Як наслідок різко впав рівень доступності ПК та інтернету. Багато людей тепер не має ні комп'ютера, ні можливості придбати його.

Далі потрібно також проаналізувати конкуренцію на ринку. Хоча досліджувана область і є перспективною, проте в багатьох містах комп'ютерні клуби залишаються не конкуренто- спроможними. На це безумовно впливає важке становище малого бізнесу в Україні. Іншим важливим чинником є не досконале управління. Хоча сфера інтернет-кафе з'явилася вже понад 20 років тому, зараз вона переживає етап метаморфози. Це призводить до ускладнення процесу управління такими проектами і як наслідок – до браку досвіду в керівників.

Ще одна з актуальних тенденцій останніх років – це зростання популярності кіберспорту та соціального геймінгу. Кіберспорт зараз переживає період розквіту. Ріст призових на турнірах призводить до росту зацікавленості у молоді. На підтвердження цього, на рисунку 2.1 [20] представлено графік найбільшого одночасного числа глядачів турнірів з однієї гри. При цьому, як і в звичайному спорті – перегляд турнірів з комп'ютерних ігор надихає простих гравців на покращення власних навичок гри. Проте у

випадку з комп'ютерами це радше означає підвищення мінімальної планки апаратного забезпечення. В свою чергу соціальний геймінг – не менш цікаве явище. По суті воно полягає в переосмисленні цінностей геймінгу. Сучасна молодь все більше втрачає інтерес до комп'ютерних ігор, як до виду розваг та дозвілля. Для них комп'ютерні ігри стають способом соціалізації. Через них вони знаходять нових друзів та спосіб підтримувати комунікацію із тими, хто знаходиться на відстані.



Рисунок 2.1 – Пікова кількість глядачів найбільших турнірів з дисципліни Dota 2 (The International) у 2017–2021 роках

Не в останню чергу треба визначити цільову аудиторію проекту, тобто потенційних користувачів. У випадку з комп'ютерним клубом це не складно. Для такого проекту основною цільовою аудиторією будуть підлітки та юнаки,

тобто хлопці від 12 до 25 років. Проте слід також виділити додаткові категорії потенційних клієнтів, що можуть скласти значну частину від всіх користувачів. В умовах, що склалися на ринку, та які були описані вище, такою цільовою групою стають чоловіки віком від 25 до 40 років, які з певних причин не мають доступу до ПК. Знання потреб кожної з цих груп користувачів має допомогти сформуванню загального змісту проєкту, а також підібрати правильний підхід до маркетингу.

В решті решт, у команди розробників повинно сформуватись бачення проєкту, тобто загальна ідея того, як має виглядати проєкт і яким має бути його наповнення. У випадку комп'ютерного клубу – це бачення структури та послуг проєкту. Зважаючи на аналіз ринку та цільової аудиторії, можна впевнено сказати, що він має бути добре оснащеним технічно та молодіжним. Слід розглядати можливість партнерства з виробниками периферії та залучення клієнтів до участі в турнірах, в цілях розвитку кіберспорту в регіоні.

У підсумку розуміємо, що метою створення проєкту комп'ютерного клубу є не тільки отримання прибутку, а й задоволення потреб цільової аудиторії та локальний розвиток кіберспорту та культури геймерів.

2.2 Планування етапів реалізації проєкту

Для того, щоб перетворити загальне бачення проєкту, на чіткий та детальний план дій, потрібно розбити цей проєкт на етапи. Це дозволить чітко визначити його завдання, їх послідовність, кількість необхідних ресурсів, часові обмеження та ролі, які потрібні для розподілення відповідальності.

На цьому етапі стоять наступні завдання:

– окреслити необхідні для реалізації ролі, та поділити зони відповідальності між ними;

- визначити сильні та слабкі сторони проєкту та після цього описати ризики, що можуть привести до виникнення проблем у ході розробки;
- розподілити процес створення комп'ютерного клубу на етапи.

Перш за все потрібно визначитися із зонами відповідальності та поділити їх. Зважаючи на те, що проєкт малий, визначення його завдань, аналіз ринку, пошук інвесторів, розробку графіку виконання завдань та керування фінансами варто доручити власнику цього проєкту. Це пов'язано із важливістю цих етапів та одночасною простотою виконання, що пов'язана із масштабом. Наступна за важливістю роль – керуючий. Людині, що візьме на себе цю роль, буде доручено керувати та перевіряти процес виконання завдань. Крім того, на керуючого буде покладено відповідальність за комплектування та навчання всього іншого персоналу. Останньою, та не менш важливою, задачею менеджера є звітність про результати проєкту. Це означає, що він є відповідальним за підбиття підсумків окремих етапів, або проєкту в цілому, а також за донесення цих звітів до власника. Наступним буде IT-спеціаліст. На нього буде покладено завдання з розробки проєкту в системі керування та подальша автоматизація. Крім того, зважаючи на малу величину проєкту, на нього можна буде перекласти відповідальність за налаштування обладнання та системне адміністрування. Наступний член команди – SMM-менеджер. В його обов'язки входить все, пов'язане з маркетингом та веденням соціальних мереж, а також допомога керівнику в аналізі тенденцій ринку, конкурентів, цільової аудиторії, тощо. Остання роль у проєкті – це адміністратор. В його обов'язки входить обслуговування клієнтів, підтримка чистоти в приміщенні та будь-яка підтримка користувачів. Таке розподілення обов'язків покликане попередити «зависання» задач, дублювання функцій, а також сприяє прозорості та зрозумілості в управлінні.

Для того, щоб зрозуміти сильні та слабкі сторони проєкту, а також визначити ризики, пов'язані з його реалізацією, варто скористатися методом SWOT-аналізу. Він базується на розділенні чинників і явищ, які впливають на проєкт, на чотири категорії. При використанні цього методу, всі фактори

діляться на внутрішні та зовнішні. А кожна з цих категорій, в свою чергу, ділиться ще на дві – позитивні та негативні. В результаті виходить список сильних сторін проєкту (Strengths), слабких сторін (Weaknesses), можливостей, що з'являються під час реалізації (Opportunities), та загроз, пов'язаних з його здійсненням (Threats) [21].

Сильними сторонами проєкту, що розробляється, мають бути вигідне розташування та якісне устаткування. Зважаючи на можливість проаналізувати ці властивості у конкурентів, можна з легкістю домогтися високої конкурентоспроможності у даних характеристиках проєкту.

Не варто забувати і про слабкі сторони. До них можна віднести невеликий стартовий бюджет, маленьку команду розробників та значну залежність від сезонності відвідувань. Відсутність впізнаваного бренду також буде заважати, проте лише на початку.

До можливостей слід віднести слабкий рівень розвитку конкурентів та стрімкий розвиток ринку. В цьому пункті також відмічається ріст популярності кіберспорту та комп'ютерних ігор в цілому у світі. Гарними можливостями для розвитку можуть стати вірогідні партнерства із місцевими організаціями та виробниками.

Загрозами для проєкту потрібно вважати інфляцію та погіршення фінансового становища малого бізнесу. Моральне старіння техніки також негативно вплине на проєкт. Проте слід зазначити, що цей чинник може почати впливати на проєкт не одразу, а після певного часу.

Для завершення планування залишається лише визначити завдання, що повинні бути виконані для реалізації мети проєкту, та поділити їх на ключові етапи.

Першим та вкрай важливим етапом має бути юридична реєстрація та укладання договорів оренди та співпраці. Цей етап може зайняти найбільше часу через пошук підходящого приміщення та інвесторів або партнерів.

Наступним завданням є ремонт приміщення та закупівля всього необхідного обладнання. На цьому етапі буде використано переважну

більшість усього проєкту. Під час проходження цієї контрольної точки повинні почати з'являтися перші співробітники: при покупці та налаштуванні обладнання не обійтися без системного адміністратора, а закриття вакансії головного менеджера допоможе передати управління від власника проєкту.

