

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

РАДЗИХОВСЬКА АНАСТАСІЯ ОЛЕГІВНА

Допускається до захисту:
завідувач кафедри
інформаційних технологій,
к. т. н., доцент
Зелінська Оксана Владиславівна
«_____» _____ 2024р.

**РОЛІ ПРОЄКТНОГО МЕНЕДЖЕРА ТА БІЗНЕС АНАЛІТИКА ПІД ЧАС
КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА
ПРИКЛАДІ ПРОЕКТУ ОБРАННЯ НАУКОВОГО КЕРІВНИКА**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Науковий керівник:
Антонов Ю.С., доцент кафедри
інформаційних технологій,
к. фіз.-мат. наук, доцент

Оцінка: ____ / ____ / ____
(бали за шкалою ЄКТС/за національною шкалою)
Голова ЕК: _____
(підпис)

Анотація

Радзіховська А. О. Ролі проєктного менеджера та бізнес аналітика під час командної розробки програмного забезпечення на прикладі проєкту обрання наукового керівника. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (бакалаврській) роботі досліджено взаємодію користувача з існуючою системою обрання наукового керівника. На основі чого був розроблено подальні практичні кроки щодо реалізації додатку для покращення процесу. Розроблений SWOT аналіз майбутнього додатку, user story, реєстри ризиків та інше.

Ключові слова: проєктний менеджмент, бізнес аналітик, розробка продукту, ризик менеджмент.

84 сторінки , 15 рис., 2 табл., 64 джерела.

Abstract

Radzikhovska A. O. The roles of the project manager and business analyst during the team development of software on the example of the scientific director election system project. Specialty 122 'Computer Science', educational program 'Computer Science'. Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

In the qualification (bachelor's) research work, I have investigated the user's interaction with existing systems for selecting a supervisor. Based on this, we developed further practical steps to implement the application to improve the process. The SWOT analysis of the future application, user's story, risk register, etc. was developed.

Keywords: project management, business analyst, product development, risk management.

84 pages, 15 figures, 2 tables, 64 sources.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ I.....	5
ОСНОВНІ ТЕОРЕТИКО-МЕТОДИЧНІ ПОЛОЖЕННЯ УПРАВЛІННЯ ПРОЕКТАМИ В ІТ-СФЕРІ.....	5
1.1. Огляд та аналіз поняття "проект" як об'єкта управління.....	5
1.2 Життєвий цикл розробки програмного забезпечення.....	8
1.3. Особливості роботи SRUM команди.....	13
1.4. ІТ інструменти, що використовуються у роботі бізнес аналітика.....	18
1.5. ІТ інструменти, що використовуються у роботі проджект менеджера...	20
РОЗДІЛ II.....	22
РОЛЬ БІЗНЕС АНАЛІТИКА ПІД ЧАС КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	22
2.1 Основні завдання та функції бізнес-аналітика в проєкті.....	22
2.2 Ролі та відповідальності бізнес-аналітика в команді розробки, та його вплив на якість та успіх проєкту.....	23
2.3 Приклади задач, що вирішувались бізнес аналітиком під час роботи над проєктом обрання наукового керівника.....	24
2.3.1 Epics and User stories.....	24
2.3.2 Взаємодія з замовником.....	27
2.3.3 Взаємодія зі стейкхолдерами.....	33
РОЗДІЛ III.....	39
РОЛЬ ПРОЄКТНОГО МЕНЕДЖЕРА ПІД ЧАС КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	39
3.1 Основні завдання та функції проєктного менеджера в проєкті.....	39
3.2 Управління ресурсами.....	43
3.2.1 Фінансові ресурси.....	44
3.2.2 Людські ресурси.....	44
3.3 Управління проєктом.....	45
3.4 Управління командою.....	49
3.5 Управління ризиками.....	50
3.5.1 Класифікація ризиків.....	51
3.5.2 Приклади ризиків, що виникали під час роботи над проєктом обрання наукового керівника.....	53
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ.....	60

ВСТУП

Актуальність роботи:

Актуальність роботи базується на зростаючій потребі в кваліфікованих спеціалістів, які вміють ефективно керувати проєктами та мати глибоке розуміння бізнес-процесів. У сучасному світі динамічного розвитку та конкуренції вміння швидко реагувати на зміни та ефективно керувати ресурсами стають критичними для успіху. Підприємства активно шукають фахівців, які здатні не лише виконувати визначені завдання, а й вміють аналізувати бізнес-дані та прогнозувати тенденції з метою прийняття обґрунтованих стратегічних рішень.

Проєкт по розробці рішення, яке автоматизує процес обрання наукового керівника, є гарним майданчиком для прояву компетентностей в галузі проєктного менеджменту та бізнес-аналітики. У рамках цього проєкту студент має можливість не лише застосовувати свої знання та навички на практиці, але й співпрацювати зі студентами та науковими керівниками, задля досягнення спільної мети - покращення навчального процесу.

Мета дослідження: аналіз технічних вимог до процесу вибору наукового керівника для студентів, розробка юзер-сторі та проведення оцінки ризиків, які можуть вплинути на успішність проєкту. Цей підхід передбачає врахування потенційних негативних сценаріїв і розробку стратегій їх уникнення або зменшення впливу. В результаті виконання дослідження будуть вдосконалені та відточені навички проєктного менеджера та бізнес-аналітика, що має сприяти підвищенню ефективності й результативності управління проєктами в ІТ сфері.

Завдання дослідження:

- визначення кола стейкхолдерів для даної задачі;
- визначення потреб стейкхолдерів;
- дослідження бізнес процесів, що супроводжують процес створення (розробки) підсистеми обрання наукового керівника;
- розробка деталізованого технічного завдання для підсистеми (Еріс, user story, тощо);

- обрання технологій для реалізації проєкту;
- контроль реалізації проєкту та забезпечення його успішного завершення;
- оцінка внутрішніх та зовнішніх факторів, що впливають на проєкт.
- оцінка ризиків на проєкті.

Об'єкт дослідження: процес впровадження автоматизованої системи обрання наукового керівника в навчальний процес.

Предмет дослідження: ефективність та перспективи впровадження автоматизованої системи обрання наукового керівника в навчальному процесі для спрощення бізнес-процесів.

Апробація результатів дослідження: робота пройшла апробацію на наступних конференціях:

- III Всеукраїнській науково-практичній конференції «Комп'ютерні технології обробки даних» (КТОД 2022)
- II Міжнародної науково-практичної конференції «Прикладні аспекти сучасних міждисциплінарних досліджень»
- V Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих вчених «Прикладні інформаційні технології».

Структура кваліфікаційної роботи: кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та двох додатків. Робота містить 84 сторінки.

РОЗДІЛ I

ОСНОВНІ ТЕОРЕТИКО-МЕТОДИЧНІ ПОЛОЖЕННЯ УПРАВЛІННЯ ПРОЕКТАМИ В ІТ-СФЕРІ

1.1. Огляд та аналіз поняття "проект" як об'єкта управління

У сучасному світі, де є зростаюча з кожним днем конкуренція та технологічні зміни є неухильними факторами, ефективне управління стає критичним для досягнення поставлених цілей в будь-якій галузі. І це розуміє кожна успішна компанія. Якщо замислитись, то кожного дня нас оточують певні цифрові проекти. Наприклад, вранці викликати таксі чи поповнити рахунок через додаток, замовити доставку їжі чи оплатити комунальні послуги. Щодня ми використовуємо десятки різноманітних додатків, які колись були лише проектами [1]. Отже, для початку потрібно розібратись з терміном "проект". Відповідно до настільної книги менеджерів проектів в усьому світі "A Guide to the Project Management Body of Knowledge" або "PMBOK Guide" пояснює слово "проект" як тимчасову діяльність, спрямовану на створення унікального продукту, послуги або результату. Проекти можуть бути як самостійними, так і частиною програми або портфелю [2].

Проекти відрізняються від звичайних видів діяльності тим, що мають такі особливості [3]:

Унікальність – проект спрямований на створення нового продукту, послуги або результату, який не існував раніше.

Тимчасовість – проект має певний період виконання, який обмежений у часі.

Ціль – у проекту є чітко визначена мета, яку необхідно досягти.

Бюджет – проект має певний бюджет, який обмежений у коштах.

Команда – проект реалізується командою, яка складається з різних фахівців.

Проекти можна класифікувати за різними критеріями. Найбільш поширеними є такі:

Сфера проекту:

- Бізнес-проекти спрямовані на покращення ефективності бізнес-процесів, впровадження нових технологій або створення нових продуктів і послуг.
- Інфраструктурні проекти спрямовані на створення або модернізацію ІТ-інфраструктури організації, наприклад, мереж, серверів, систем зберігання даних або програмного забезпечення.
- Дослідницькі та розробницькі проекти спрямовані на розробку нових ІТ-технологій або продуктів.
- Освітні проекти спрямовані на навчання користувачів ІТ-технологій.

Тривалість проекту [4]:

- Короткострокові – мають тривалість до 1 року.
- Середньострокові – мають тривалість від 1 до 3 років.
- Довгострокові - мають тривалість понад 3 роки.

Склад і структура проекту:

- Проекти розробки програмного забезпечення спрямовані на створення нового програмного забезпечення або модернізацію існуючого.
- Проекти інтеграції систем спрямовані на об'єднання двох або більше ІТ-систем в єдину систему.
- Проекти покращення бізнес-процесів спрямовані на оптимізацію бізнес-процесів за допомогою ІТ-технологій.
- Проекти управління проектами спрямовані на впровадження методів і інструментів управління проектами в організації.

Обсяг проекту:

- Малі (до 10 осіб).
- Середні (від 10 до 100 осіб).
- Великі (понад 100 осіб).

Фокус проекту:

- Технічні проекти спрямовані на реалізацію технічних завдань, таких як розробка програмного забезпечення, інтеграція систем або створення IT-інфраструктури.
- Бізнес-проекти спрямовані на досягнення бізнес-цілей, таких як підвищення ефективності бізнес-процесів або створення нових продуктів і послуг.

Характер проекту:

- Одноразові проекти виконуються один раз і не передбачають повторення.
- Повторювані проекти виконуються неодноразово, наприклад, щорічні оновлення програмного забезпечення.
- Неперервні проекти виконуються постійно, наприклад, підтримка та обслуговування IT-інфраструктури.

Складність проекту:

- Нескладні – мають добре визначені цілі, завдання і ресурси.
- Складні – мають невизначені цілі, завдання або ресурси.
- Дуже складні – мають високу невизначеність і ризик.

За ступенем новизни:

- Проекти розробки нового продукту або послуги – спрямовані на створення унікального продукту або послуги, який не існував раніше.
- Проекти вдосконалення існуючого продукту або послуги – спрямовані на вдосконалення існуючого продукту або послуги.

Отже, поняття "проект" не лише визначає тимчасове підприємство, але і є стратегічним інструментом для досягнення цілей та розвитку в умовах сучасної мінливості. Важливо враховувати його основні характеристики та використовувати їх у контексті ефективного управління, щоб забезпечити успіх та стійкість в конкурентному середовищі.

1.2 Життєвий цикл розробки програмного забезпечення

Життєвий цикл розробки програмного забезпечення (SDLC – Software Development Life Cycle) – це послідовність кроків розробки, необхідних для створення проекту, від його первинної ідеї до виходу продукту на ринок і подальшої його підтримки [5]. Він забезпечує структуровану основу для управління всім процесом розробки програмного забезпечення, від початкового збору вимог до підтримки після розгортання. Метою SDLC є забезпечення завершення програмних проєктів вчасно, в рамках бюджету та відповідно до визначених стандартів якості, тобто мінімізувати всі проєктні ризики.

Розробка програмного забезпечення може бути складним завданням у зв'язку з мінливими вимогами, оновленням технологій і міжфункціональними командами. Методологія життєвого циклу розроблення програмного забезпечення (SDLC) забезпечує систематичну структуру управління з конкретними результатами на кожному етапі процесу розроблення ПЗ. У результаті всі зацікавлені сторони заздалегідь узгоджують цілі та вимоги до розробки програмного забезпечення, а також мають план досягнення цих цілей [6].

Ключовими перевагами SDLC є [7]:

- Підвищення зрозумілості процесу розробки для всіх зацікавлених сторін;
- Ефективне оцінювання, планування та складання розкладів;
- Покращене керування ризиками та оцінка витрат,
- Підвищена ефективність: спрощує процес розробки, розділяючи його на керовані фази. Це дозволяє розробникам одночасно працювати на різних етапах, що призводить до більш швидких циклів розробки та поліпшення управління часом.

Етапи життєвого циклу розробки програмного забезпечення (SDLC):



Рис. 1 - Етапи SDLC [8]

Перша стадія розробки програмного забезпечення – це **планування**, яке включає визначення цілей та обсягу програми, аналогічно до вибору найкращого маршруту для подорожі. На цьому етапі чітко формуються завдання та розробляється стратегія для ефективної реалізації проєкту.

Команда разом генерує гіпотези та ідеї, аби зрозуміти ключові потреби користувачів та цілі, які має допомогти вирішити програмне забезпечення. Іншими словами, шукають відповіді на ключові запитання: "Яку проблему вирішує це програмне забезпечення?" і "Яку вигоду це приносить користувачеві?".

На етапі планування також виконується техніко-економічний аналіз. Фахівці команди оцінюють технічні та фінансові аспекти, які можуть мати вплив на успішність програмного забезпечення.

Таким чином, на цьому етапі генеруються основні документи, такі як план проєкту та специфікація вимог до програмного забезпечення, які детально описують функціональність, ресурси, ризики та графік проєкту.

Етап планування сприяє ефективному спілкуванню та взаємодії всередині команди. Визначення чітких ролей, обов'язків та очікувань формує міцний фундамент для успішного процесу розробки програмного забезпечення.

Другий етап – **аналіз**. Він спрямований на визначення конкретних очікуваних результатів від кінцевих користувачів щодо програмного забезпечення. Команда на цьому етапі має за мету з'ясувати: "Які вимоги висуваються користувачами до нашого програмного продукту?" Це включає збір необхідної інформації.

Команда активно збирає інформацію від зацікавлених сторін, таких як: кінцеві користувачі, замовники, аналітики, менеджери через різні види діяльності, такі як: інтерв'ю, опитування та фокус-груп з метою вивчення очікувань та потреб користувачів.

Після збору даних команда проводить їх аналіз, виділяючи ключові характеристики від необов'язкових. Цей аналіз допомагає розуміти функціональність програмного забезпечення. Результатом має бути документ із вимогами, який описує призначення програмного забезпечення, його основні характеристики та функціональні можливості, створюючи загальне розуміння для розробників подальшого спектру робіт [9]. Цей документ перевіряється на точність, повноту та реальність у реалізації.