Третій етап, що має початися паралельно із виконанням попереднього, пов'язаний з розробкою стратегії маркетингу та безпосередньо проєкту, створення бренду, пошуком підходящого програмного забезпечення, тощо. В основному тут будуть задіяні власник та, меншою мірою, керуючий проєктом.

Далі потрібно почати тестувати системи бронювання, моніторингу, зворотного зв'язку та безпеки. Паралельно менеджер може почати набір основного персоналу.

Останнім етапом буде запуск проєкту, тобто відкриття клубу. На цій стадії головною задачею персоналу буде обслуговування клієнтів та збір відгуків. Задачею менеджера буде моніторинг роботи.

2.3 Ресурсне забезпечення

Опис ресурсного забезпечення – важливий етап планування проєкту, що робить можливим перехід від його ідеї до практичної реалізації. У випадку даної кваліфікаційної роботи варто детально дослідити три основні категорії: людські ресурси, технічне забезпечення та фінансові ресурси.

Будь-який проєкт не буде функціональним за умови відсутності команди. Саме тому варто почати з опису людських ресурсів, необхідних для реалізації. Спеціалізація та кількість персоналу у команді проєкту залежать напряду від його масштабу. Як вже було зазначено вище, для нормального функціонування комп'ютерного клубу знадобляться менеджер, системний адміністратор, SMM-менеджер та кілька звичайних адміністраторів. Для всіх цих ролей, окрім адміністратора, буде достатньо одного працівника, з можливістю розширення до двох для SMM-менеджера та системного

адміністратора. А от для категорії персоналу, що залишилась, потрібно як мінімум два працівники на день. Зважаючи на необхідність у вихідних робимо висновок, що для ефективного функціонування потрібно залучити чотири людини на роль адміністраторів. Окрім штатних працівників слід також розглянути можливість залучення фахівців з певної сфери. У випадку комп'ютерного клубу такими спеціалістами можуть бути фахівці з ведення бухгалтерії, проведення тематичних подій та ремонтних робіт – загалом до десяти людей. Отже загалом необхідні людські ресурси складають від обов'язкових семи до дев'ятнадцяти членів персоналу.

Наступним розглянемо технічне забезпечення комп'ютерного клубу. Для даного проекту це дуже важливий пункт, тому що вся його суть полягає в наданні якісної та безперебійної роботи ігрового середовища. При підборі обладнання варто враховувати що воно буде під щоденним інтенсивним навантаженням. Також уся техніка має бути сертифікованою та з гарантією. До того ж доцільно одразу запланувати цикл оновлення обладнання кожні два чи три роки. Це необхідно для попередження морального старіння техніки та для збереження конкурентоспроможності на ринку.

Ключовими компонентами технічного забезпечення в комп'ютерному клубі є:

- ПК з високою продуктивністю;
- монітори з високою частотою оновлення (165–240 Гц);
- мережеве обладнання, для забезпечення швидкісного інтернет-з'єднання;
- спеціальні периферійні пристрої (клавіатури, комп'ютерні миші, навушники, тощо);
- системи захисту (відеоспостереження та сигналізація);
- додаткове обладнання (система кондиціонування, холодильники, електричний гриль, тощо).

Варто зауважити, що найбільше витратитись слід на ігрові ПК та на додаткове обладнання через їх важливість для створення комфортних умов для клієнтів.

На останок розглянемо фінансові ресурси. Фінансування проєкту повинно забезпечити розв'язання двох основних завдань:

- забезпечити динаміку інвестицій, яка зробила б можливим виконання проєкту відповідно до фінансових та часових обмежень;

- зменшити витрати фінансових ресурсів та ризики проєкту, шляхом оптимізації структури інвестицій.

Для будь-якого проєкту потрібно узгоджувати потоки коштів у часі, тобто в кожний часовий період, наприклад, на місяць, квартал або рік. у розпорядженні підприємства має бути більше коштів, ніж йому потрібно сплачувати. Це необхідно на випадок незапланованих витрат, сплати податків, для повернення інвестицій та сплати відсотків за ними, тощо [22]. Джерелами фінансування проєкту можуть бути власні заощадження, кредит, інвестиції, краудфандинг, державна чи муніципальна програма з підтримки бізнесу, тощо. Визначення формату фінансування є важливим тому, що від нього залежать терміни, потенційні ризики, обсяг відповідальності, та ділові зобов'язання перед інвесторами.

Початок фінансового планування – це формування стартового бюджету. Він повинен враховувати одноразові витрати на стадії розробки та постійні, що будуть присутні навіть після запуску проєкту. Окрім того фінансовий план повинен включати прогноз доходів на перші місяці (від шести до дванадцяти), а також можливі шляхи покриття витрат у разі низького стартового прибутку, або непередбачуваного зростання необхідних коштів. Для мінімізації ризиків варто започаткувати резервний фонд, що буде складати певну частку загального бюджету.

Одноразові витрати включають в себе юридичне оформлення бізнесу, ремонт приміщення, закупівлю обладнання та меблів, оплату послуг фахівців (електриків, майстрів системи вентиляції, тощо), розробку сайту та

маркетингову кампанію. Ця стаття витрат також може включати закупівлю ліцензійного програмного забезпечення, необхідного для функціонування проєкту.

Постійними витратами є оренда приміщення, комунальні послуги, витрати на рекламу, обслуговування техніки, заробітна плата працівників, витратні матеріали (вологі серветки, паперові стаканчики для кави, тощо), а також амортизаційні витрати.

Окремо варто приділити увагу можливим не запланованим витратам. До них головним чином відносяться кошти, необхідні на ремонт обладнання у разі поломки. Сюди ж можна віднести витрати на розхідні матеріали наприклад: фарбу, будівельні матеріали, канцелярські товари, тощо.

У підсумку розуміємо, що ресурсне забезпечення є балансом між потребами та можливостями проєкту. Чітке та якісне планування на цьому етапі мінімізує ризики та дозволить реалізувати проєкт, що матиме якісний старту, а також підтримуватиме високий рівень сервісу, конкурентоспроможності, актуальності та рентабельності у майбутньому.

2.4 Вибір системи проєктного управління на основі їх аналізу

У пункті 1.3 даної кваліфікаційної роботи були проаналізовані методи проєктного управління Gantt, Kanban та Scrum. Результати їх порівняння наведені нижче у таблиці 2.1.

На основі цього аналізу можна зробити висновки, що для реалізації обраного проєкту слід вибирати між системами управління, що використовують саме метод Kanban. Scrum залишається менш підходящим, проте достойним уваги. В свою чергу інструменти, що використовують діаграми Ганта – не бажані у використанні.

Таблиця 2.1 – Порівняння підходів проєктного управління

Характеристика	Діаграми Ганта	Канбан	Scrum
Метод підходу	Лінійне планування	Візуалізація об'єму задач, своєчасність розробки	Ітеративна розробка, контроль якості
Основи методу	Дати, час, залежності завдань	Потік завдань, інкрементні зміни	Ролі, спринти, регулярні вдосконалення
Структура учасників проєкту	Не визначається методом. Не є вагомою	Не визначається методом. Працює з уже визначеними ролями	Розподілення на ролі: «курей» та «свиней»
Інструмент	Діаграми Ганта	Канбан-дошка	Спрощена Канбан-дошка, церемонії
Тип планування	Статичне	Динамічне	Ітераційне
Гнучкість	Низька	Висока	Висока
Умови використання	Чіткі терміни кожного завдання	Постійна підтримка та техпідтримка	Змінні вимоги до продукту

Щоб остаточно визначитися з програмною базою проєкту, проаналізуємо кілька найбільш популярних сервісів для керування проєктами, для визначення найбільш підходящого з них. Цей аналіз допоможе визначити програмне забезпечення, за допомогою якого буде виконана розробка проєкту та подальша його автоматизація.

2.4.1 Asana

Asana – це програмне забезпечення, призначене для керування проєктами доступне у вебзастосунках та на мобільних пристроях. Головною особливістю цього сервісу є обов'язковість використання електронної пошти. Даний сервіс був створений Дастіном Московіцем і Джастіном

Розенштейном у 2008 році, коли вони працювали над покращенням продуктивності роботи серед співробітників у компанії Facebook [23].

У 2012 році компанія Asana розробила власний інтерфейс прикладного програмування (application programming interface, API). Він дозволяє користувачу програмно зчитувати та вносити інформацію, а також автоматизувати операції. Доступність цього API робить інтеграцію Asana з іншими інструментами підвищення продуктивності, такими як Dropbox, Zendesk, Evernote та Google Drive, дуже простою, що безумовно є великим плюсом.

Як можна побачити на рисунку 2.2, Asana використовує метод Kanban у поєднанні з діаграмами Ганта. Таке рішення є незвичайним для засобів управління проєктів. З цього ж рисунку можна відзначити ще одну з переваг даного сервісу – простий у розумінні та використанні UI застосунку.

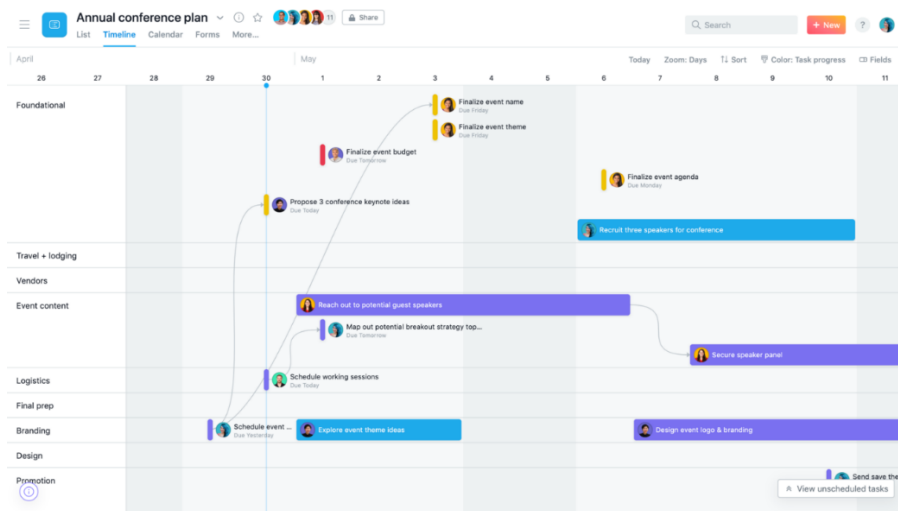


Рисунок 2.2 – Інтерфейс застосунку Asana

Остання ж із важливих переваг Asana – це наявність потужних та гнучких вбудованих функцій та шаблонів.

Слабкими ж сторонами даного продукту є погана підтримка Agile та DevOps. Окрім цього, варто звернути увагу на обмеженість інструментарію Asana за умови використання безкоштовного пакету. Ну і на останок – мала

глибина звітів та аналітики, що є незначним, проте достатньо неприємним недоліком.

2.4.2 Jira

JIRA – це система відстеження помилок, призначена для управління проєктами та для організації технічної підтримки користувачів. Вона була розроблена компанією Atlassian у 2002 році.

Головними елементами JIRA є проблема і робочий процес, де проблемою є опис роботи, яка має бути виконана, а робочий процес – це набір дій, які можуть бути виконані над проблемою для того, щоб її вирішити, або змінити. У проблеми обов’язково має бути назва. Іншими додатковими атрибутами проблеми, в залежності від її виду, можуть бути хто має працювати над нею, її пріоритет, статус та хто створив проблему. Статус відображає, на якому етапі знаходиться робота над проблемою. В JIRA як проблема, так і робочий процес для неї можуть налаштовуватись адміністратором. Виконавець цієї ролі може створювати нові елементи, або модифікувати стандартні. Також адміністратор JIRA відповідає за надання доступу користувачам в залежності від їх ролі в команді.

Однією з великих переваг JIRA є висока функціональність звітів. Система може показувати стан роботи над проєктом. Також вона збирає статистику щодо роботи над проєктом. Це набагато спрощує його етап підбиття підсумків. Іншою важливою перевагою є можливість інтеграції з такими сервісами, як: Bitbucket, Confluence та GitHub.

JIRA більшою мірою спрямована на розробку ПЗ. Для цього в ній поєднані методології Kanban, Agile та Scrum. Через це програма достатньо складна для не підготованих користувачів, а її інтерфейс – досить важкий у розумінні [24]. Це відображено на рисунку 2.3.

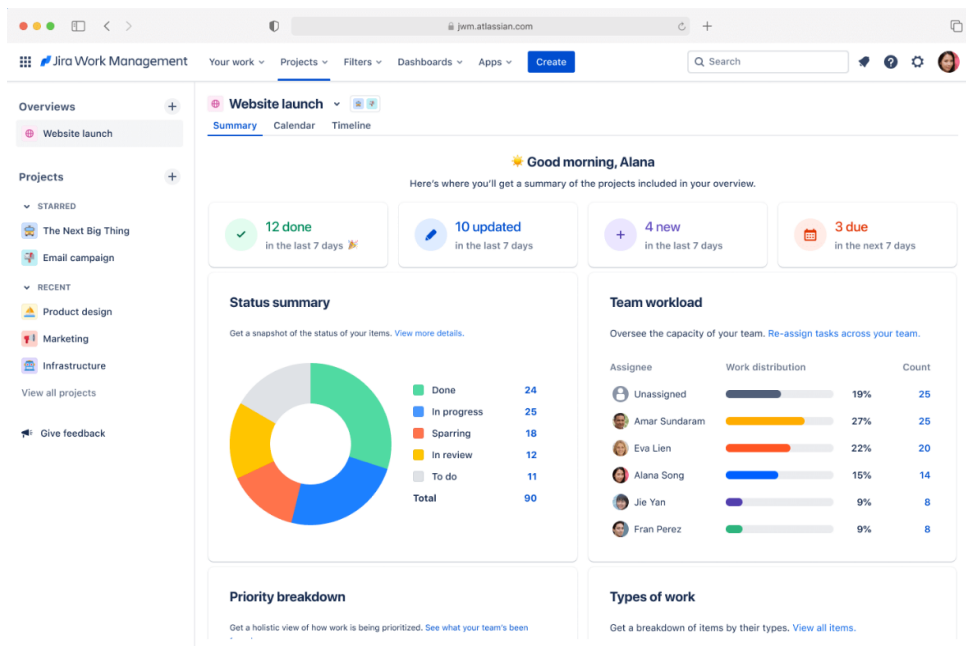


Рисунок 2.3 – Інтерфейс системи JIRA

Зараз JIRA містить три проекти: JIRA Software (для розробників), JIRA Service Desk (для підтримки проекту) та JIRA Core (для управління проектами). Кожен із них можна купити окремо. Це можна віднести як до сильних, так і до слабких сторін. З одного боку ці три сервіси разом утворюють надійну та зручну екосистему. З іншого ж це підвищує витрати на використання, що може бути критичним у рамках невеликого проекту.

2.4.3 Trello

Trello – це безкоштовна система управління проектами від Trello Inc., дочірньої компанії Atlassian. Вона була створена у 2011 році компанією Fog Creek Software, яка тепер називається Glitch. У 2014 році під неї було вирішено створити окрему компанію в Нью-Йорку. В решті решт, Trello Inc. була продана Atlassian в січні 2017 року [25]. Trello використовує метод Kanban, не поєднуючи його з жодним іншим.

Інтерфейс цієї системи, зображений на рисунку 2.4, навіть простіший за Asana. Він працює у форматі drag-and-drop, а всі дані оновлюються динамічно

на фоні. Простота використання та різноманіття вкладених шаблонів роблять Trello одним з найпопулярніших сервісів для керування проєктами. При всьому цьому безплатна версія продукту є дуже об'ємною та не сильно обмежена у функціях та інструментах.