Успішний аналіз вимог є ключовим для успішної реалізації проєкту. Правильно зроблений аналіз допомагає сформулювати чітке завдання для створення програмного рішення, яке повністю задовольнить потреби користувачів.

Третій етап – **проєктування**. Під час цього етапу розробляється основа програмного продукту. Команда розробників відповідає за створення програмного забезпечення, описуючи його функціональність та зовнішній вигляд. Головна увага приділяється структурі програми, навігації, інтерфейсу

користувача та організації бази даних. Цей етап гарантує, що програма буде зручною для користувача та ефективно виконає свої завдання.

Команда ставить перед собою декілька завдань. Це включає створення діаграм потоку даних, діаграм сутностей та зв'язків, а також розробку макетів інтерфейсу користувача. Крім того, визначаються системні залежності та точки інтеграції. Також встановлюються обмеження програмного забезпечення, такі як апаратні обмеження та вимоги до продуктивності. Тобто, роблять дорожню карту для команди під час написання коду програми. Цей етап детально описує дизайн програмного забезпечення, включаючи архітектуру системи, дизайну застосунку та особливості інтерфейсу користувача.

Четвертий етап – **реалізація**. Фактично це означає реалізацію додатку, а отже, перехід до написання коду. Інженери та розробники перетворюють дизайн програмного забезпечення в фактичний код [10].

Цей етап спрямований на створення програмного забезпечення, яке буде функціональним, ефективним та зручним для користувача. Розробники використовують відповідні мови програмування для реалізації коду, дотримуючись усіх вимог.

Ще одним ключовим аспектом цієї фази є систематичні ревізії коду. Члени команди уважно переглядають роботу один одного, щоб виявити будь-які помилки чи невідповідності. Ці перевірки допомагають підтримувати високі стандарти коду та забезпечують надійність програмного забезпечення. Також проводиться попереднє внутрішнє тестування, щоб переконатися у базовій функціональності програми.

П'ятий етап – **тестування та впровадження**.

Етап тестування включає аналіз на виявлення помилок чи недоліків в коді, які можуть виникнути під час розробки. Головна мета – переконатися, що програмне забезпечення працює бездоганно до того, як його отримають кінцеві користувачі, а також ідентифікувати можливості для подальшого вдосконалення.

Процес тестування розпочинається з встановлення чітких критеріїв відповідно до вимог програмного забезпечення. Це охоплює визначення умов і окреслення різних сценаріїв для їх перевірки. Це допомагає сформувати ефективну стратегію тестування. Після створення тестових сценаріїв тестувальники проводять докладне різне тестування програмного забезпечення. Вони перевіряють різні аспекти, починаючи від окремих компонентів до роботи всієї системи.

Якщо тест виявляє помилку, вона докладно документується з описом, методами відтворення та впливом на програмне забезпечення. Ці помилки повертаються розробникам для виправлення. Після внесення необхідних змін програмне забезпечення знову проходить тестування. Цей цикл продовжується, поки програмне забезпечення не відповідає усім встановленим вимогам.

Після успішного тестування програми настає час представлення його для користувачам у виробничому середовищі. Етап включає у себе впровадження та налаштування програмного забезпечення перед його використанням кінцевими користувачами.

Проте розгортання не обмежується лише запуском програмного забезпечення. Важливо забезпечити, щоб користувачі могли легко й ефективно ним користуватися. Це може означати створення посібників для користувачів, проведення навчальних сесій або надання підтримки на місці.

Етап означає перехід від фази розробки до фази використання продукту, коли програмне забезпечення починає виконувати свої призначення.

Шостий етап – **супровід**. У життєвому циклі розробки програмного забезпечення фаза обслуговування орієнтується на постійну підтримку та вдосконалення ПЗ, щоб забезпечити найефективніше функціонування продукту відповідно до очікувань клієнтів.

Ключовим аспектом є адаптація програмного забезпечення до змінних потреб користувачів. Це означає відповідну реакцію на відгуки користувачів, вирішення непередбачених проблем та оновлення програмного забезпечення за новими вимогами клієнтів. Це безперервний процес покращення.

Важливою частиною є також підтримка користувачів, надання допомоги і порад тим, хто має проблеми з програмним продуктом.

На етапі обслуговування також розглядаються довгострокові стратегії, такі як оновлення або заміна програмного забезпечення. Це рішення залежить від життєвого циклу продукту та технологічних тенденцій.

Життєвий цикл розробки програмного забезпечення є важливою основою для ефективного управління проектами з розробки програмного забезпечення. Дотримуючись структурованого процесу, який включає планування, аналіз вимог, проєктування, реалізацію, тестування, впровадження та супровід, організації можуть покращити управління проєктами, забезпечити якість програмного забезпечення, управляти ризиками та підвищити рівень задоволеності клієнтів.

1.3. Особливості роботи SRUM команди

Перед розглядом особливостей роботи Scrum-команди важливо зазначити, що існує кілька підходів до управління проектами, серед яких одним з традиційних є методологія водоспаду. Особливістю даного підходу є його лінійний характер, що передбачає послідовну реалізацію етапів розробки, де кожен новий етап розпочинається лише після завершення попереднього. Такий підхід часто використовується у великих і вибіркових проєктах, де вимагається чітка установка вимог та мінімізація змін протягом процесу розробки.

Однак, з плином часу, методологія Scrum, що належить до сімейства Agile, набула значної популярності, особливо в умовах швидкозмінних технологічних ринків та постійних вимог до швидкості реакції на зміни.

Його методологія базується на ітераційному підході, що дозволяє командам розробників ефективно пристосовуватися до змін у вимогах, сприяючи швидкому випуску та постійному вдосконаленню продукту.

Scrum створює структуру для гнучкого та ефективного керування проєктами, сприяючи взаємодії між командою розробників та замовником, що призводить до зниження ризиків та забезпечує високу результативність [11]. У цьому

контексті, методологія Scrum стала важливим інструментом для компаній, що прагнуть досягти успіху в умовах швидко змінюючогося ринкового середовища.

Відповідно до попереднього розділу, команда для розробки програмного забезпечення доволі комплексна. Scrum-команда, що є ключовим інструментом у реалізації Scrum, являє собою міжфункціональну команду, що складається з різних співробітників, кожен із яких має спеціалізовані звання.

Команда Scrum складається з одного Scrum-майстра, одного product owner та розробників [12].

Scrum Master (SM): Однією з основних обов'язків Scrum Master є забезпечення дотримання командою принципів Scrum, таких як спринти та щоденні зустрічі. Вони, як вправні координатори, підтримують єдність команди та ефективно вирішують будь-які виникаючі проблеми.

Власник продукту (PO) – визначає очікування зацікавлених сторін щодо продукту та забезпечує, щоб розробка була орієнтована на досягнення цих цілей. Вони є посередниками між бізнесом і розробниками, приймаючи обґрунтовані рішення щодо функціоналу продукту, зокрема, враховуючи фінансові та операційні аспекти [13].

Члени команди Scrum (розробники): Це ключові фігури команди Scrum, які працюють під керівництвом досвідченого керівника. Їх завдання – ефективно виконувати спринти та забезпечувати оптимальні витрати, постійно вдосконалюючи робочі процеси. Кожен розробник має різноманітні навички, включаючи проектування, програмування, тестування та аналіз бізнес-процесів. Тренери Scrum допомагають розробникам у підвищенні кваліфікації та наданні високоякісних послуг.

У складі команди Scrum немає підкоманд або ієрархій. Це колектив професіоналів, що мають спільну мету. Команди Scrum є крос-функціональними, що означає, що учасники мають всі необхідні навички для створення цінності. Вони також самоорганізовані, що означає, що вони внутрішньо вирішують, хто, коли і як. Команда Scrum доволі невелика,

зазвичай 10 або менше людей. Загалом дослідження показують, що менші команди краще спілкуються і є більш продуктивними. Якщо команди Scrum стають занадто великими, вони повинні розглянути можливість реорганізації в декілька єдиної команд Scrum, кожна з яких фокусується на одному продукті. Команда відповідає за всі продуктові діяльності, від співпраці з зацікавленими сторонами, до розробки та верифікації. Вони самостійно організовані для керування своєю власною роботою [12].

Розглянемо детальніше особливості SCRUM команди:

- Самоорганізація: Команда SCRUM не має жорсткої ієрархії. Її члени самі організовують свою роботу, визначають завдання та розподіляють відповідальність.
- Крос-функціональність: Складається з фахівців з різних дисциплін, таких як розробники, тестувальники, аналітики та менеджери продуктів. Це дозволяє їм самостійно вирішувати всі питання, пов'язані з розробкою продукту.
- Невеликий розмір: Зазвичай складається з 3-10 осіб. Це робить її більш гнучкою та адаптивною до змін.
- Прозорість: Уся інформація про проект є доступною для всіх членів команди. Це забезпечує кращу комунікацію та координацію дій.
- Відповідальність: Кожен член команди несе відповідальність за результат роботи.
- Постійне вдосконалення: Команда постійно шукає шляхи для покращення свого процесу роботи. Це робиться за допомогою ретроспектив, де члени команди аналізують свою роботу, визначають проблеми та шукають шляхи їх вирішення.

Основні переваги команди SCRUM полягають у наступному [14]:

- Ефективність та швидкість – SCRUM команди подібно до злагоджених оркестрів, де кожен учасник чітко знає свою роль. Завдяки цьому, розробка продукту йде швидше, адже чітка структура та ітеративний

підхід дозволяють фокусуватися на конкретних завданнях, уникаючи простоїв.

- Висока якість та адаптивність – команда не просто генерує код, а й постійно вдосконалюють продукт, роблячи його максимально зручним та функціональним. Завдяки постійному зворотному зв'язку та гнучкості, продукт еволюціонує, відповідаючи актуальним потребам користувачів.
- Мотивація та командний дух – в команди завжди присутнє відчуття свободи та відповідальності. Кожен член команди відчуває себе важливим елементом механізму, що генерує спільний результат. Це веде до самоорганізації, командної роботи та високої мотивації, адже члени команди розуміють, що їхня робота має значення.

Хоч команда має доволі багато переваги та приносить користь, також є і недоліки, а саме:

- Складність для новачків – SCRUM, як і будь-який інструмент, потребує знання та досвіду. Командам, які вперше стикаються з цим фреймворком, може знадобитися час на адаптацію та освоєння його принципів.
- Відсутність жорсткої ієрархії – деяким людям може бути складно працювати без чіткої вертикалі управління. SCRUM дає більше свободи, але й потребує самодисципліни та відповідальності від кожного члена команди.
- Важливість комунікації та самоорганізації – ефективна робота SCRUM команди залежить від чіткої комунікації та злагоденої самоорганізації. Якщо в команді виникають проблеми з цими аспектами, це може негативно вплинути на результат.

У крос-функціональному Scrum процесі прийняття рішень є ключовим елементом успішної роботи команди. Учасники команди мають можливість висловлювати свої думки та внести власний внесок у колективне прийняття рішень під час зборів, таких як щоденні зустрічі, планування спринту та огляди. Scrum сприяє простому та прозорому процесу прийняття рішень, сприяє оперативності та постійному вдосконаленню. Команди також мають

інструменти для вирішення конфліктів, які можуть виникати в процесі, що допомагає їм досягати успіху та вдосконалювати свою роботу.

Існують специфічні принципи, які слід дотримуватися при ухваленні колективних рішень.

По-перше, важливо досягти згоди серед учасників команди, якщо це можливо. Це означає, що всі члени мають погодитися з пропонованим вирішенням до його прийняття. Якщо один з учасників не згоден з прийнятим рішенням, інші члени команди спільно працюють з ним, щоб вирішити його застереження. Цей процес передбачає пошук взаємоприйнятної рішення, яке відповідає всім учасникам команди.

По-друге, у випадках, коли досягнення консенсусу неможливе, можливе проведення голосування. Кожен член команди має можливість висловити свою думку, і тільки при наявності чіткої більшості буде прийняте відповідне рішення.

У методології Scrum консенсус передбачає загальну згоду членів команди стосовно конкретних дій, процесів і рішень, пов'язаних із розробкою продукту. Це досягається завдяки відкритим каналам зв'язку, які дозволяють всім учасникам висловлювати свої думки та ідеї, без критики чи осуду.

Scrum-команди можуть стикатися з різними проблемами, такими як недостатня комунікація, великі завдання, недоліки управління продуктом, недосконале використання методології Scrum та відсутність ретроспектив. Для подолання цих проблем важливо забезпечити ефективну комунікацію у команді, розбивати завдання на менші елементи, забезпечувати зрозумілі вимоги до продукту, налагоджувати процес роботи з використанням Scrum та проводити регулярні ретроспективні зустрічі для аналізу і вдосконалення процесів роботи.

У підсумку, SCRUM-команда є динамічним та гнучким організаційним одиницею, яка забезпечує швидкий розвиток програмного забезпечення з врахуванням змін у вимогах та умовах ринку. Її особливості відображають принципи співпраці, взаємодопомоги та постійного вдосконалення, що робить її невід'ємною складовою успішних проектів розробки ПЗ.

1.4. IT інструменти, що використовуються у роботі бізнес аналітика

Бізнес-аналітик в IT – це фахівець, який забезпечує зв'язок між бізнес-потребами і технічними можливостями в сфері інформаційних технологій [15]. Він аналізує потреби бізнесу, визначає вимоги до програмного забезпечення та інформаційних систем, спілкується з клієнтами і розробниками, тестує і валідує програмне забезпечення, а також надає підтримку бізнесу під час впровадження нових рішень. Він відіграє ключову роль у забезпеченні того, щоб розроблені продукти відповідали потребам і стратегічним цілям підприємства, допомагаючи вирішувати бізнес-проблеми та підвищувати ефективність управління.

В своїй роботі бізнес-аналітик використовує різноманітні IT-інструменти для збору, аналізу та візуалізації даних. Нижче наведені деякі з найпопулярніших IT-інструментів, які використовуються у роботі [16]:

- [Google Sheets](#) [17] - це інструмент, який, здається, всім відомий зі шкільних років. Проте його можливості далеко виходять за межі простого створення таблиць та ведення розрахунків. Часто він використовується замовниками, зокрема в фінансовій та продажній сферах, як основний інструмент для роботи з даними. Зведені таблиці для аналізу даних є базовими функціями для бізнес-аналітиків, які працюють з великими обсягами інформації.
- [Google Presentations](#) [18] можна використовувати для презентації комерційних пропозицій та демонстрації майбутніх планів користувачам. При створенні презентації рекомендується спочатку прописувати заголовки слайдів, а потім наповнювати їх контентом, щоб уникнути перевантаження інформацією.
- [Google Docs](#) [19] часто використовуються для складання чек-листів та документації невеликих проєктів. Для великих проєктів їх також можна використовувати, але, можливо, не так ефективно.

- [Google Disk](#) [20] використовується для зберігання та передачі інформації. Це особливо важливо для бізнес-аналітиків, які часто діляться інформацією та об'ємними медіафайлами зі своєю командою та замовниками. Пересилання проєктної документації поштою може бути довгим та незручним, а також небезпечним з точки зору безпеки.
- [Slack](#) [21] / [MS Teams](#) [22] / [Zoom](#) [23] - це найпопулярніші інструменти комунікації в ІТ-сфері, які дозволяють зручно спілкуватися та обмінюватися інформацією.
- [Balsamiq](#) [24] - це програма для створення макетів та прототипів інтерфейсів будь-якої складності.
- [Figma](#) [25] - ще один потужний інструмент для візуалізації систем та розташування елементів інтерфейсу.
- [Axure RP](#) [26] дозволяє створювати динамічні прототипи з переходами між сторінками. Всі ці інструменти є важливими для бізнес-аналітиків у сфері ІТ.
- [Diagrams.net](#) [27], [Visual Paradigm](#) [28] та [Lucid Chart](#) [29] - це потужні інструменти для створення діаграм.

Diagrams.net має велику бібліотеку елементів, включаючи UML та BPMN, але не виконує перевірку нотацій. Натомість Visual Paradigm спеціалізується на цьому та має безкоштовну веб-версію.

- Крім того, варто звернути увагу на [Bizagi](#) [30] та [Camunda](#) [31] - це потужні платформи не лише для створення діаграм, але й для автоматизації бізнес-процесів.

Отже, кожен з цих інструментів має свої унікальні функції та можливості, які сприяють покращенню робочих процесів бізнес-аналітиків. Важливо вміло використовувати ці інструменти з метою збільшення ефективності роботи, поліпшення спілкування та забезпечення якісної аналітичної роботи. У цілому, правильний вибір інструментів і їх раціональне використання є ключовими елементами успішної та ефективної роботи бізнес-аналітичної команди.

1.5. ІТ інструменти, що використовуються у роботі проджект менеджера

Проджект менеджер в ІТ – це фахівець, відповідальний за керування проектами в галузі інформаційних технологій [32]. Він забезпечує успішну реалізацію проектів з розробки програмного забезпечення, впровадження нових технологій, створення інформаційних систем тощо.

Завдання проджект менеджера в ІТ включають планування, організацію, виконання та контроль проектних процесів [33]. Він співпрацює з різними відділами компанії, замовниками та командами розробників для досягнення поставлених цілей проекту.

Проджект менеджер в ІТ володіє різноманітними навичками, такими як планування ресурсів, управління витратами, комунікації зі зацікавленими сторонами, ризик-менеджмент, вирішення конфліктів та управління змінами [34]. Він також може використовувати різноманітні інструменти ІТ для кращого керування проектами, спілкування та аналізу даних, а саме:

- Програми для управління проектами (Project Management Tools): Такі як [Jira](#) [35], [Asana](#) [36], [Trello](#) [37], [Microsoft Project](#) [38]. Ці інструменти допомагають встановлювати завдання, призначати їх відповідальним особам, встановлювати терміни виконання та відстежувати прогрес робіт.
- Системи комунікації (Communication Systems): Сервіси для спілкування в реальному часі, такі як [Slack](#) [21], [Microsoft Teams](#) [22], а також електронна пошта. Ці інструменти дозволяють команді проджект менеджера ефективно обмінюватися інформацією, спілкуватися та вирішувати питання.
- Інструменти для спільного доступу до документів (Collaboration Tools): [Google Drive](#) [20], [Dropbox](#) [39], [SharePoint](#) [40]. Вони дозволяють зберігати та обмінюватися документами, редагувати їх в реальному часі та забезпечувати доступ до них з будь-якого пристрою.
- Інструменти для планування (Planning Tools): [Microsoft Excel](#) [41], [Google Sheets](#) [17]. Вони допомагають створювати графіки, таблиці, діаграми та

інші інструменти для планування ресурсів, графіків виконання та бюджетів проекту.

- Інструменти відстеження часу (Time Tracking Tools): [RescueTime](#) [42], [Toggl](#) [43], [Clockify](#) [44]. Ці інструменти допомагають відстежувати час, витрачений на різні завдання, а також оцінювати продуктивність та ефективність роботи.
- Інструменти для віртуальних зустрічей (Virtual Meeting Tools): [Zoom](#) [23], [Skype](#) [45], [Microsoft Teams](#) [22]. Завдяки цим інструментам, команда може здійснювати віддалені зустрічі, проводити онлайн-конференції та обговорювати плани проекту.

З цікавих ресурсів, також хочу відмітити наступні:

- [Loom](#) [46] – це програма, яка записує відео зі звуком з екрана та/або вебкамери і дозволяє надсилати його електронною поштою або ділитися посиланням.
- [Canva](#) [47] – це відмінний сервіс для створення презентацій, інформаційних буклетів, інфографіки та іншого контенту, що відзначається великою кількістю шаблонів і сучасним дизайном.
- [Read Aloud](#) [48] – це розширення для браузера, яке перетворює текстові файли на аудіо, що може бути корисним для швидкого ознайомлення з вмістом документів.

Використання цих ІТ-інструментів допомагає проджект менеджерам ефективно керувати різними аспектами проектів, підтримувати комунікацію в команді та замовниками, а також забезпечує гнучкість та швидкість реакції на зміни в проєкті.

РОЗДІЛ II

РОЛЬ БІЗНЕС АНАЛІТИКА ПІД ЧАС КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Основні завдання та функції бізнес-аналітика в проєкті

Бізнес-аналітик — це посередник між бізнесом і командою розробників. Його функції можуть відрізнятися залежно від розміру компанії, величини і тривалості проєкту [49].

Зазвичай роль даного спеціаліста включає шість основних областей діяльності, які взаємодіють між собою:

1. Взаємодія з клієнтами та зацікавленими сторонами для визначення бізнес-потреб, збір і аналіз вимог.
2. Документування вимог і відстеження їх змін.
3. Стратегічний аналіз, що включає вивчення бізнесу клієнта для забезпечення ефективних рішень.
4. Розробка рішень у співпраці з командою розробників з урахуванням бізнес-процесів клієнта.
5. Планування власних та командних завдань розробки, а також відстеження результатів.
6. Оцінка прийнятих рішень.

Проводячи аналіз кожної з складових складає рекомендації щодо оптимізації бізнес-процесів та прийняття управлінських рішень, заснованих на даних. Для досягнення поставлених цілей, бізнес-аналітик використовує різноманітні інструменти та методи аналізу даних. Серед них варто відзначити статистичний аналіз, машинне навчання, бізнес-моделювання тощо. Ці інструменти дозволяють аналізувати великі обсяги даних, виявляти закономірності та патерни, що важливі для бізнесу, та робити прогнози на їх основі.

Бізнес-аналіз надає організаціям можливість не лише переконатися в ефективності технології для підтримки їхньої діяльності, але й виявити оптимальні варіанти для впровадження змін, ураховуючи обмеження з боку бюджету та часу. Бізнес-аналітики здатні надати об'єктивні оцінки, що можуть

піддати сумніву загальноприйняті погляди, виявити кореневі причини проблем та визначити стратегії змін, спрямовані на досягнення реальних переваг для підприємства [50].

2.2 Ролі та відповідальності бізнес-аналітика в команді розробки, та його вплив на якість та успіх проєкту

У сучасному бізнесі бізнес-аналітики грають ключову роль у прийнятті стратегічних та тактичних рішень. Вони допомагають бізнесу розуміти своїх клієнтів, виявляти потреби ринку, визначати конкурентні переваги та недоліки, а також вдосконалювати бізнес-процеси з метою підвищення ефективності та прибутковості [51]. А для досягнення таких задач, потрібно мати гарно натреновані soft skills.

Початково, варто зазначити, що лідерські якості бізнес-аналітика виявляються в його здатності ефективно керувати командою та спрямовувати її зусилля на досягнення поставлених цілей. Це включає у себе формулювання вимог до проєкту, виявлення стратегій вирішення проблем та спрощення комунікації всередині команди. Чітко визначені вимоги та бізнес-процеси, а також їхня відповідність потребам клієнтів, дозволяють уникнути помилок та забезпечити успішне виконання проєкту. Крім того, глибоке розуміння потреб бізнесу дозволяє створити продукт, що повністю задовольняє вимоги клієнта та має конкурентні переваги на ринку.

Далі, співпраця бізнес-аналітика з розробниками, адже вони спільно працюють над розумінням та реалізацією вимог до програмного продукту, що гарантує відповідність результату очікуванням замовника та його бізнес-потребам.

Тестування вимог, ще одна функція, яка відображає його здатність перевіряти правильність та відповідність вимог бізнес-процесам та бізнес-потребам. Це є необхідним етапом для забезпечення якості та ефективності програмного продукту.

Окрім того, бізнес-аналітики відіграють важливу роль у впровадженні нових технологій та інновацій у бізнесі. Вони допомагають виявляти можливості для

впровадження нових продуктів та сервісів, оцінюють їхній потенціал та ризики, а також визначають стратегії їхнього впровадження.

У підсумку, бізнес-аналітик є невід'ємною складовою сучасного бізнесу, який допомагає підприємствам адаптуватися до змінного середовища, приймати обґрунтовані рішення і досягати успіху в умовах постійної конкуренції.

2.3 Приклади задач, що вирішувались бізнес аналітиком під час роботи над проєктом обрання наукового керівника

2.3.1 Epics and User stories.

У методології Agile термін "Epics" відноситься до великих та складних обсягів роботи, які можуть бути поділені на менші завдання, відомі як "user stories". Epics використовуються для організації та категоризації роботи, надаючи високорівневе уявлення про вимоги до проєкту. Зазвичай Epics занадто об'ємні, щоб бути завершеними протягом однієї ітерації, тому їх розбивають на більш дрібні user stories. Таким чином, Epics служать засобом фіксації та передачі загального обсягу функціоналу або групи пов'язаних між собою функцій. Вони створюють фундамент для подальшого розбиття на деталізовані user stories, які розробляються на етапах планування та виконання проєкту. Дані документи відіграють важливу роль у сприянні структурованому та ітеративному наданні задач, створюючи стратегічну основу для досягнення цілей проєкту [52].

Під час участі у проєкті обрання наукового керівника в якості бізнес-аналітика, було створено ряд документів user stories. Однією з частин одного з цих документів є наступне:

USER STORY 1 USER PORTFOLIO

View Interests

URL: /scinterests/{PersonId}

Method: GET

Expected result: повертає коло інтересів для вказаної персони

Add Interests

URL: /scinterests/{PersonId}

Method: POST

Expected result: додає коло інтетересів для заданої персони. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

Update Interests

URL: /scinterests/{PersonId}

Method: PUT

Expected result: оновлює коло інтетересів для заданої персони. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

View person working project experience

URL: /personprojects/{PersonId}

Method: GET

Expected result: повертає проектний досвід вказаної персони.

Contract: дата початку, дата кінця, назва проекту, роль.

View working project experience details

URL: /personproject/{ProjectId}

Method: GET

Expected result: повертає проектний досвід вказаною персони коло інтетересів для заданої персони.

Contract: повний опис проекту.

Add working project experience

URL: /personproject/{PersonId}

Method: POST

Expected result: додає до вказаної персони, новий проектний досвід. У подальшому буде доступним лише для авторизованих користувачів (ключ). У

разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

Update working project experience

URL: /personproject/{ProjectId}

Method: PUT

Expected result: оновлює вже існуючий проектний досвід. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

Delete working project experience

URL: /personproject/{ProjectId}

Method: DELETE

Expected result: видаляє вже існуючий проектний досвід персони. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

View external profile types

URL: /externalprofiletypes/

Method: GET

Expected result: Повертає тип зовнішнього профілю (Google Академія, Scopus, Оркіт тощо)

Add external profile type

URL: /externalprofiletype/

Method: POST

Expected result: Додає тип зовнішнього профілю

Update external profile type

URL: /externalprofiletype/{Id}

Method: PUT

Expected result: Оновлює тип зовнішнього профілю

Вище наведено лише фрагмент даної роботи, так як через обмеженість в об'ємі роботи неможливо навести весь. Більше розписаних user story знаходяться в додатку 1.

Даний набір user stories визначає функціональні можливості API для керування інформацією про користувачів та їхній проектний досвід. Основні дії, які можна виконати за допомогою цих API, включають перегляд, додавання, оновлення та видалення інформації.

Наприклад, user story "View Interests" визначає можливість переглядати інтереси користувача за його ідентифікатором. Аналогічно, user story "Add Interests" дозволяє додавати нові інтереси для вказаної персони. У випадку невдачі, система повинна повернути відповідний HTTP статус, що відповідає проблемі.

User stories, пов'язані з досвідом викладачів, також визначають можливості отримання, додавання, оновлення та видалення інформації. Наприклад, user story "View person working project experience" надає можливість отримати детальну інформацію про досвід вказаної персони. Аналогічно, user story "Add working project experience" дозволяє додавати новий досвід для користувача.

Кожна з цих дій має свій URL та метод, що визначається HTTP запитом (GET, POST, PUT, DELETE). Також вказуються очікувані результати та статуси відповідей у випадку успішного чи неуспішного виконання операції.

2.3.2 Взаємодія з замовником

У рамках взаємодії з замовниками було проведено опитування серед студентів щодо процесу обрання наукового керівника з метою оцінки їхньої задоволеності даним процесом. Опитування було спрямоване на збір даних про думку студентського колективу щодо ефективності процедур обрання, а також на виявлення можливих аспектів, які можна покращити в майбутньому. Цільовою аудиторією даного опитування були студенти бакалаврату, магістратури та випускники університетів України, віком від 18 до 27 років.

Методика опитування включала в себе створення опитувальника у вигляді гугл форми з питаннями, що охоплювали різні аспекти процесу вибору наукового керівника. Питання були сформульовані з урахуванням ключових етапів

процедури обрання, таких як оцінка критеріїв вибору та організація остаточного вибору. Після розробки опитувальника він був розповсюджений серед студентів та випускників через месенджери, які брали участь у процесі вибору наукового керівника.

В додатку 2 підкріплений шаблон опитування для ознайомлення.

Дане анкетування пройшло 50 кількість студентів. Дані, отримані в результаті опитування, були аналізовані з метою виявлення позитивних та негативних аспектів процесу обрання наукового керівника.