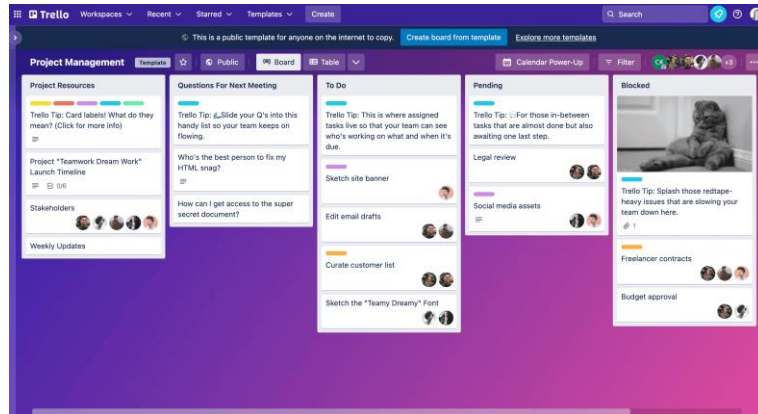


Рисунок 2.4 – Інтерфейс системи Trello

Серед недоліків системи можна зазначити неможливість працювати в режимі офлайн та погана автоматизація. Безплатні плани хоч і мають великий функціонал, проте обмежені у розмірах приєднаних файлів. Крім того, у них відсутня можливість редагувати коментарі. Також Trello, як можна було очікувати від системи, що використовує метод Kanban, погано підходить для великих проєктів.

2.4.4 Monday.com

Monday.com – це вебсервіс для управління проєктами із використанням технології «хмар», що був розроблений у 2014 році компанією Darpulse [26]. Він дозволяє створювати адаптовані робочі простори і, на відміну від багатьох інших, не прив'язаний до жорсткої методології, хоча і використовує елементи методів Kanban та Gantt.

Дана система управління високо цінується за свою гнучкість та адаптивність. Крім цього, вона добре підходить для будь-яких типів команд як

для малих, так і для масштабних проєктів. Ще однією перевагою Monday є зручна, створена розробниками API, що дозволяє налаштувати систему під себе та автоматизувати її.

З недоліків слід виділити слабку адаптованість до Agile-розробки та певну складність у розумінні інтерфейсу, представленого на рисунку 2.5.

Q3 project overview					
Main table Timeline Kanban Dashboard +					
Integrate Automate / 2					
This month					
	Owner	Status	Timeline	Due date	Priority
Finalize kickoff materials		Done	<div style="width: 100%;"></div>	Sep 15	★★★★★
Refine objectives		Working on it	<div style="width: 75%;"></div>	Sep 19	★★★★★
Identify key resources		Stuck	<div style="width: 25%;"></div>	Sep 22	★★★★★
Test plan		Done	<div style="width: 100%;"></div>	Sep 26	★★★★★
Next month					
	Owner	Status	Timeline	Due date	Priority
Update contractor agreement		Done	<div style="width: 100%;"></div>	Oct 10	★★★★★
Conduct a risk assessment		Working on it	<div style="width: 75%;"></div>	Oct 13	★★★★★
Monitor budget		Stuck	<div style="width: 25%;"></div>	Oct 19	★★★★★
Develop communication plan		Done	<div style="width: 100%;"></div>	Oct 22	★★★★★

Рисунок 2.5 – Інтерфейс вебзастосунку Monday.com

Крім того, велика кількість функцій доступна лише у платній версії сервісу. Це робить його менш привабливим для невеликих проєктів через зростання витрат на використання.

2.5 Аналіз можливостей обраної системи керування проєктами

На основі результатів детального аналізу, проведеного у попередньому пункті даної кваліфікаційної роботи, було розроблено порівняльну таблицю, для наочної демонстрації переваг і недоліків обраних систем керування проєктами (табл. 2.2).

Таблиця 2.2 – Порівняння сервісів для керування проєктами

Характеристика	Asana	Jira	Trello	Monday
Використовувана методологія	Kanban, Scrum	Kanban, Scrum, Agile	Kanban	Kanban, Waterfall
UI	Зручний, велика функціональність	Складний, технічний	Інтуїтивно зрозумілий, зручний	Складний, кастомний
Гнучкість	Висока	Дуже висока	Середня	Дуже висока
Автоматизація	Розширена	Багатофункціональна (Jira Automation)	Обмежена (Power-Ups та Butler)	Розширена (шаблони)
Аналітика та звітність	Базова	Детальна	Мінімальна	Дуже детальна
Інтеграції	Slack, Teams, Google Workspace,	Confluence, GitHub, Bitbucket,	Slack, Google Drive, Zapier	Slack, Google, Excel
Тарифний план	Безплатна версія – обмежена	Безплатна версія забезпечує базові інструменти	Безплатна версія має широкий функціонал	Безплатна версія – сильно обмежена

Після розгляду обраних інструментів керування проєктами, було вирішено розробляти проєкт комп'ютерного клубу на базі системи Trello. Головним фактором такого вибору стала безплатна версія, що зберігає достатньо широкий функціонал у порівнянні з платною. До того ж цей інструмент є візуально простим у розумінні.

Перед початком роботи із даним застосунком варто більш детально розглянути можливості, які він надає.

Як вже було зазначено, Trello базується на методі Канбан і використовує канбан-дошки для візуалізації проєкту. В межах дошки створюються списки, наприклад: «План», «Виконується», «Готово» та «На перевірку». У цих списках містяться картки, що відповідають окремим завданням. В свою чергу

картка містить детальну інформацію про це завдання, наприклад, дедлайни, опис, вкладення, мітки для категоризації та призначених учасників.

Попри свою зовнішню простоту, Trello – це досить потужний інструмент для управління проєктами. Серед інших подібних сервісів його виділяють чек-листи, спрямовані на деталізацію завдань, можливість архівації виконаних завдань для збереження історії розробки проєкту та зручна система фільтрації та пошуку по картках за певними тегами, виконавцями або датами. До всього цього додається можливість регулювання дедлайнів, нагадувань та календаря завдань, членів команди, а також можливість коментування завдань чи процесів. Останньою з вагомих можливостей, що надає система Trello, є мобільна версія та можливість синхронізації між різними пристроями.

Важливим аспектом Trello є підтримка так званих «Power-Ups» плагінів, тобто розширювачів звичного функціоналу застосунку [27]. Варто звернути увагу, що в безкоштовній версії доступне невелике число Power-Ups, але у платних планах такого обмеження немає.

Найкориснішими плагінами для управління проєктом є:

- «Calendar» для наглядної демонстрації дедлайнів у вигляді календаря;
- «Card Repeater» для появи можливості використання автоматизації повторення завдань;
- «Custom Fields» для того, щоб додавати свої, нестандартні поля до карток;
- «Google Drive/Dropbox» для використання файлів із сервісів Google.

Іншою важливою перевагою Trello є велика кількість інтеграцій з популярними сервісами. Вони дозволяють синхронізуватися з CRM, бухгалтерією, маркетингом тощо. Це особливо добре впливає на масштабування проєкту. До Trello можна інтегрувати як невеликі програмні застосунки, наприклад, Toggl чи Clockify для відстеження часу, виділеного на завдання, або Slack для налаштування сповіщень у робочому чаті, так і цілі системи чи навіть інші інструменти проєктного управління, такі як Google

Workspace для покращення роботи з документами, Jira для підвищення технологічної оснащеності середовища, тощо.

Окрім цього, Trello має власну вбудовану систему автоматизації, яка називається Butler. Вона дозволяє створювати правила роботи на основі певних тригерів, наприклад, коли наближається дедлайн, створюється картка, або вона має якусь специфічну мітку, тощо.