Результати наведені нижче:

Ваш статус на момент опитування

отримав диплом
бакалавра/магістра/спеціаліста і зараз
не навчаюся

студент бакалаврату

студент магістратури

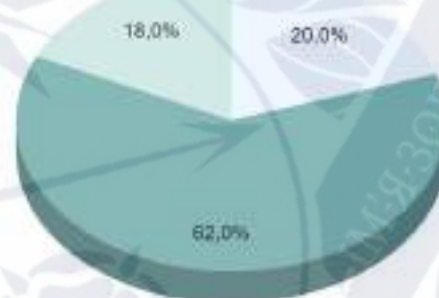


Рис. 2 - результат опитування

Вкажіть назву навчального закладу, в якому ви навчаєтесь/навчалися

Закарпатський економіко-
2.0%

Миколаївський державний
2.0%

Вінницький торговельно-економічний
2.0%

Полтавський національний педагогічний
2.0%

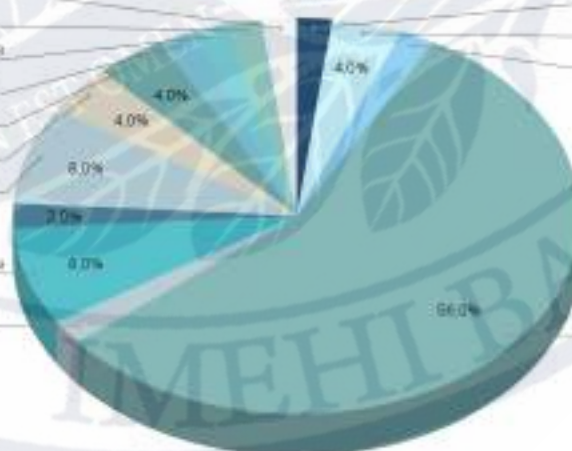
Вінницький національний
4.0%

Вінницький національний аграрний
4.0%

Вінницький національний технічний
8.0%

Дніпропетровський торговельно-економічний
8.0%

Уманський державний педагогічний
2.0%



Вінницький державний педагогічний
2.0%

Вінницький державний педагогічний
4.0%

Київський національний університет
2.0%

Дніпропетровський торговельно-економічний
8.0%

Рис. 3 - результат опитування

Яким чином ви обирали свого наукового керівника?

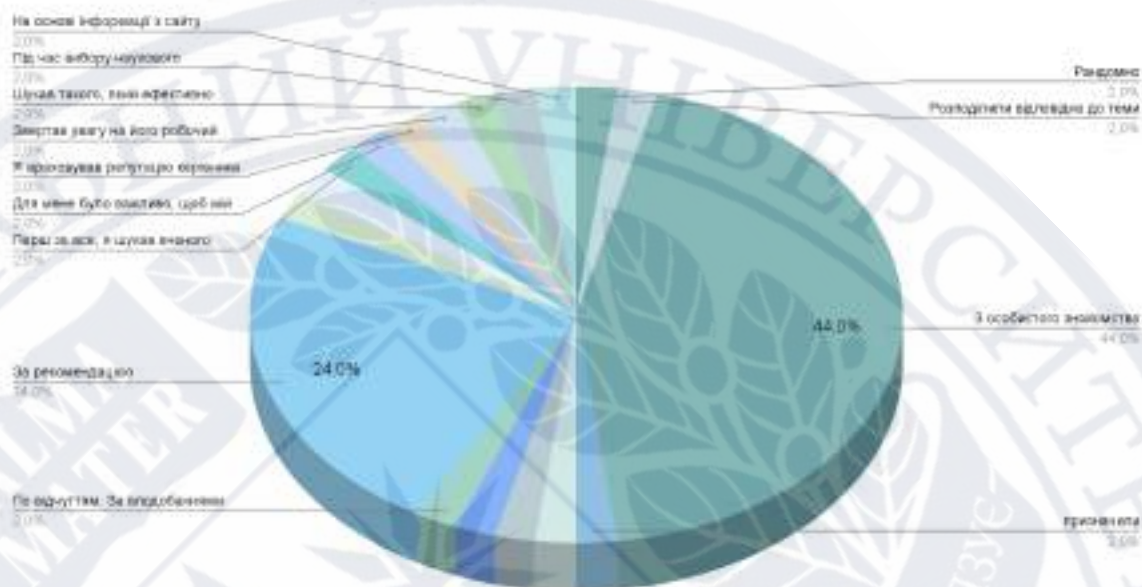


Рис.4 - результат опитування

Наскільки прозорим на вашу думку є процес обрання наукового керівника у вашому університеті

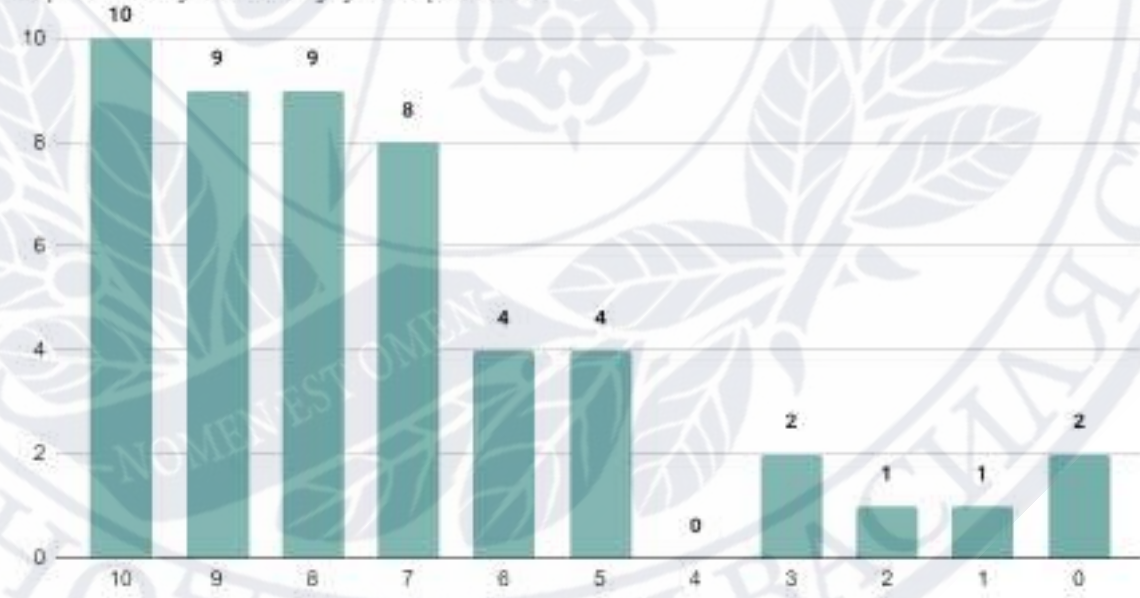


Рис. 5 - результат опитування

З якими труднощами ви стикалися при виборі наукового керівника?

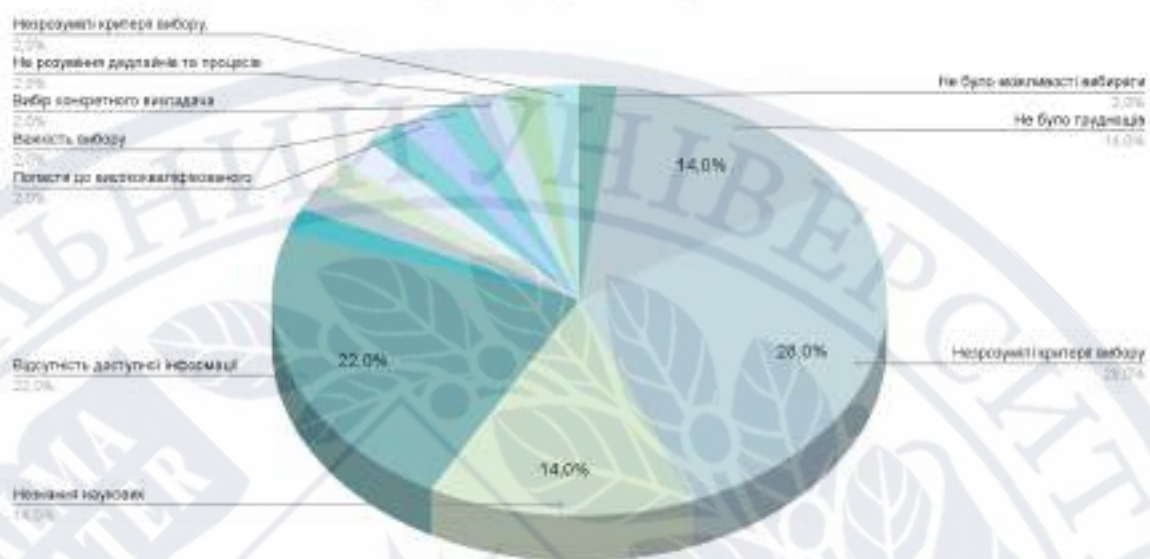


Рис. 6 - результат опитування

Які фактори для вас найважливіші при виборі наукового керівника?

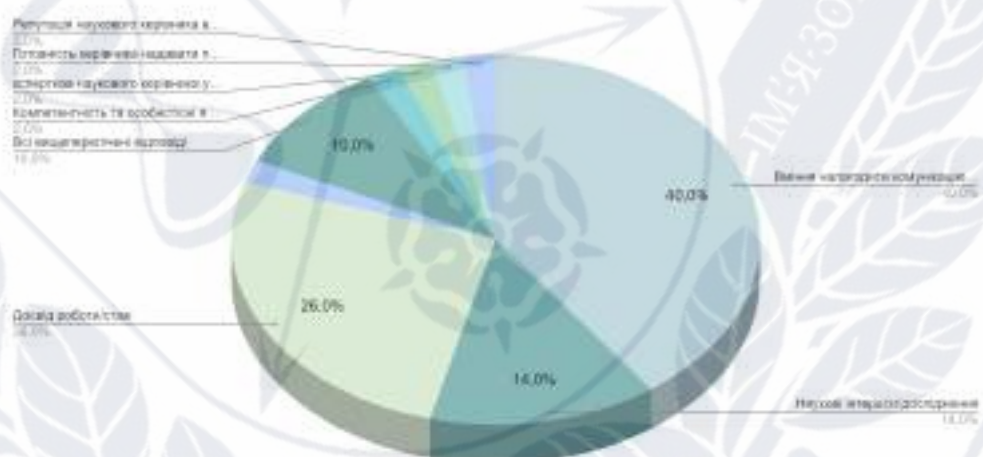


Рис. 7 - результат опитування

Чи вважаєте ви, що автоматизована система допомогла б вам прийняти більш обґрунтоване рішення щодо вибору керівника?

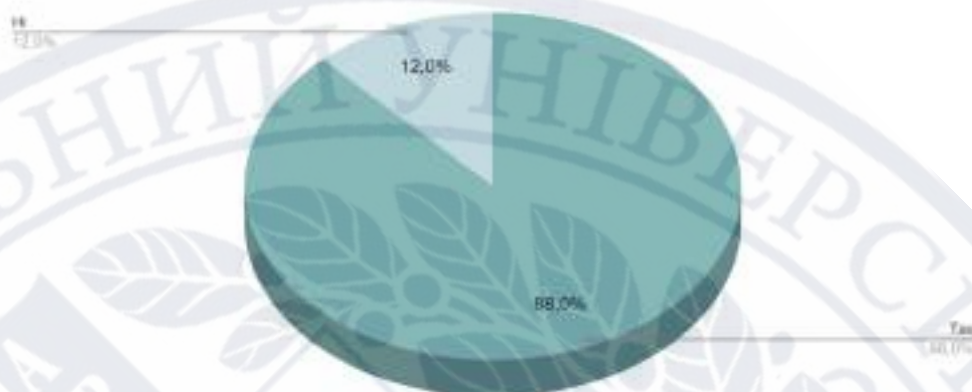


Рис. 8 - результат опитування

Чи допомогла би вам наявність розширеного портфолію викладачів у процедурі вибору керівника

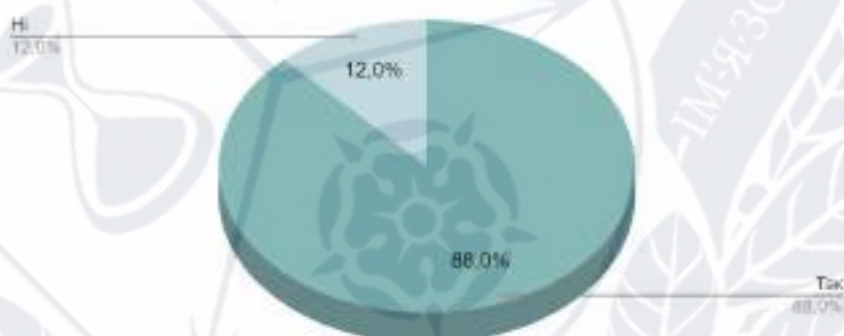


Рис. 9 - результат опитування

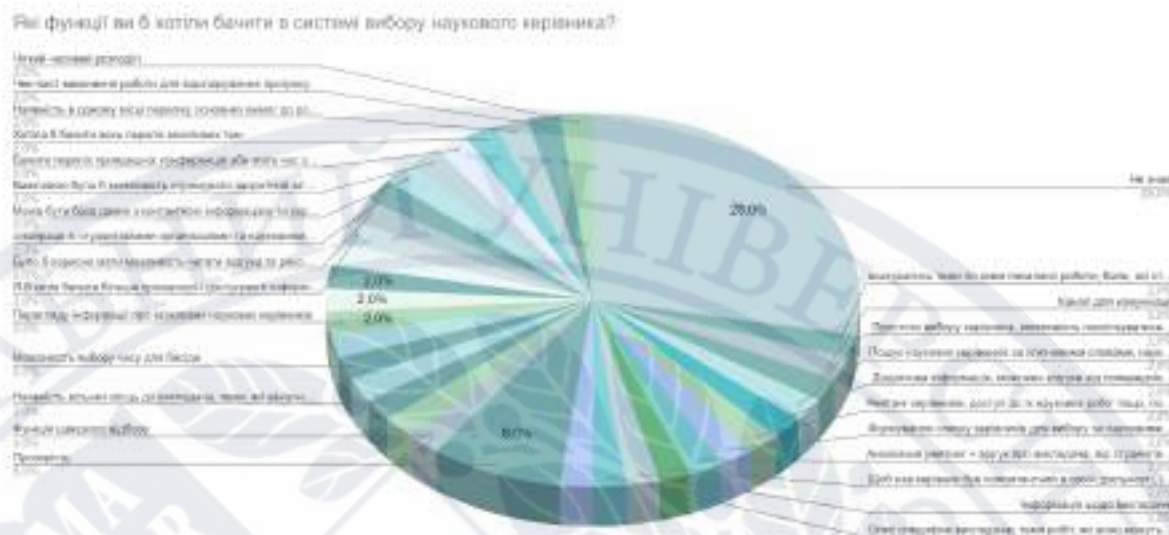


Рис. 12 - результат опитування

Дані відповіді показують наявну систему обрання наукового керівника з певними сторонами для покращення. Зокрема, увага була зосереджена на виявленні можливих проблемних моментів таких як: прозорість та анонімність; недостатнє розуміння попереднього досвіду керівництва та переліку сфер, які цікаві викладачу; складність процесу обрання, відсутність єдиної нормативної бази з матеріалами, тощо.