Прикладом використання системи Butler можуть бути:

- створення повторюваного завдання;
- автоматичне призначення відповідальної особи у випадку додавання якоїсь картки до певного списку;
- перенесення картки до списку «Готово» у разі завершення виконання чек-листа;
- відправлення нагадування у Telegram чи Slack у випадку, коли до дедлайну залишився один день.

Проте як можна побачити, дана система не є досконалою та працює з неважкими, завчасно встановленими завданнями.

3 РОЗРОБКА ПРОЄКТУ КОМП'ЮТЕРНОГО КЛУБУ З ВИКОРИСТАННЯМ TRELLO

3.1 Побудова Trello-дошки для управління проєктом

Практичне здійснення керування проєктом комп'ютерного клубу слід почати з розробки робочого середовища в системі управління. У межах даної кваліфікаційної роботи було розроблено Trello-дошку, що виступає простором для керування усіма аспектами етапів реалізації проєкту.

Першим кроком розробки дошки є розробка списків, що будуть представляти етапи виконання поставлених завдань. Вирішено розробити чотири основні списки: «Ідеї», «Заплановані», «В розробці» та «Виконано».

Список №1 – це «Ідеї». Він призначений для зібрання пропозицій, концептів, потенційних задач та різних ініціатив. Зважаючи на характер списку, можна одразу зазначити специфікацію карток, що будуть в ньому знаходитися. Через те, що вони будуть представляти потенційні елементи, картки не повинні мати ні будь-яких міток, ні термінів реалізації.

Список №2 – це «Заплановані». Він повинен містити картки з попереднього списку, що були переглянуті менеджером та відібрані для реалізації, але ще не почали виконуватись. Тут картки вже повинні мати конкретну специфікацію: опис, мітки, терміни реалізації та відповідальних за виконання осіб.

Список №3 – це «В розробці». Даний список буде представляти завдання, над якими почала виконуватись робота. На цій стадії виконання важливо відобразити стан і прогрес задач, а також завантаженість у реальному часі.

Список №4 – це «Виконано». Він буде зберігати картки вже виконаних завдань. Наявність цього списку дозволяє візуально відслідковувати прогрес виконання задач проєкту.

Результатом створення цих чотирьох списків є робоче середовище проєкту комп'ютерного клубу, зображене на рисунку 3.1.

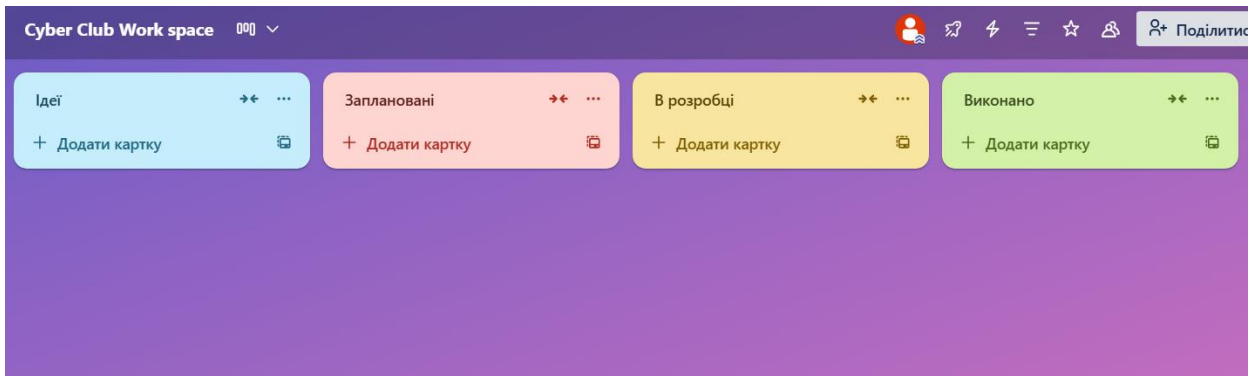


Рисунок 3.1 – Первинне робоче середовище проєкту

Наступним логічним кроком є наповнення цих списків набором карток. До кожної категорії додаємо по одній-дві задачі.

У списку «Ідеї» створимо картки кількох завдань, наприклад «Зробити сторінку в Instagram» та «Пошукати меблі на сайті Jysk». Як вже було зазначено вище – карткам цього списку не обов'язково додавати мітки чи специфікації.

Далі створимо картки завдання для списку «Заплановані». Нехай вони будуть називатися «Зареєструвати бізнес», «Знайти приміщення» та «Закупити обладнання». Починаючи з цього етапу варто розробити систему міток та додати її до створених карток.

Для початку вистачить чотирьох міток. Створюємо їх, та даємо їм назви, наприклад: «грошові операції», «документація», «невідкладність» та «додаткові учасники». Система Trello надає можливість присвоїти кожній мітці свій колір, що буде далі використовуватись для позначення карток. Такий підхід значно покращує візуальне сприйняття та зрозумілість середовища. Після призначення кольорової схеми, можна починати додавати розроблені мітки до карток. Так завдання «Зареєструвати бізнес» отримує всі мітки, окрім «додаткові учасники». Результат усіх вище описаних операцій

представлено на рисунку 3.2. На цьому етапі залишається лише додати необхідні мітки до всіх карток у списку.

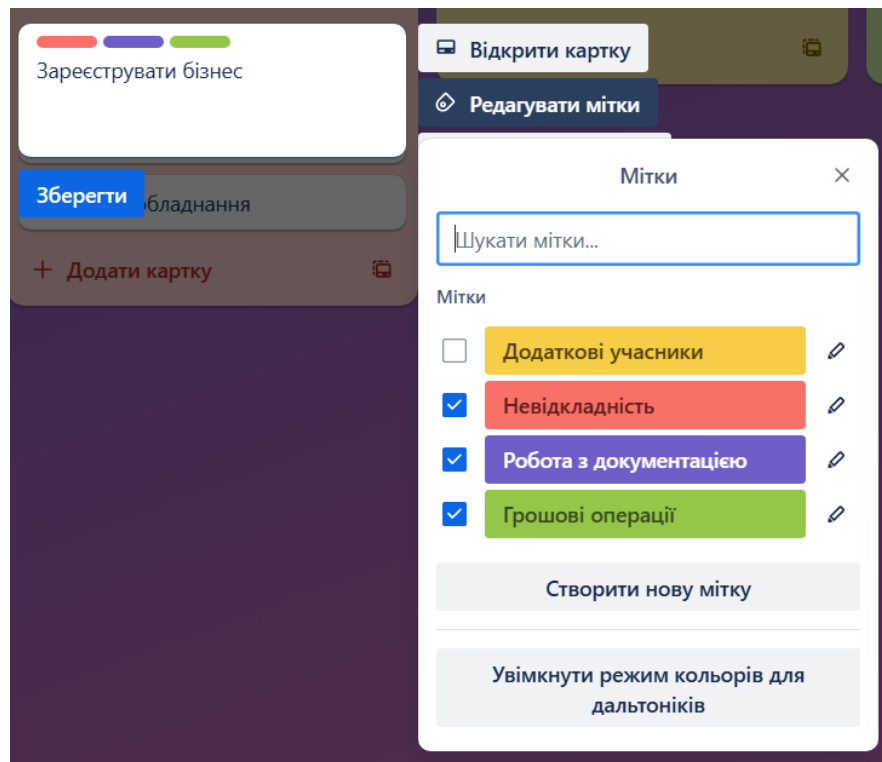


Рисунок 3.2 – Розроблена системи міток

Наступним кроком додамо кожному з завдань дедлайн. Для зразку установимо одній із карток вже минулу дату у якості дедлайну. Також можна долучити кілька інших елементів, наприклад: для картки «Закупити обладнання» додаємо чек-лист з трьох елементів, а для «Знайти приміщення» – посилання на Google Maps із розташуванням бажаного міста для відкриття.

Останнім кроком створимо картки «Створити Trello-дошку» та «Придумати назву» для списку «В розробці». Першу позначимо як виконану, а для другої додамо необхідні мітки та короткий опис, а спеціальні позначки, такі як дедлайн чи чек-листи, не створюватимемо.