Отримані результати опитування стали основою для формулювання пропозицій щодо подальшого вдосконалення процедур обрання наукового керівника - а саме фокусування на прозорості та анонімності і полегшенню взаємодії з додатком. Обґрунтовані рекомендації щодо покращення механізму опитування мають на меті підвищення ефективності та якості процесу обрання, що сприятиме подальшому успішному виконанню проєкту.

2.3.3 Взаємодія зі стейкхолдерами

Взаємодія з зацікавленими особами виявляється важливим кроком під час процесу розробки додатку, оскільки вони вносять суттєвий внесок у його концепцію, функціональність та використання.

Ця взаємодія включає комунікацію з науковим керівником як представником викладацького складу та ключовим стейкхолдером, адже керівник допомагає в формуванні загального бачення щодо функціональних вимог та особливостей,

які повинні бути враховані у програмному забезпеченні. Взаємодія з ним включала регулярні зустрічі для обговорення вимог, презентування проміжних результатів та отримання зворотного зв'язку.

Також немало важливим була комунікація з програмістами та розробниками, адже бізнес-аналітик повинен забезпечити чітке розуміння кожного з членів команд вимог та очікувань наукового керівника та інших стейкхолдерів. Крім того, аналітик відповідає за координацію робіт, вирішення технічних проблем і забезпечення взаємодії між усіма членами команди. Регулярні зустрічі, спільні сесії постановки задач та відлагодження процесу розробки були важливими частинами цієї комунікації.

Одним з завдань, які були пророблені це SWOT аналіз з метою оцінки сильних і слабких сторін нашого продукту для глибшого розуміння кожного зі стейкхолдерів.

SWOT-аналіз (англ. Strengths, Weaknesses, Opportunities, Threats) – це простий, але потужний інструмент, який використовується для оцінки сильних і слабких сторін, а також можливостей і загроз, з якими стикається організація чи проєкт [53]. Він може бути використаний для прийняття кращих рішень щодо стратегії, планування та ресурсів.

Цей аналіз допомагає розуміти переваги та недоліки програмного забезпечення, виявляти можливості для вдосконалення та потенційні загрози [54]. З цими відомостями можна розробити стратегії управління ризиками, плани розвитку і підвищити конкурентоспроможність проєкту. Крім того, такий аналіз допомагає визначити потреби в ресурсах і зосередити увагу на аспектах, які сприятимуть досягненню успіху.

Крім того, SWOT-аналіз може стати інструментом для постійного моніторингу та оновлення стратегій, оскільки бізнес-середовище постійно змінюється, і необхідно відповідати на нові виклики і можливості. Таким чином, SWOT-аналіз виявляється важливим етапом стратегічного управління, який сприяє досягненню успіху та стійкості в конкурентному бізнес-середовищі.

Даний аналіз має наступну структуру:

Сильні сторони - визначають сильні сторони продукту. При формуванні даного переліку потрібно відповісти на наступні запитання:

- Що ми робимо краще?
- У чому переваги проєкту?
- Які фактори роблять нас гідними цієї можливості?
- У чому полягає наша ціннісна пропозиція?

Слабкі сторони визначають недоліки продукту. Під час складання цього переліку необхідно відповісти на такі запитання:

- Що ми можемо покращити?
- Що ми повинні припинити робити?
- Що є у конкурентів чого немає у нас і що впливає на нас?

Можливості ринку для вашого розвитку проєкту:

- Які ринкові тенденції впливають на дану можливість?
- Які технологічні зміни можуть відкрити нові можливості?
- Що ми повинні зробити чого ми не робили, проте можливості це вимагають?
- Як інші використовують / ставляться до цієї можливості?

Зовнішні фактори, що можуть становити потенційну загрозу для проєкту:

- Як змінюється бачення ринку?
- Що роблять конкуренти?
- Чи є слабкі місця, які можуть становити загрозу для компанії?
- Чи змінюються стандарти якості нашого продукту?
- З якими перешкодами ми стикаємося?
- Як урядові заходи впливають на нас?

Нижче представлений SWOT-аналіз для цього проєкту:

Таблиця 1 - SWOT аналіз

Сильні сторони	Слабкі сторони	Можливості	Загрози
Застосування програмного забезпечення	Залежність від даних та постійне	Існує потенціал для розвитку програмного	Потрібно забезпечувати безпеку даних,

дозволяє значно збільшити ефективність процесу вибору наукового керівника, скорочуючи час та зусилля, необхідні для пошуку й аналізу кандидатів.	оновлення разом з викладачами.	забезпечення шляхом додавання нових функцій та модулів, що розширить його застосування на різних етапах вибору наукового керівника.	адже збір та обробка особистих даних вчителів та студентів може стати об'єктом зловживання, що може негативно вплинути на довіру до програмного забезпечення.
Програмне забезпечення може бути легко масштабоване для використання в різних університетах.	Для ефективної роботи програмного забезпечення потрібна постійна підтримка та оновлення, щоб забезпечити актуальність даних та функціональність програми, а для цього постійно потрібні студенти, які залучені в підтримку додатка.	Співпраця з академічними установами задля розширення кількості користувачів.	Швидкий темп технологічних змін може призвести до швидкого застаріння програмного забезпечення, якщо воно не постійно оновлюється та адаптується до нових технологій.
Програмне забезпечення здатне надати студентам об'єктивні критерії, які допоможуть в прийнятті обґрунтованих рішень.	Інтеграція нового програмного забезпечення може вимагати часу та ресурсів на навчання користувачів.	У зв'язку з потребою в науковому керівництві у різних університетів, є можливість розширення програмного забезпечення на	Зміни в законодавстві. Поява нових правил та регуляцій може вимагати значних змін у програмному забезпеченні, що може затримати його розвиток.

		міжнародному ринку.	
Прозорість в процесі обрання (всі студенти та викладачі знають коли і як проходить опитування)		Впровадження технологій штучного інтелекту може покращити точність та швидкість вибору наукового керівника.	Поява інших програмних засобів або традиційних методів вибору керівників може ускладнити проникнення на ринок.
Використання програмного забезпечення допомагає уникнути впливу особистих вподобань чи підозрілих настроїв при виборі керівника, забезпечуючи більш об'єктивний підхід.			Недостатня довіра до автоматизованих систем може призвести до опору використанню програмного забезпечення, особливо якщо немає достатньої прозорості в алгоритмах вибору.

Програмне забезпечення має значні переваги у вигляді ефективності та автоматизації процесу вибору наукового керівника, що дозволяє скоротити час та зусилля, необхідні для аналізу кандидатів. Однак існує залежність від даних та необхідність постійного оновлення разом із викладачами, а також потреба у підтримці та навчанні користувачів.

У той же час, існують перспективи для розвитку програмного забезпечення через додавання нових функцій та співпрацю з академічними установами. Проте, швидкий темп технологічних змін може стати загрозою, адже програмне забезпечення може застаріти, якщо не буде постійно оновлюватися.

Незважаючи на це, використання програмного забезпечення допомагає уникнути впливу особистих вподобань та забезпечує більш об'єктивний підхід у виборі керівника. Таким чином, щоб забезпечити успішну впровадження проєкту, важливо враховувати всі зазначені раніше аспекти.



РОЗДІЛ III

РОЛЬ ПРОЄКТНОГО МЕНЕДЖЕРА ПІД ЧАС КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Основні завдання та функції проєктного менеджера в проєкті

Успішність будь-якого проєкту залежить від багатьох факторів, але ключове з них це управління. Тобто застосування знань, умінь, інструментів та методів до операцій проєкту для забезпечення відповідності вимогам проєкту. Управління проєктом означає спрямування роботи проєкту на досягнення запланованих кінцевих результатів [2].

Фахівець який керує реалізацією кожного проєкту - Project Manager (PM).

Виступаючи як керівник проєкту, проєктний менеджер (PM) має перед собою ряд важливих завдань, які визначають успіх проєкту в цілому [55]. Він є основним організатором, координатором та керівником усіх аспектів проєкту, починаючи від планування та закінчуючи його успішним завершенням. Основні завдання проєктного менеджера включають в себе ефективне управління ресурсами, встановлення та дотримання термінів, забезпечення якості виконання завдань, здійснення контролю над бюджетом, а також забезпечення відмінної комунікації та співпраці в команді. Його роль полягає в тому, щоб керувати процесами, мотивувати команду до досягнення поставлених цілей та реагувати на будь-які виклики та зміни, що виникають протягом проєкту [56]. Усі ці завдання вимагають від проєктного менеджера не лише стратегічного мислення, але й високої організаційної та лідерської здатності, щоб ефективно керувати проєктом і досягати його успішного завершення.

В мене як PM на даному проєкті було ряд ключових завдань, перше з них це **визначення обсягів проєкту:**

- Опис функціональності додатку:

Мета проєкту: створення додатка, для автоматизації процесу обрання наукового керівника (розподілу студентів між науковими керівниками).

Опис проекту. Проект полягає в розробці програмного засобу для обрання студентами наукового керівника для своїх наукових робіт. Ця програма має спростити процес вибору наукового керівника, забезпечити доступ до інформації про викладачів, студентів та їхні наукові досягнення, захоплення та інше. Основною метою даного продукту є допомога студентам знаходити відповідного наукового керівника, який відповідає їхнім інтересам і потребам, а також організувати автоматичний і спрощений алгоритм обрання та погодження даного виду діяльності зі сторони топ-менеджменту кафедри / університету.

Опис функціональності додатку для вибору наукового керівника для студентів включає наступні основні функції:

1. Реєстрація користувача:

- Можливість створення облікового запису для студента та наукового керівника (обліковий профіль підв'язаний до корпоративної пошти).
- Аутентифікація користувача через електронну пошту або інші методи.

2. Профіль користувача:

- Можливість створення та редагування профілю користувача.
- Зберігання основних даних про студента та наукового керівника: ім'я, контактна інформація, наукові зацікавлення тощо.

3. Пошук наукових керівників:

- Механізм пошуку наукових керівників за різними критеріями: спеціалізація, досвід роботи, наукові досягнення, коло інтересів тощо.
- Фільтрація результатів пошуку за різними параметрами.

4. Перегляд профілів наукових керівників:

- Можливість перегляду детальної інформації про кожного наукового керівника: наукові інтереси, публікації, досвід роботи тощо.

5. Вибір наукового керівника та організація роботи деканату та кафедри під час розподілу:

- Студент може обрати наукового керівника для своєї кваліфікаційної роботи за допомогою опитування, що полегшує роботу представників кафедри та деканату під час розподілу.

Визначення обмежень проєкту є ключовим етапом у процесі планування, оскільки воно уточнює цілі проєкту, встановлює обсяг доступних ресурсів (людських, фінансових, матеріальних), допомагає ідентифікувати можливі ризики та управляти ними, забезпечує реалістичність очікувань щодо досяжності цілей та фокусує увагу на головних пріоритетах проєкту. Врахування обмежень допомагає забезпечити ефективне виконання проєкту та уникнути надмірних очікувань або ресурсних конфліктів.

Тому далі буде описано детальні можливі **обмеження проєкту**:

1. Бюджетні обмеження:

- Визначення обмежень бюджету на розробку додатку.
- Необхідно здійснювати контроль над витратами, щоб не перевищувати виділений бюджет.

Проте, варто зазначити, що так як даний проєкт розробляється на волонтерських засадах, бюджетні обмеження не є основною проблемою. Замість того, щоб дотримуватися строгих обмежень у витратах, управління ресурсами в такому волонтерському проєкті може базуватися на максимально ефективному використанні доступних можливостей та ресурсів.

2. Часові обмеження:

- Проєкт повинен бути завершений у встановлені терміни, відповідно попередньо планувалась дата завершення проєкту до 31.06.2024, але в зв'язку з залученістю членів команди на написання їх

кваліфікаційних робіт, наразі проєкт перебуває на паузі, відповідно дата завершення проєкту трохи зміститься. Хочу зауважити, що обмеження часу може вплинути на обсяг роботи, функціональність та якість продукту.

3. Людські ресурси:

- Обмеженість кількості розробників програмного забезпечення, які можуть бути залучені до проєкту. На даному проєкті попередньо було 3 розробника: один Frontend-розробник та два Back-end-розробника.
- Залученість спеціаліста у галузі дизайну для створення користувацького інтерфейсу додатку.
- Обмеженість часу для тестувальника додатку, для якісного тестування.

4. Обмеження з правової точки зору:

- Авторські права та ліцензії: необхідно забезпечити, щоб усі матеріали, які використовуються у додатку (такі як текст, зображення, відео), були правовідомі. Розробники повинні мати право на використання цих матеріалів або отримати необхідні ліцензії.
- Конфіденційність даних: Додаток може збирати та обробляти особисті дані користувачів, такі як електронні адреси чи інші конфіденційні дані. Необхідно забезпечити, щоб ці дані були захищені від несанкціонованого доступу та використання.
- Законодавство щодо захисту даних: Проєкт повинен дотримуватися відповідного законодавства щодо захисту даних, такого як Загальний регламент з захисту даних (GDPR) в Європейському Союзі або аналогічні закони в інших країнах. Це стосується збору, зберігання та обробки особистої інформації користувачів.
- Правила конфіденційності та умови використання: Додаток повинен мати чітко визначені правила конфіденційності та умови

використання, які користувачі повинні прийняти перед використанням додатку. Це включає в себе політику щодо збору даних, використання файлів cookie, відповідальність за вміст, правила використання тощо.

У підсумку, проєктний менеджер у проєкті відіграє ключову роль у забезпеченні успішної реалізації завдань та досягненні мети. Він виступає як керівник, який не лише координує дії команди, але й відповідає за вирішення проблем, мотивацію, і забезпечення ефективного використання ресурсів. Проєктний менеджер є основною рушійною силою, яка утримує всю структуру проєкту, надаючи впевненість команді та стейкхолдерам у його успішному завершенні. Його лідерські якості та здатність до адаптації до змін створюють фундамент для ефективної співпраці та досягнення високих результатів.

3.2 Управління ресурсами

Управління ресурсами – це процес ефективного та успішного використання різних видів корпоративних ресурсів. Ці ресурси можуть включати людські ресурси, активи, засоби, обладнання та інше [57]. Управління ресурсами стає все більш актуальним в умовах зростаючої конкуренції та швидко змінюючогося бізнес-середовища.