В результаті виконання всіх вище-описаних операції було створено та наповнено Trello-дошку, зображену на рисунку 3.3, яка представляє собою робоче середовище проєкту.

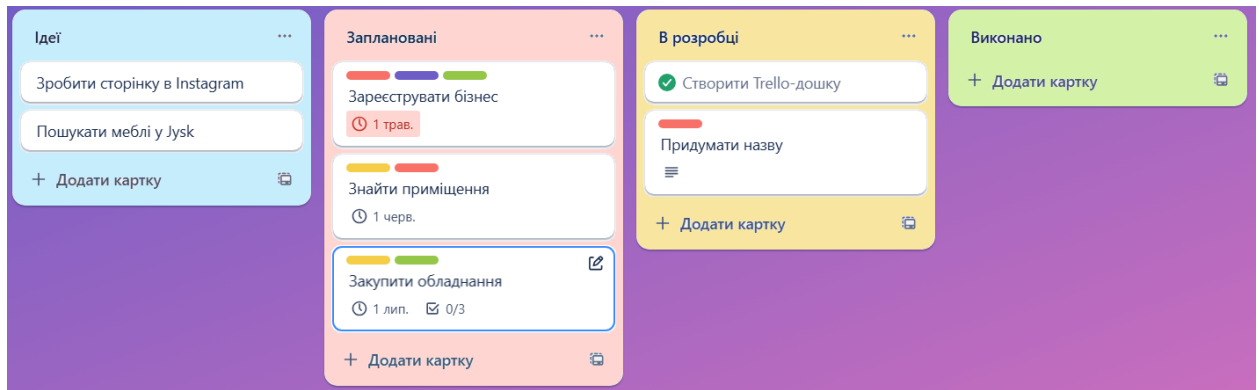


Рисунок 3.3 – Фінальна Trello-дошка проєкту

3.2 Програмна автоматизація проєкту

Як вже не одноразово відмічалось у даній кваліфікаційній роботі – застосунок Trello має дуже обмежені вбудовані інструменти автоматизації. Таким чином, наступним логічним кроком реалізації проєкту є написання програми для автоматизації процесів.

Було вирішено розробляти дану програму на базі мови програмування Python. Взаємодія програми зі створеною дошкою буде відбуватися через вбудоване API. Завдяки його використанню з'являється можливість налагодити взаємодію із зовнішніми сервісами та системами, сприяючи автоматизації, обміну даними та розширенню функціональності програм [28].

Trello API є надзвичайно потужним інструментом. З його допомогою можна створювати повноцінні мобільні, або вебзастосунки для інтеграції з робочим середовищем даної системи керування проєктами [29].

При роботі з Trello API першим важливим кроком є генерація ключа та токена. Створення API Key відбувається на офіційному сайті застосунку у підрозділі «admin», розділу «power-ups». Також потрібно дізнатися ідентифікатор (identifier, ID) створеної дошки та базове посилання на API. Всі отримані дані заносимо у програму у вигляді текстових полів, як показано у лістингу 3.1.

Лістинг 3.1 Дані для доступу до Trello-дошки:

```
API_KEY = '21a7d60d823bc6ffd4f7c7fc21916f34'  
TOKEN = 'ATTA07678c565a9f3d93b957c3e82ee146f2c2b14f13ebdee660ce  
26ffaafcefaad764E75BA4'  
BOARD_ID = '6821848e336e8a2c79594857'  
BASE_URL = 'https://api.trello.com/1'
```

Для роботи із вебзастосунком до Python також треба підключити бібліотеку запитів – requests. Дана бібліотека об'єднує відправлення HTTP/1.1 запитів, що в свою чергу робить можливим створення надійних програм для спілкування через HTTP [30].

У лістингу 3.2 продемонстровано метод, з допомогою якого відбувається занесення інформації з дошки до застосунку. Саме на цьому етапі необхідні створені раніше змінні API_KEY, TOKEN, BOARD_ID та BASE_URL, що містять відповідну інформацію про дошку проекту.

Лістинг 3.2 Запит на отримання інформації з сайту Trello.com:

```
url = f"{BASE_URL}/boards/{BOARD_ID}/lists"  
params = {'key': API_KEY, 'token': TOKEN}  
response = requests.get(url, params=params)  
lists = response.json()
```

Наступними після відображення всієї дошки є такі базові функції, як додавання та видалення списків, карток та міток. Одразу слід зазначити, що картки та мітки можна як додавати, так і видаляти. В свою чергу списки – лише додавати та архівувати. Проте ця формальність майже не впливає на розробку застосунку. Всі ці функції базуються на отриманні інформації з дошки через функцію «get», та її зміни через методи «post», «delete» та «put» (для

архівування списків) з бібліотеки «requests». Отримання нової інформації відбувається через функцію «input», тобто через введення користувачем.

Наступні дві функції відповідають за оновлення та видалення додаткової інформації про картку. Зважаючи на те, що в цих методах основна робота відбувається з додатковими параметрами карток, при відправці запиту на зміну даних – їх варто вказати, причому параметри чек-листів – вносяться окремо, як це продемонстровано у лістингу 3.3.

Лістинг 3.3 запит на оновлення параметрів картки:

```

update_url = f"{BASE_URL}/cards/{card_id}"
update_data = {
    'key': API_KEY,
    'token': TOKEN,
    'desc': desc,
    'due': due if due else None,
    'idLabels': ", ".join(label_ids) if label_ids else None,
    'locationName': location if location else None
}
requests.put(update_url, params=update_data)
if checklist:
    checklist_url = f"{BASE_URL}/cards/{card_id}/checklists"
    checklist_data = {
        'key': API_KEY,
        'token': TOKEN,
        'name': checklist
    }
    requests.post(checklist_url, params=checklist_data)

```

Варто зауважити, що без підключення додаткових бібліотек чи застосунків, функція додавання місцезнаходження оперує лише

координатами, проте не забороняє ручне введення назв (при такому методі у розділі «місцезнаходження» не буде відображатись мапа, а лише текст, введений користувачем).

Наступним кроком потрібно розробити кілька дрібних функцій. Такими є функції переміщення карток та списків через метод «move», а також функції базової звітності, що друкують відсортовану інформацію.

На останок, розробимо функції, що автоматизують процеси всередині дошки. Вони є складнішими у реалізації відносно усіх попередніх. Першою розглянемо функцію автоматичного переміщення прострочених карток. Її суть полягає у пошуку окремого списку для завдань, термін виконання яких сплив, створенні нового списку, якщо пошук був невдалий, перевірці термінів усіх карток, та подальшому перенесенні в окремий список тих з них, які є «простроченими». У лістингу 3.4 відображено елемент коду, що відповідає за перші два етапи цієї функції, тобто за пошук окремого списку із назвою «Провалені завдання» та створенні нового списку із таким ім'ям, якщо його не існувало. Результат виконання даної функції відображено у додатку (рис. А.1).

Лістинг 3.4 Пошук та створення списку «Провалені завдання»:

```

failed_list = next((lst for lst in lists if lst['name'] == "Провалені завдання"),
None)

if not failed_list:
    create_url = f"{BASE_URL}/lists"
    create_resp = requests.post(create_url, params={
        'name': "Провалені завдання",
        'idBoard': BOARD_ID,
        'key': API_KEY,
        'token': TOKEN
    })
    failed_list = create_resp.json()

```

Зазначимо, що такий метод є дуже не практичним з огляду на те, що пошук відбувається за ім'ям списку. Проте такий підхід зумовлений бажанням представити усі можливості програмної реалізації додатку для інтеракції з Trello API. У наступній функції автоматичного переміщення виконаних завдань запропоновано підхід, при якому користувач може обрати список для переміщення усіх карток, що відповідають критерію. Лише в разі, коли поле із номером списку залишиться порожнім, передбачено механізм створення нового із відповідною назвою. У лістингу 3.5 відображено елемент вводу номеру списку та створення нового, за умови що введене поле буде порожнім. Слід звернути увагу, що для новоствореного списку значення параметру «позиція» встановлене на «вгорі». Це означає, що у будь-якому випадку він буде останнім на дошці. Переконалися у цьому можна поглянувши на релюьтат виконання даної функції, відображений на рисунку у додатку (рис. А.2).