На нашому проєкті управління ресурсами вкрай важливе. Це означає оптимальне використання людських, фінансових, технічних та інших ресурсів з метою досягнення цілей проєкту. Ефективне управління ресурсами допомагає максимізувати продуктивність команди студентів-волонтерів, ефективно використовувати обмежені ресурси, забезпечити якість результатів проєкту та мінімізувати ризики. Це також допомагає забезпечити стабільність проєкту, оскільки дозволяє планувати та контролювати використання ресурсів з урахуванням поточних потреб і майбутніх вимог проєкту.

Управління ресурсами охоплює ефективне використання як фінансові так і людських ресурсів.

3.2.1 Фінансові ресурси

Управління фінансами включає в себе бюджетування, фінансовий аналіз, контроль витрат і максимізацію прибутку. Ефективне використання фінансових ресурсів дозволяє організації вести стійкий та прибутковий бізнес.

Важливо підмітити, що даний проєкт здійснювався на благодійній основі без певного бюджету. Незважаючи на відсутність офіційного бюджету, важливо визначити потреби проєкту в фінансових ресурсах. Це включає аналіз витрат, пов'язаних з розробкою програмного забезпечення, проведенням рекламних заходів, організацією подій та інших витрат.

Наступним кроком є пошук джерел фінансування. Після оцінки потреб у фінансах необхідно шукати можливості залучення фінансової підтримки. Це може бути здійснене через пошук грантів, спонсорську допомогу від підприємств або організацій, а також участь у різних конкурсах задля отримання фінансування через призові місця.

Для забезпечення прозорості та ефективного використання фінансових ресурсів важливо вести контроль над їх витратами. Це може включати складання звітів про витрати, регулярний моніторинг фінансового стану проєкту, а також забезпечення звітності перед спонсорами або фінансовими донорами.

3.2.2 Людські ресурси

Організація роботи команди. Після визначення потреб у ресурсах необхідно організувати роботу команди студентів-волонтерів. Це включає в себе розподіл ролей та відповідальності між учасниками проєкту, планування зустрічей і комунікації, а також створення сприятливого середовища для спільної роботи.

На проєкті розподіл ролей відбувся наступним чином:

Оврамець Ілля - роль Backend розробника

Просянніков Андрій - роль Backend розробника

Соловйова Анастасія - роль дизайнера

Павлов Дмитро - роль Frontend розробника

Швець Христина - роль тестувальника

Процес розробки додатку передбачав постійні зміни у складі команди, які відбувалися відповідно до навантаження студентів. Організація роботи команди включала регулярні щотижневі зустрічі на платформі Teams. Під час цих зустрічей обговорювалися результати роботи, яку виконано за минулий тиждень, формулювалися плани на майбутній період та вирішувалися будь-які завдання або проблеми, що виникали під час розробки додатку. Також активно обговорювалися та вирішувалися будь-які стопери, що виникали в ході роботи над проєктами, з метою забезпечення безперебійного прогресу та вчасного завершення проєктів.

3.3 Управління проєктом

Управління ІТ-проєктами – це комплексний процес, який включає в себе планування, виконання, контроль та їх закриття. Для успішного управління такими проєктами необхідно дотримуватися певних принципів, зокрема [58]:

- Прозорість і підзвітність. Усі учасники команди повинні бути ознайомлені щодо статусу проєкту, його прогресу та результатів. Для цього необхідно регулярно проводити зустрічі, подавати чітко структуровану звітність та інформувати зацікавлені сторони про будь-які зміни.
- Адаптивність. ІТ-проєкти часто стикаються з непередбачуваними обставинами, як-от зміни в вимогах, бюджеті або термінах. Для успішного їх управління необхідно бути готовим до змін та адаптуватися до них.
- Оптимізація. Весь процес управління має бути спрямований на досягнення найкращих результатів з мінімальними витратами. Для цього необхідно постійно аналізувати, на якій стадії знаходиться проєкт, та впроваджувати необхідні зміни.

Центральним елементом управління ІТ-проєктом є розробка чіткого та зрозумілого плану проєкту. План включає в себе опис обсягу робіт, розподіл завдань, визначення ресурсів та витрат, оцінку ризиків та стратегію управління

ними. Планування дозволяє забезпечити структуроване виконання робіт та ефективне використання ресурсів.

Одним з важливих функцій виконання даного плану є **моніторинг ходу виконання проєкту**, а саме через відстеження прогресу проєкту через системи управління проєктами.

Постало питання який краще застосунок використовувати для моніторингу прогресу. Вибір був між 3 додатками з схожим функціоналом, а саме:

- Trello [37] - онлайн-інструмент для керування проєктами, який використовує систему дошок, списків та карток для організації завдань та спільної роботи над проєктами.
- Notion [59] - це універсальний онлайн-інструмент для організації та спільної роботи над проєктами, завданнями та нотатками. Він об'єднує в собі функції створення документів, таблиць, списків завдань, календарів, баз даних та багато іншого.
- ClickUp [60] - це інструмент для керування проєктами та задачами, який надає різноманітні функції для організації робочого процесу. Він включає в себе можливість створення завдань, розподіл їх між командою, встановлення термінів виконання та відстеження прогресу.

Так як наш проєкт заснувався на волонтерських засадах, ключовим показником була зручність та легкість в роботі і безкоштовний доступ. Нижче ви можете побачити тарифні плани кожного з них:



Рис. 13 - тарифні плани

Після аналізу усіх параметрів, було прийнято рішення використовувати додаток "Trello".

Моніторинг продуктивності команди через додаток Trello є важливим елементом успішного виконання проєктів. Це дозволяє забезпечити чіткість та структурованість завдань, контроль термінів виконання, сприяє спільній роботі та комунікації у команді, моніторингу прогресу та виявленню можливих проблем для їх подальшого вирішення. Такий підхід сприяє підвищенню ефективності та досягненню успішних результатів у роботі над проєктами.

Даний інструмент включає 5 наступних колонок:



Рис. 14 - Колонки в додатку Trello

Розберемося детальніше що означає кожна з них:

1. Backlog - в цій колонці зазвичай зберігаються всі завдання, які ще не заплановані для виконання. Це можуть бути нові ідеї або вимоги які очікують на розгляд та призначення пріоритетів.
2. Committed- в цій колонці розміщені завдання або задачі, які вже призначені конкретним членам команди та є запланованими для виконання протягом певного періоду часу.
3. In development - в цій колонці зазвичай розташовуються завдання, над якими в даний момент активно працюють. Це може означати, що вони перебувають у процесі написання коду, розробки функціоналу тощо.
4. Code review - завдання, які вже були реалізовані та готові для перегляду, перевірки коду або оцінки якості виконання. Ця колонка може включати завдання, які чекають на відгуки від інших членів команди або код-рев'ю.
5. Ready to testing - завдання, які пройшли через перевірку та готові до тестування функціональності, виправлення помилок та валідації відповідно до вимог проєкту.
6. Retrospective - ця колонка використовується для збору зворотнього зв'язку та оцінки результатів попередніх ітерацій або спроб розробки.

Вона дозволяє команді аналізувати та вдосконалювати свої процеси розробки на майбутнє.

Ось як виглядає приклад однієї з колонок, наприклад, “Committed”:

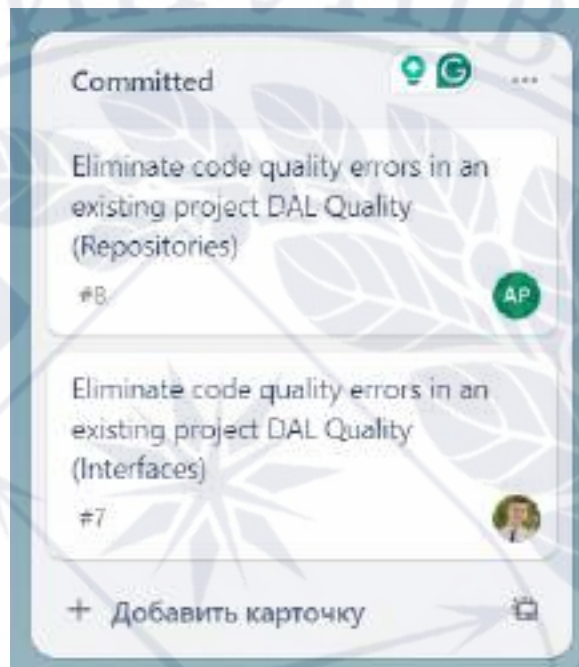


Рис. 15 - Колонка “Committed” в додатку Trello

Одним із ключових питань стало визначення платформи для спільної роботи над проектом. Після ретельного аналізу десятків варіантів, фінальний вибір було зроблено між двома лідерами: GitHub та GitLab. GitHub [61] (веб-платформа для спільної роботи над програмними проектами з використанням системи контролю версій Git) та GitLab [62] (веб-платформа для управління репозиторіями Git, яка надає широкі можливості для розробки програмного забезпечення та співпраці над проектами). Після аналізу усіх параметрів, остаточний вибір впав на GitHub, який дозволяє розробникам зберігати свої проекти в репозиторіях, відстежувати зміни в коді, співпрацювати з іншими учасниками команди, вносити внески у вигляді патчів чи заявок на злиття (pull requests), вести обговорення та відслідковувати проблеми (issues), використовувати різні інструменти для автоматизації робочих процесів та багато іншого. Забезпечення якості продукту є ключовим завданням управління ІТ-проектом. Необхідно встановлювати критерії якості,

розробити план тестування та виконати систематичний контроль якості на кожному етапі розробки.

3.4 Управління командою

Успішна реалізація проєкту залежить від ефективної організації людей, які залучені до його виконання. Ці люди можуть належати до команди проєкту, або бути підпорядковані зовнішнім структурам, таким як підрозділи або відділи організації, яка здійснює проєкт, а також інші компанії.

Під час розробки додатка по вибору наукового керівника, здійснювались наступні кроки по управління командою:

1. Розподіл завдань: перш за все завдання проєкту було розподілено поміж членів команди відповідно до їхніх навичок та досвіду. Кожен учасник знав свої обов'язки та відповідальності.
2. Створення чітких термінів: на кожній із щотижневих зустрічей було розподілено завдання та встановлено терміни їх виконання відповідно до завантаженості студентів.
3. Комунікація та співпраця: члені команди підтримували постійний зв'язок між собою, обмінюючись ідеями, відгуками та інформацією про прогрес робіт. Комунікація відбувалася як особисто, так і за допомогою електронних засобів, таких як месенджери.
4. Управління змінами: якщо з'являлися нові термінові завдання або зміни в специфікаціях проєкту, вони аналізувалися та впроваджувалися у відповідності до завантаженості розробників. Це вимагало гнучкості та готовності змінювати напрямок роботи команди.
5. Мотивація та підтримка: постійна робота над підтримкою мотивації кожного члена команди, стимулюючи їх до високої продуктивності та співпраці, шляхом надання цікавих задач для саморозвитку, самовдосконалення та набуддя досвіду для подальшого портфолію.
6. Тестування та виправлення помилок: після завершення кожного етапу розробки, додаток проходив тестування для виявлення помилок та

недоліків. Знайдені помилки виправлялися, що дозволяло забезпечити високу якість продукту.

Управління командою під час розробки додатка по вибору наукового керівника було спрямоване на забезпечення ефективної роботи команди, вчасне виконання завдань та досягнення високої якості продукту. Цей підхід дозволив досягти успіху в проєкті та задовольнити потреби користувачів.

3.5 Управління ризиками

Для початку розберемо, що означає термін ризик. Ризики – це негативні події, які можуть статися та вплинути на проєкт. Наприклад, держава випустить новий закон чи розробник тимчасово не зможе працювати над проектом.

Якщо негативна подія станеться у будь-якому разі, це не ризик, а завдання. Наприклад, проджект знає, що новому QA-інженеру потрібно вдвічі більше часу на тестування продукту. Його завдання – зважити на факт і закласти достатньо часу на цей етап.

За типом ризики можна поділити дві групи: на зовнішні та внутрішні.

Зовнішні ризики знаходяться в оточенні проєкту - постачальниках, підрядниках та клієнтах, державі, навколишньому середовищі. До них відносяться зрив строків підрядниками, стихійне лихо чи новий закон.

Внутрішні ризики – це все потенційні проблеми організації та проєктної команди. Наприклад, хтось несподівано звільниться або команда почала використовувати нову технологію та заклала недостатньо часу на її вивчення та експерименти з нею [63].

Ризик-менеджмент – один з напрямків сучасного менеджменту, що вивчає проблеми управління ризиками, що виникають в діяльності самостійної господарської організації [64]. Ефективний процес управління ризиком не може бути сукупністю фрагментарних дій, оскільки він повинен бути сформованим у комплекс дій, який є частиною загального управління бізнесом [65]. Управління ризиками – це систематичний процес ідентифікації, аналізу, оцінки, контролю та мінімізації можливих негативних впливів на досягнення цілей та результатів організації чи проєкту [66]. Цей процес є ключовим для ефективного

керівництва будь-якою діяльністю, оскільки ризики завжди присутні у будь-якій діяльності та можуть призвести до небажаних наслідків, якщо їх не врахувати.

Основна мета управління ризиками полягає у зменшенні ймовірності негативних впливів ризиків та максимізації можливостей [67]. Для досягнення цієї мети необхідно провести ряд ключових кроків:

1. Ідентифікація ризиків: Цей етап включає в себе виявлення всіх потенційних ризиків, які можуть вплинути на проект чи організацію. Це може бути виконано шляхом аналізу попередніх досвідів, проведення спеціальних аналітичних процедур та консультацій з експертами.
2. Аналіз ризиків: На цьому етапі визначається ймовірність та вплив кожного ідентифікованого ризику. Це дозволяє визначити, які ризики потребують найбільшої уваги та ресурсів для їх управління.
3. Оцінка ризиків: Під час оцінки ризиків враховуються їх потенційні наслідки для проекту чи організації та визначається оптимальна стратегія управління кожним з них. Це може включати прийняття ризику, його уникнення, зменшення чи передачу.
4. Контроль ризиків: Після того, як ризики ідентифіковані та оцінені, необхідно встановити механізми контролю, щоб переконатися, що управління ризиками виконується ефективно. Це може включати моніторинг ризиків, внесення змін у стратегії управління ризиками та вжиття відповідних заходів у разі виникнення нових ризиків або змін у вже ідентифікованих.

Управління ризиками є невід'ємною складовою успішного керівництва будь-якою діяльністю. Це дозволяє організаціям та проектам бути більш гнучкими, ефективними та стійкими до непередбачуваних змін та викликів, забезпечуючи їхній стабільний розвиток та досягнення поставлених цілей.

3.5.1 Класифікація ризиків

Розуміння різних видів ризиків допоможе забезпечити комплексний підхід до управління ними та зниження негативних наслідків для організації чи проекту. Це важливий аспект в процесі розробки стратегій та прийняття рішень, оскільки

різні типи ризиків вимагають різних підходів до їхнього управління. Наприклад, технічні ризики можуть вимагати технічних заходів та експертизи, фінансові ризики можуть вимагати фінансового планування та стратегічного управління ресурсами, а організаційні ризики можуть вимагати зміни в структурі організації чи управлінських процесах.

При розгляді різних видів ризиків, важливо також враховувати їхню взаємодію та взаємозалежність. Наприклад, один тип ризику може викликати інші, або підвищити їхню ймовірність виникнення. Такий інтегрований підхід допомагає не лише ефективно управляти окремими ризиками, а й побудувати загальну стратегію управління ризиками, яка враховує всі аспекти діяльності організації чи проєкту.

Нижче наведена частина класифікації проєктних ризиків [68]:

1. Технічні ризики:

З цією категорією ризиків ІТ індустрія знайома дуже добре. Якщо запитати команду, які вони бачать ризики на проєктах, то дана категорія буде найпопулярнішою.

Сюди відноситься те, наскільки правильно та точно визначино обсяг робіт, вимоги до проєкту, оцінки робіт, припущення та обмеження, технічні процеси, яку технологію використовують та наскільки з нею знайомі.

2. Менеджмент ризику:

Сюди відносяться ризики, пов'язані із самим процесом управління проєктом, як керують у компанії, наскільки швидко зможуть підібрати відповідних спеціалістів.

З одного боку, здається, що ці ризики слабо вплинуть на проєкт. Але якщо менеджмент у компанії залишає бажати кращого, на проєкті будуть затримки з ухваленням рішень, виділенням ресурсів на проєкт тощо. Цю категорію не можна ігнорувати.

3. Комерційні ризики

Це ризики із закупівель (якщо в рамках проекту вони є). Наприклад, не вчасно привезли якусь версію мобільного телефону або якийсь пристрій, без якого, неможливо нормально здійснити тестування.

Особливо актуально за наявності великої кількості інтеграцій із різними системами на проєкті.

Таким чином, розгляд різних видів ризиків є важливим етапом у процесі управління ризиками, що допомагає організаціям та проєктам досягати своїх цілей більш ефективно та стійко до зовнішніх впливів.

3.5.2 Приклади ризиків, що виникали під час роботи над проєктом обрання наукового керівника

У процесі розробки та впровадження проєкту по автоматизації процесу обрання наукового керівника для написання дипломних робіт, виникають численні ризики, які можуть вплинути на успішність та ефективність цього процесу. Розуміння цих ризиків є ключовим для розробки стратегій їхнього управління та мінімізації можливих негативних наслідків [69].

У цьому розділі ми детально розглянемо приклади ризиків, що виникали під час роботи над проєктом обрання наукового керівника. Кожен приклад буде розглянутий з точки зору його можливого впливу на проєкт, а також запропоновані стратегії управління та реагування на ці ризики. Це дозволить краще зрозуміти характеристики та особливості ризиків, що виникають у контексті обрання наукового керівника та розробки відповідного додатку, та виявити найбільш ефективні способи їхнього управління.

Реєстр ризиків складається з декількох основних пунктів. Ось розподіл кожного критерію з реєстру ризиків:

- Номер ризика
- Опис ризика: Це опис самого ризику, його характеристики та потенційні наслідки для проєкту.
- Ймовірність настання: Визначення ймовірності того, що ризик може статися. Вимірюється за шкалою:

1- "дуже ймовірно"

2- "ймовірно"

3 - "малоймовірно".

- Вплив ризику на результат: Оцінка того, як великий вплив має ризик на результати проекту, включаючи його обсяг, якість, бюджет тощо. Вимірюється за шкалою від 1 - "малий ризик" до 5 - "великий ризик"
- Загальний рівень ризику: Підсумкова оцінка ризику, яка враховує як ймовірність настання, так і вплив на результат. Визначається як добуток ймовірності та впливу.
- На який параметр проекту впливає: Визначення конкретного аспекту проекту, на який впливає даний ризик. Це може бути технічний аспект, фінансові ресурси, графік робіт, якість тощо.
- Наслідки: Опис потенційних наслідків випадку, якщо ризик виявиться реальним та матиме вплив на проєкт.
- Стратегія реагування: Стратегія, яка буде прийнята для управління цим ризиком у разі його виявлення.
- План дій: Конкретні кроки, які потрібно вжити для реалізації стратегії реагування та мінімізації наслідків ризику.
- Вплив: Оцінка впливу стратегії реагування та плану дій на ризик та проєкт в цілому. Вимірюється за шкалою:

Високий

Середній

Низький

Нижче наведено таблицю, де висвітлено частину реєстра ризиків:

№п/п	Опис ризику	Ймовірність настання	Вплив ризику на результат	Загальний рівень ризику	На який параметр проекту впливає	Вплив
1	Низька зацікавленість з боку студентів	3	2	6	Залучення	Середній
2	Нестача ресурсів	2	3	6	Розробка	Середній
3	Недостатньо кваліфікована	2	4	8	Розробка	Високий

	команда розробників					
4	Незлагоджена робота в команді	2	3	6	Розробка	Середній
5	Недооцінка необхідності документування коду та процесів розробки	2	4	8	Розробка	Середній
6	Зміна пріоритетів у розробників	3	2	6	Терміни	Високий
7	Непередбачувані технічні проблеми	2	3	6	Терміни	Середній
8	Технічні складнощі	2	3	6	Функціонал	Середній
9	Нечіткі вимоги до проекту	3	3	9	Функціонал	Високий
10	Зміна вимог до проекту	2	3	6	Функціонал	Середній
11	Недостатній тестувальний період перед випуском продукту	3	2	6	Функціонал	Середній
12	Складність у зборі даних	3	3	9	Якість даних	Низький
13	Обмежений доступ до інформації про викладачів	2	2	4	Якість даних	Середній
14	Конкуренція з боку інших проектів	1	3	3	Залучення	Високий
15	Зміна законодавства про освіту	2	4	8	Функціонал	Високий

Таблиця 2 - Реєстр ризиків

Також, в аналізі реєстру ризиків було розглянуто три основні аспекти: наслідки, стратегії реагування та плани дій. Усі ці параметри наведено нижче:

1. Низька зацікавленість з боку студентів:

1.1. Наслідки: Недосягнення цільової кількості користувачів

- 1.2. Стратегія реагування: Проведення інформаційної кампанії, розширення функціоналу
- 1.3. План дій: Розробка маркетингової стратегії, додавання нових можливостей
2. Нестача ресурсів:
 - 2.1. Наслідки: Затримки в розробці
 - 2.2. Стратегія реагування: Пошук додаткової робочої сили серед студентів, оптимізація ресурсів
 - 2.3. План дій: Пріоритизація задач
3. Недостатньо кваліфікована команда розробників:
 - 3.1. Наслідки: Недоліки в роботі додатку
 - 3.2. Стратегія реагування: Залучення менторів, розширення команди
 - 3.3. План дій: Пошук досвідчених фахівців, онлайн-курси
4. Незлагоджена робота в команді:
 - 4.1. Наслідки: Затримки в розробці
 - 4.2. Стратегія реагування: Застосування методологій Agile, комунікація
 - 4.3. План дій: Скрам, щоденні мітинги
5. Недооцінка необхідності документування коду та процесів розробки:
 - 5.1. Наслідки: Збільшення складності управління кодом та розробкою, можливість втрати даних та збоїв
 - 5.2. Стратегія реагування: Систематичне документування, використання кращих практик управління версіями
 - 5.3. План дій: Встановлення стандартів документування, регулярна перевірка та оновлення документації
6. Зміна пріоритетів у розробників:
 - 6.1. Наслідки: Затримка випуску проекту
 - 6.2. Стратегія реагування: Створення чіткого планування, за потреби замороження проекту
 - 6.3. План дій: Розробка календарного плану, за потреби пауза проекту
7. Непередбачувані технічні проблеми:

- 7.1. Наслідки: Затримка випуску проєкту
- 7.2. Стратегія реагування: Використання хмарних сервісів, резервне копіювання
- 7.3. План дій: Вибір надійного хостинг-провайдера, дублювання даних
8. Технічні складнощі:
 - 8.1. Наслідки: Недоліки в роботі додатку
 - 8.2. Стратегія реагування: Використання готових рішень, тестування та доопрацювання
 - 8.3. План дій: Вибір фреймворків, багаторівневе тестування
9. Нечіткі вимоги до проєкту:
 - 9.1. Наслідки: Необхідність доопрацювання
 - 9.2. Стратегія реагування: Забезпечення чіткої постановки задач, детальний опис функціоналу
 - 9.3. План дій: Розробка ТЗ, прототипування
10. Зміна вимог до проєкту:
 - 10.1. Наслідки: Необхідність доопрацювання
 - 10.2. Стратегія реагування: Забезпечення гнучкості проєкту, постійний зворотний зв'язок
 - 10.3. План дій: Модульна розробка, збір відгуків користувачів
11. Недостатній тестувальний період перед випуском продукту:
 - 11.1. Наслідки: Помилки та недоліки у функціонуванні додатку
 - 11.2. Стратегія реагування: Ретельне планування та виконання тестових процедур
 - 11.3. План дій: Розробка тестових сценаріїв, широке тестування та вирішення проблем до випуску продукту
12. Складність у зборі даних:
 - 12.1. Наслідки: Недостовірні дані
 - 12.2. Стратегія реагування: Створення зручних форм для заповнення, верифікація даних
 - 12.3. План дій: Розробка шаблонів, співпраця з адміністрацією

13. Обмежений доступ до інформації про викладачів:
 - 13.1. Наслідки: Обмежена корисна функціональність
 - 13.2. Стратегія реагування: Пошук додаткових джерел інформації, можлива співпраця з університетом, кафедрою та деканатом
 - 13.3. План дій: Встановлення партнерств з іншими установами чи застосування альтернативних методів збору інформації
14. Конкуренція з боку інших проектів:
 - 14.1. Наслідки: Низька зацікавленість користувачів
 - 14.2. Стратегія реагування: Розширення функціоналу, акцент на унікальних можливостях
 - 14.3. План дій: Додавання нових модулів, проведення маркетингових кампаній
15. Зміна законодавства про освіту:
 - 15.1. Наслідки: Необхідність доопрацювання проекту
 - 15.2. Стратегія реагування: Моніторинг змін у законодавстві, гнучкість у розробці
 - 15.3. План дій: Створення юридичної консультативної групи зі студентів, внесення змін до проекту

У процесі роботи над проектом обрання наукового керівника виникало безліч ризиків, які вимагали пристосування та ефективного управління з метою забезпечення успішного завершення проекту. Зокрема, серед ризиків виявлено такі, як недооцінка ресурсів, технічні складнощі, зміни вимог та інші.

Застосування ефективних стратегій управління ризиками є ключовим фактором успішності проекту. Наприклад, відповідне планування, розробка альтернативних сценаріїв, регулярний моніторинг та аналіз ризиків дозволяють забезпечити реагування на негативні події та мінімізувати їх вплив на проєкт.

Отже, управління ризиками вимагає системного підходу та уважного аналізу ситуації, але при цьому дозволяє забезпечити стабільність та успішне завершення проекту.

ВИСНОВКИ

У даній бакалаврській роботі було детально проаналізовано наявний процес обрання наукового керівника виконано глибокий аналіз, який дозволив виявити ключові проблеми та можливості для подальшого вдосконалення процесу. На основі отриманих результатів було розроблено концепцію додатку, спрямованого на покращення процесу вибору наукового керівника. Цей додаток враховує потреби користувачів і забезпечує більш зручний та ефективний спосіб взаємодії зі структурами, відповідальними за надання наукового керівництва.

Під час аналізу перспектив реалізації додатку був проведений SWOT-аналіз, що дозволив ідентифікувати сильні та слабкі сторони, а також можливості та загрози, пов'язані з проектом. Це дало можливість розробити стратегію подальшого розвитку додатку та забезпечити його успішну імплементацію. Крім того, були розроблені user story та реєстри ризиків, що дозволило зрозуміти потреби користувачів та потенційні загрози для проекту та підготувати план дій для їх врахування.

Отже, на основі проведеного дослідження та розроблених інструментів можна зробити висновок про важливість та актуальність розробки додатку для покращення процесу обрання наукового керівника, а також про необхідність подальшої роботи над його реалізацією та впровадженням.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. А.О. Радзіховська, Ю.С. Антонов. Роль практичних навичок у сфері ІТ для проєктного менеджера на прикладі розробки створення клієнтсерверних програм. Вінниця, 2022.
2. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Sixth Edition (2017p., 756ст.)
3. Л. М. Добровська, О. В. Аверьянова, УПРАВЛІННЯ ІТ-ПРОЕКТАМИ В MICROSOFT PROJECT, КПІ ім. Ігоря Сікорського, Київ, 2020, 152.
4. Поняття та класифікація проєктів - Бібліотека BukLib.net. Головна - Бібліотека BukLib.net. URL: <https://buklib.net/books/22265/> (дата звернення: 14.04.2024).
5. Етапи життєвого циклу розробки ПЗ. Веб-студія розробки програмного забезпечення - IC Studio, Україна. URL: <https://icstudio.online/post/etapi-zhittyevogo-ciklu-rozrobki-pz> (дата звернення: 14.04.2024).
6. Что такое SDLC? – Описание жизненного цикла разработки программного обеспечения – AWS. Amazon Web Services, Inc. URL: <https://aws.amazon.com/ru/what-is/sdlc/> (дата звернення: 14.04.2024).
7. LibreTexts. 10.2: Модель життєвого циклу розробки систем (SDLC). LibreTexts - Ukrayinska. URL: <http://surl.li/sopvw> (дата звернення: 14.04.2024).
8. Что такое SDLC. Главная. URL: <https://nataliasnitko.com/sdlc.html> (дата звернення: 14.04.2024).
9. Махум Z. Життєвий цикл розробки програмного забезпечення (Software Development Life Cycle - SDLC). Махум Zosym. URL: <https://www.maxzosim.com/software-development-life-cycle-sdlc/> (дата звернення: 14.04.2024).
10. С. О. Цибульник К. С. Барандич. ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЧАСТИНА 1. ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ : підручник. Київ: КПІ ім. Ігоря Сікорського, 2022. 270 с.