Лістинг 3.5 Вибір, або створення нового списку для перенесення карток:

```
target_input = input("Введіть номер списку для перенесення: ").strip()
if target_input == "":
    new_list_name = "На перевірку"
    create_url = f"{BASE_URL}/lists"
    create_params = {
        'key': API_KEY,
        'token': TOKEN,
        'name': new_list_name,
        'idBoard': BOARD_ID,
        'pos': 'bottom'
    }
```

Останньою є функція сортування карток у списку в алфавітному порядку. Вона не потребує жодних дій від користувача, окрім власне її запуску, а в результаті не друкує відсортовані картки, а надсилає запит на

оновлення даних, згідно з результатами сортування. Процес сортування та оновлення даних відображено у наступному лістингу 3.6.

Лістинг 3.6 Сортування карток в алфавітному порядку:

```
sorted_cards = sorted(cards, key=lambda x: x['name'].lower())
print("\nВиконується сортування карток...")
for idx, card in enumerate(sorted_cards):
    update_url = f"{BASE_URL}/cards/{card['id']}"
    update_params = {
        'key': API_KEY,
        'token': TOKEN,
        'pos': idx
    }
    requests.put(update_url, params=update_params)
```

Абсолютно усі функції розподілені по категоріях, наприклад: «дії над картками», «автоматизація» тощо. В свою чергу ці категорії складають «голове меню». У лістингу 3.7 відображено принцип доступу до них. Після оголошення нової змінної «choise», алгоритм зчитує введений користувачем номер пункту меню. Після цього він запускає функцію виклику «меню категорії», що працює схожим чином.

Лістинг 3.7 Елемент вибору пункту «головного меню» для виконання:

```
choice = input("Виберіть пункт: ")
if choice == "1":
    lists_menu()
elif choice == "2":
    cards_menu()
elif choice == "3":
    automation_menu()
```

```
elif choice == "4":  
    report_menu()  
elif choice == "5":  
    other_menu()  
elif choice == "0":  
    print("Вихід із програми.")  
    break  
else:  
    print("Невірний вибір. Спробуйте ще раз.")
```

Таким чином усі елементи програми поєднуються у дуже простий, проте від того легкий у розумінні інтерфейс користувача. Це значно полегшує роботу із розробленою програмою.

3.3 Інструкція користувача

Після запуску файлу застосунку для інтеракції з Trello-дошкою відкривається консоль (рис. 3.4).

Як видно з нього, програма спершу виводить усю необхідну користувачу інформацію про дошку, а саме пронумеровані списки, картки із номерами в цих списках, існуючі мітки з їх додатковим параметром (кольором) та меню доступних інструментів.

```

=====
Вихідна Trello-дошка:
1. Ідеї
  1) Зробити сторінку в Instagram
  2) Пошукати меблі у Jysk
2. Заплановані
  1) Зареєструвати бізнес
  2) Знайти приміщення
  3) Закупити обладнання
3. В розробці
  1) Створити Trello-дошку
  2) Придумати назву
4. Виконано
  (немає карток)
=====
Мітки:
1. Невідкладність (red)  2. Грошові операції (lime)
3. Робота з документацією (purple_dark)  4. Додаткові
   учасники (yellow)
=====

=====Інструментарій=====
1. Списки
2. Картки
3. Автоматизація
4. Звіти
5. Інше
0. Вийти
Виберіть пункт:

```

Рисунок 3.4 – Головне меню застосунку

Подальший вибір підменю інструментів для роботи відбувається шляхом вводу відповідного номеру з меню. Наприклад, після вводу цифри один відбувається «відкриття» (формально друк) підменю для роботи зі списками, продемонстроване на рисунку 3.5.

```

=====Меню списків=====
1. Створити список
2. Архівувати список
3. Перемістити список
0. Назад
Виберіть пункт:

```

Рисунок 3.5 – Підменю інструментів для роботи зі списками

На рисунках 3.6 – 3.9 продемонстровані всі інші існуючі підменю. Кожне з них містить в собі інструменти для керування, поділені на категорії відповідно до назв у меню.

```

=====Меню карток=====
1. Створити картку
2. Видалити картку
3. Додати інформацію до картки
4. Видалити інформацію з картки
5. Позначити картку як виконану
6. Перемістити карту
0. Назад
Виберіть пункт:

```

Рисунок 3.6 – Підменю інструментів для роботи з картками

```

=====Меню автоматизації=====
1. Перемістити завершені картки
2. Перемістити провалені завдання
3. Сортувати картки за алфавітом
0. Назад
Виберіть пункт:

```

Рисунок 3.7 – Підменю інструментів для автоматизації процесів

```

=====Звіти=====
1. Загальний звіт
2. Звіт по завданням
3. Звіт по дедлайнам
0. Назад
Виберіть пункт:

```

Рисунок 3.8 – Підменю інструментів для представлення звітів

Окремо зазначимо, що підменю «Інше» розраховане на функції, які складно віднести до окремої категорії. В даному випадку в ньому знаходяться лише функції для роботи з мітками.

```
=====Інше=====
1. Створити мітку
2. Видалити мітку
0. Назад
Виберіть пункт:
```

Рисунок 3.9 – Підменю інструментів без категорії

Подальший вибір функції для виконання відбувається таким же шляхом, як і вибір підменю, тобто через введення користувачем номеру відповідного інструменту. Після вибору функції запускається скрипт з її виконання. На цьому етапі від користувача вимагається введення даних для занесення їх у поля. Часто їх можна залишити порожніми. Такі поля відмічені відповідним повідомленням. Як приклад на рисунку 3.10 представлено хід виконання функції номер один «Створити картку» з підменю «Картки». Як з нього видно, що після завершення вводу даних користувачем друкується повідомлення про успішність або помилку виконання запиту.

```
Введіть номер списку, до якого додати картку: 1
Введіть назву картки: Тестова_картка 1
Опис картки (необов'язкове поле): Тест тест
Дедлайн (YYYY-MM-DD, необов'язкове поле):
Доступні мітки:
1. Невідкладність (red)
2. Грошові операції (lime)
3. Робота з документацією (purple_dark)
4. Додаткові учасники (yellow)
Введи номери міток(необов'язкове поле): 2,3
Картку 'Тестова_картка 1' успішно створено.
```

Рисунок 3.10 – Результат виконання функції «Створити картку»
із введеними користувачем даними

Деякі функції не потребують внесення даних. Приклад їх виконання представлено на рисунку 3.11.

```
Пошук прострочених карток...
Створено список 'Провалені завдання'
Переміщено: Зареєструвати бізнес
Готово. Переміщено 1 прострочених карток.
```

Рисунок 3.11 – Результат виконання функції
«Автоматизація прострочених карток»

Після завершення виконання функції підменю, до якого вона відносилася, заново друкується. Користувач може продовжити роботу в ньому або вийти, вписавши номер відповідного пункту. У разі виходу з будь якого із підменю відбувається повернення до початкового меню з повторним друком вхідних даних дошки та самого «Головного меню».

У «Головному меню» при виборі інструменту під номером нуль, тобто «Вихід» – програма завершує свою роботу, а консоль зачиняється.

3.4 Заходи з покращення застосунку

Розроблена програма для керування та автоматизації проекту має великий потенціал до вдосконалення. Наприклад, при подальшому вдосконаленні застосунку можна вирішити наступні проблеми:

– розширення інструментів автоматизації у вигляді впровадження додаткових автоматизованих процесів призведе до зменшення обсягів рутинної роботи та підвищення точності обробки інформації, що в свою чергу сприятиме зниженню ймовірності людських помилок та підвищенню ефективності організації роботи. Зокрема, можна реалізувати автоматичне оновлення статусів карток, нагадування про дедлайни та автоматичне формування звітів;

- покращення користувацького інтерфейсу поліпшить досвід від користування та знеможливить некоректну роботу із застосунком;

- оптимізація коду: заходи з оптимізації коду забезпечать підвищення стабільності, ефективності та швидкодію програми. Цей процес включає рефакторинг повторюваних фрагментів, зменшення кількості зайвих запитів до API та покращення структури функцій. Упорядкований та оптимізований код полегшить підтримку, масштабування та впровадження нових функцій, а також сприятиме зниженню ймовірності виникнення помилок. Також можна розглянути підключення нових, більш функціональних Python бібліотек, таких як `ru-trello`;

- налагодження зворотного зв'язку у формі введення інструментів зворотного зв'язку покращить аналіз потреб та вимог користувачів та підвищить оперативність реагування. Варто розглянути можливість додавання коротких оцінок або коментарів до етапів роботи, проведення періодичних опитувань щодо зручності користування інтерфейсом, що призведе до вдосконалення функціональності на основі реальних відгуків;

- розширення можливостей за рахунок інтеграції з картографічними сервісами, засобами зв'язку та календарями дозволить підвищити функціональність платформи. До прикладу, підтримка Google Maps або OpenStreetMap дасть змогу прикріплювати до карток локації, а інтеграція з месенджерами – налагодити систему оперативних сповіщень;

- впровадження інструментів для більш детальних звітів дасть змогу покращити аналіз проєкту, виявити його недоліки та спрогнозувати результати. Наприклад, можна реалізувати функцію перегляду середньої тривалості задач;

- впорядкування інформації в межах системи дозволить уникнути дублювання та спростити пошук елементів. Для цього можна, як приклад, запровадити єдині правила для назв списків, карток, міток і чеклістів. Такий підхід забезпечить послідовність та зрозумілість в роботі з даними;

– оптимізація командної взаємодії у вигляді впровадження чіткої системи розподілу відповідальності та ролей. Вона допоможе підвищити злагодженість командної роботи. Окрім цього можна також запровадити механізми призначення відповідальних осіб за окремі списки чи картки, за відображення прогресу завдань та за призначення дедлайнів для кожного окремого виконавця. Такий підхід дозволить уникати плутанини та посилить контроль над виконанням задач.

Таким чином, розроблений застосунок має велику кількість шляхів покращення, які можна реалізувати в майбутньому під час його вдосконалення. Всі вони допоможуть покращити функціонал програми та підвищити зручність користування нею.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований метод проєктного підходу для створення комп'ютерного клубу.

Основною метою роботи була розробка проєкту «Комп'ютерний клуб», що відповідає усім принципам та вимогам проєктного управління, його практичне застосування та автоматизація роботи.

Під час розробки були використані сучасні технології, бібліотеки та системи керування. Проєкт було розроблено на базі вебзастосунку для управління Trello. Технічна частина застосунку була розроблена у Python, із застосуванням таких бібліотек, як Request та Datetime.

Робочу версію застосунку успішно розгорнуто та протестовано. Результати тестування показали повну функціональність, відповідно до запланованих вимог. Після проведення детального аналізу було виявлено можливості для подальшого вдосконалення застосунку. Вони стосуються покращення технічних аспектів проєкту, а також користувацького досвіду, що дозволить зробити застосунок більш зручним та зрозумілим для користувачів.

Результатом роботи є функціональний інструмент, що використовується для спрощення керування проєктом та його первинної автоматизації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Archibald, R. D. (2003). *Managing high-technology programs and projects*. Wiley.
2. Загальна характеристика систем управління проектами URL: <https://buklib.net/books/28871/> (дата звернення 18.04.2025)
3. Ієвлєва С. М., & Руденко Д. О. (2017). *Основи управління проектами: навч. посібник*. Харків: ХНУРЕ.
4. Управління проектами. URL: https://uk.wikipedia.org/wiki/Управління_проектами#Структура_управління_проектами (дата звернення 19.04.2025)
5. Five Phases of Project Management Life Cycle | Complete Guide URL <https://project-management.com/project-management-phases/> (дата звернення 19.04.2025)
6. Ткаченко, В. П., & Силантьєв, В. С. (2023). Аналіз життєвого циклу розробки проекту WEB системи.
7. 5 фаз життєвого циклу управління проектами у 2023 році URL: <https://www.ranktracker.com/uk/blog/the-5-phases-of-the-project-management-lifecycle-in-2023/> (дата звернення 20.04.2025)
8. Project Management Life Cycle: The 5 Phases Explained URL: <https://www.coursera.org/articles/project-management-lifecycle> (дата звернення 21.04.2025)
9. Храпкін, О., Кіндрат, О., & Чопей, Р. (2023). Управління проектами в ІТ-галузі: методики, інструменти та керування ризиками. *Економіка та суспільство*, (55).
10. Основні принципи управління проектами для початківців. URL: <https://worksection.com/ua/blog/basic-project-management-principles-for-absolute-beginners.html> (дата звернення 25.04.2025)
11. Project Management Institute. (2021, July). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)–Seventh Edition and The Standard for Project Management*. Project Management Institute.

12. Лях, І. М., & Кіш, Ю. В. (2023). Особливості ризик-менеджменту протягом життєвого циклу тестування програмного забезпечення.
13. Kogon, K., Blakemore, S., & Wood, J. (2015). *Project management for the unofficial project manager: A FranklinCovey title*. BenBella Books, Inc..
14. ТОП 7 інструментів для управління проектами у 2025 році URL: <https://worksection.com/ua/blog/5-project-management-tools.html> (дата звернення 25.04.2025)
15. Діаграма Ганта: застосування у менеджменті проєктів URL: <https://shelfy.com.ua/newsroom/diagrama-ganta/> (дата звернення 01.05.2025)
16. Anderson, D. J. (2010). *Kanban: successful evolutionary change for your technology business*. Blue hole press.
17. Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.
18. Інтернет-кафе URL: <https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BA%D0%B0%D1%84%D0%B5> (дата звернення 01.05.2025)
19. COVID-19 pandemic URL: https://en.wikipedia.org/wiki/COVID-19_pandemic#Impact (дата звернення 12.05.2025)
20. Complete Viewership Stats of The International 10 (TI10) URL: <https://escharts.com/news/international-10-complete-statistics26> (дата звернення 04.05.2025)
21. SWOT-аналіз URL: <https://osvita.ua/vnz/add-education/glossary/9635/> (дата звернення 12.05.2025)
22. Микитюк П. П., Брич В. Я., Микитюк Ю. І., & Труш І. М. (2021). *Управління проектами: підручник. Тернопіль: ЗУНУ*.
23. Asana URL: <https://uk.wikipedia.org/wiki/Asana> (дата звернення 13.05.2025)
24. The Pros and Cons of Using Jira for Project Management URL: <https://project-management.com/the-pros-and-cons-of-using-jira-software/> (дата звернення 13.05.2025)

25. Trello URL: <https://uk.wikipedia.org/wiki/Trello> (дата звернення 16.05.2025)
26. Our story – Monday.com URL: <https://monday.com/p/about/> (дата звернення 13.05.2025)
27. What are Power-Ups? URL: <https://support.atlassian.com/trello/docs/what-are-power-ups/> (дата звернення 15.05.2025)
28. Руденко Д. О., Маренич В. В. (2024). Дослідження методів розробки програмних додатків з інтегрованими API. // Scientific research: modern challenges and future prospects. Proceedings of the 2nd International scientific and practical conference. MDPC Publishing. Munich, Germany. 2024. Pp. 144-147.
29. API Introduction URL: <https://developer.atlassian.com/cloud/trello/guides/rest-api/api-introduction> (дата звернення 16.05.2025).
30. Requests URL: <https://pypi.org/project/requests> (дата звернення 16.05.2025)