11. Що таке методологія Scrum і коли варто її використовувати. IT Jobs at SoftServe – Start tech career now. URL: <https://career.softserveinc.com/uk-ua/stories/what-is-scrum-methodology> (дата звернення: 14.04.2024).
12. Ken Schwaber and Jeff Sutherland. The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game. 2020.
13. Що таке Scrum (Скрам)? Один з найпопулярніших запитів 2024 року. Brain Rain. URL: <https://brainrain.com.ua/скрам-это/> (дата звернення: 14.04.2024).
14. SCRUM: що це таке, основи та переваги скрам методу – Wezom. IT-компанія повного циклу розробки програмного забезпечення WEZOM - Київ, Україна. URL: <https://wezom.com.ua/ua/blog/metod-scrum> (дата звернення: 14.04.2024).
15. Хто такий Бізнес Аналітик в IT - Навички та Обов'язки | Академія ШАГ. IT курси в Києві | Навчання IT спеціальностям | IT STEP. URL: <https://kiev.itstep.org/blog/business-analyst-in-it-all-about-the-profession> (дата звернення: 14.04.2024).
16. Інструменти професії: з чим працює бізнес-аналітик в IT. Happy Monday. URL: <https://happymonday.ua/z-chym-pratsyuye-biznes-analityk-v-it> (дата звернення: 14.04.2024).
17. Google Sheets: Sign-in. g.co. URL: <https://g.co/kgs/SRNHWko> (date of access: 28.04.2024).
18. Google Slides: Sign-in. g.co. URL: <https://g.co/kgs/CDVME3k> (date of access: 28.04.2024).
19. Google Документи: вхід. Sign in - Google Accounts. URL: <https://docs.google.com/document/create?hl=uk> (дата звернення: 28.04.2024).
20. Google Диск. <https://www.google.com/>. URL: https://www.google.com/intl/ru_tj/drive/ (дата звернення: 28.04.2024).
21. Slack is your productivity platform. Slack. URL: <https://slack.com/> (date of access: 28.04.2024).
22. Microsoft Teams. URL: <https://www.microsoft.com/uk-ua/microsoft-teams/log-in> (date of access: 28.04.2024).

23. Одна платформа для общения | Zoom. Zoom. URL: <https://zoom.us/ru> (дата звернення: 28.04.2024).
24. Balsamiq: Fast, focused wireframing for teams and individuals | Balsamiq. Balsamiq: Fast, focused wireframing for teams and individuals | Balsamiq. URL: <https://balsamiq.com/> (date of access: 28.04.2024).
25. Figma: The Collaborative Interface Design Tool. Figma. URL: <https://www.figma.com/> (date of access: 28.04.2024).
26. Axure RP - UX Prototypes, Specifications, and Diagrams in One Tool. Axure. URL: <https://www.axure.com/> (date of access: 28.04.2024).
27. draw.io. draw.io. URL: <http://Diagrams.net> (date of access: 28.04.2024).
28. Ideal Modeling & Diagramming Tool for Agile Team Collaboration. Ideal Modeling & Diagramming Tool for Agile Team Collaboration. URL: <https://www.visual-paradigm.com/> (date of access: 28.04.2024).
29. Intelligent Diagramming | Lucidchart. Lucidchart. URL: <https://www.lucidchart.com/pages/> (date of access: 28.04.2024).
30. Low-Code Platform for Process Automation and Enterprise Apps | Bizagi. Low-Code Platform for Process Automation and Enterprise Apps | Bizagi. URL: <https://www.bizagi.com/> (date of access: 28.04.2024).
31. The Universal Process Orchestrator | Camunda. Camunda. URL: <https://camunda.com/> (date of access: 28.04.2024).
32. Project Manager в IT: обов'язки, переваги та шлях до кар'єрного успіху – GoIT Global. GoIT Global. URL: <https://goit.global/ua/articles/project-manager-v-it/> (дата звернення: 14.04.2024).
33. Project Manager: хто такий PM, чим він займається, де навчатись професії та як знайти роботу - Wizeclub Education. Wizeclub Education. URL: <https://wizeclub.education/blog/project-manager-hto-takij-pm-chim-vin-zajmayetsya-de-navchatis-profesiyi-ta-yak-znajti-robotu/> (дата звернення: 14.04.2024).
34. Основи управління IT проектами: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки»/ КПП ім. Ігоря Сікорського ; уклад.: В. О. Кузьмініх, Р. А. Тараненко. Київ: КПП ім. Ігоря Сікорського, 2019. – 75 с.

35. Jira | Atlassian. Collaboration software for software, IT and business teams. URL: <https://www.atlassian.com/ru/software/jira> (date of access: 28.04.2024).
36. Manage your team's work, projects, & tasks online • Asana • Asana. Asana. URL: <https://asana.com/ru> (date of access: 28.04.2024).
37. Керуйте проектами своєї команди звідусіль | Trello. Manage Your Team's Projects From Anywhere | Trello. URL: <https://trello.com/uk> (дата звернення: 28.04.2024).
38. Microsoft Project-Management URL: <https://www.microsoft.com/uk-ua/microsoft-365/project/project-management-software> (date of access: 28.04.2024).
39. dropbox.com. Dropbox. URL: https://www.dropbox.com/uk_UA/ (date of access: 28.04.2024).
40. Sharepoint. URL: <https://www.microsoft.com/uk-ua/microsoft-365/sharepoint/collaboration> (date of access: 28.04.2024).
41. Microsoft Excel. URL: <https://www.microsoft.com/uk-ua/microsoft-365/excel> (date of access: 28.04.2024).
42. RescueTime: Fully Automated Time Tracking Software. RescueTime: Fully Automated Time Tracking Software. URL: <https://www.rescuetime.com/> (date of access: 28.04.2024).
43. Toggl. Toggl Track: Time Tracking Software for Any Workflow. Toggl Track: Time Tracking Software for Any Workflow. URL: <https://toggl.com/> (date of access: 28.04.2024).
44. Clockify - FREE Time Tracking Software. Clockify. URL: <https://clockify.me/> (date of access: 28.04.2024).
45. Skype | Спілкуйтеся в безкоштовних відеовикликах, хоч де ви будете. Skype | Stay connected with free video calls worldwide. URL: <https://www.skype.com/uk/> (дата звернення: 28.04.2024).
46. Free screen recorder for Mac and PC | Loom. Loom. URL: <https://www.loom.com/> (date of access: 28.04.2024).

47. Canva. URL: https://www.canva.com/uk_ua/
48. Read Aloud: A Text-to-Speech Voice Reader. Read Aloud: A Text-to-Speech Voice Reader. URL: <https://readaloud.app/> (date of access: 28.04.2024).
49. Код професії: чим займається бізнес-аналітик і як ним стати. URL: <https://dou.ua/lenta/articles/become-business-analyst/> (date of access: 05.05.2024).
50. Debra Paul James Cadle Donald Yeates. Business Analysis. 3rd ed. BCS Learning & Development Limited, 2014. 308 p.
51. Шлях до інновацій: Роль аналітики у стратегічному розвитку бізнесу. URL: https://sitniks.ua/blog_post/rol-analitiky-u-strategichnomu-rozvitku-biznesu/ (date of access: 05.05.2024).
52. Epic vs Story: Understanding The Differences With Example?. Invensis Learning Blog. URL: <https://www.invensislearning.com/blog/epic-vs-story/> (date of access: 05.05.2024).
53. SWOT-аналіз: що це таке та приклади використання. URL: <https://wedex.com.ua/blog/swot-analiz-shho-tse-take-ta-prikladi-vikoristannya/> (date of access: 28.04.2024).
54. Чому SWOT-аналіз необхідний практично кожному бізнесу?. eSputnik. URL: <https://esputnik.com/uk/blog/swot-analiz-iz-prikladami> (дата звернення: 28.04.2024).
55. Хто такий project-менеджер і як ним стати | Онлайн-курси від компанії QATestLab. Онлайн-курси від компанії QATestLab | Головна сторінка. URL: <https://training.qatestlab.com/blog/helpful-materials/who-is-a-project-manager-and-how-to-become-one/> (дата звернення: 14.04.2024).
56. Як мотивувати команду, щоб вона приносила результат - Wizeclub Education. Wizeclub Education. URL: <https://wizeclub.education/blog/yak-motivuvati-komandu-shhob-vona-prinosila-rezultat/> (дата звернення: 14.04.2024).
57. Управління ресурсами: визначення, значення та планування. BusinessYield. URL: <https://businessyield.com/uk/management/resource-management/> (дата звернення: 21.04.2024).

58. А.О. Радзіховська, Зелінська О. В. ОСНОВНІ ПРИНЦИПИ ТА ЕТАПИ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ. Вінниця, 2023.
59. Your connected workspace for wiki, docs & projects | Notion. Notion. URL: <https://www.notion.so/> (date of access: 21.05.2024).
60. ClickUp. ClickUpâ€¢ | One app to replace them all. URL: <http://surl.li/tvbri> (date of access: 21.05.2024).
61. GitHub: Let's build from here. GitHub. URL: <https://github.com/> (date of access: 21.05.2024).
62. The most-comprehensive AI-powered DevSecOps platform. The most-comprehensive AI-powered DevSecOps platform | GitLab. URL: <https://about.gitlab.com/> (date of access: 21.05.2024).
63. Skillsetter. Ризики проекту: аналіз, оцінка та стратегії управління. Skillsetter. URL: <https://skillsetter.io/blog/risk-management-ua> (дата звернення: 14.04.2024).
64. М. В. Боровик. РИЗИК-МЕНЕДЖМЕНТ. Харків : ХНУМГ ім. О. М. Бекет., 2018. 67 с.
65. Литюга Юлія Володимирівна. ОРГАНІЗАЦІЯ РИЗИК-МЕНЕДЖМЕНТУ НА ПІДПРИЄМСТВІ, Київський національний економічний університет імені Вадима Гетьмана., 2017.
66. Міністерство фінансів України. Методичний посібник щодо аспектів управління ризиками, як складової системи внутрішнього контролю у розпорядника бюджетних коштів. Київ, 2022. 22 с.
67. Вербіцька І.І к.е.н доцент кафедри фінансів і. банківської справи Чортківський ін-т підприємництва та бізнесу Тернопільський нац. економічний ун-т. РИЗИК-МЕНЕДЖМЕНТ ЯК СУЧАСНА СИСТЕМА УПРАВЛІННЯ РИЗИКАМИ ПІДПРИЄМНИЦЬКИХ СТРУКТУР. СТАЛИЙ РОЗВИТОК ЕКОНОМІКИ, 2013, Міжнародний науково-виробничий журнал.
68. Які бувають ризики на проєктах?. Е5. URL: <https://e5.ua/uk/blogpost-2/yaki-buvayut-riziki-na-proyektah/> (дата звернення: 14.04.2024).

69. А.О. Радзіховська, Антонов Ю. С. РИЗИК МЕНЕДЖМЕНТ ПРИ РОЗРОБЦІ ДОДАТКУ ДЛЯ ВИБОРУ НАУКОВОГО КЕРІВНИКА. Вінниця, 2024.



User story 1

USER PORTFOLIO

View Interests

URL: /scinterests/{PersonId}

Method: GET

Expected result: повертає коло інтересів для вказаної персони

Add Interests

URL: /scinterests/{PersonId}

Method: POST

Expected result: додає коло інтетересів для заданої персони. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

Update Interests

URL: /scinterests/{PersonId}

Method: PUT

Expected result: оновлює коло інтетересів для заданої персони. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

View person working project experience

URL: /personprojects/{PersonId}

Method: GET

Expected result: повертає проектний досвід вказаної персони.

Contract: дата початку, дата кінця, назва проекту, роль.

View working project experience details

URL: /personproject/{ProjectId}

Method: GET

Expected result: повертає проектний досвід вказаною персони коло інтетересів для заданої персони.

Contract: повний опис проекту.

Add working project experience

URL: /personproject/{PersonId}

Method: POST

Expected result: додає до вказаної персони, новий проектний досвід. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

Update working project experience

URL: /personproject/{ProjectId}

Method: PUT

Expected result: оновлює вже існуючий проектний досвід. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

Delete working project experience

URL: /personproject/{ProjectId}

Method: DELETE

Expected result: видаляє вже існуючий проектний досвід персони. У подальшому буде доступним лише для авторизованих користувачів (ключ). У разі відсутності прав для виконання дії або інших подібних помилок (400, 401, 402, 403) користувачу все одно показуємо лише 404

View external profile types

URL: /externalprofiletypes/

Method: GET

Expected result: Повертає тип зовнішнього профілю (Google Академія, Scopus, Оркід тощо)

Add external profile type

URL: /externalprofiletype/

Method: POST

Expected result: Додає тип зовнішнього профілю ь

Contract:

Update external profile type

URL: /externalprofiletype/{Id}

Method: PUT

Expected result: Оновлює тип зовнішнього профілю

Contract:

Delete external profile type

URL: /externalprofiletype/{Id}

Method: DELETE

Expected result: Видаляє тип зовнішнього профілю

Contract:

View external profiles

URL: /externalprofiles/{PersonId}

Method: GET

Expected result: Повертає вже створені профілі особи (Скільки підв'язаних під ID)

Add external profile

URL: /externalprofile/{PersonId}

Method: POST

Expected result: Додає новий зовнішній профіль особи

Contract:

Update external profile

URL: /externalprofile/{Id}

Method: PUT

Expected result: Оновлює вже існуючий зовнішній профіль особи

Delete external profile

URL: /externalprofile/{Id}

Method: DELETE

Expected result: Видаляє вже існуючий зовнішній профіль особи

User story 2

View department types

URL: /departmenttypes/

Method: GET

Expected result: Повертає назву(тип) факультету (Кафедра, лабораторія, навчальний відділ)

View faculties

URL: /faculties/

Method: GET

Expected result: Повертає перелік факультетів (ФІПТ, Економічний, Юридичний і тд)

View departments

URL: /departments/{ParentDepartmentId}

Method: GET

Expected result: Повертає перелік кафедр (Прикладної математики, Математики, Інформаційних технологій)

View specialties

URL: /specialties/{Id}

Method: GET

Expected result: Повертає перелік спеціальностей (122, 125, 111, 113)

View student's groups

URL: /studentgroups/{SpecId}

Method: GET

Expected result: Повертає перелік студентських груп (Б20_122д, Б20_125д і тд)

View list of students for group

URL: /studentslist/{GroupId}

Method: GET

Expected result: Повертає список студентів групи

Contract: фото, ПІБ, електронна адреса

View list of employee for the department

URL: /employeeelist/{DepartmentId}

Method: GET

Expected result: Повертає список викладачів що працюють на кафедрах
(Антонов Ю.С, Січко Т.В)

View scientific director tasks

URL: /scientificdirectortasks/{PersonId|DepartmentId}

Method: GET

Expected result: Повертає завдання наукового керівника (показує певні рецензії)

View scientific director task details

URL: /scientificdirectortask/{TaskId}

Method: GET

Expected result: Показує деталі завдання яке надано науковим керівником
(термін, основні положення і тд)

Add scientific director task type

URL: /scientificdirectortask/{TaskId}

Method: POST

Expected result: Додає певний тип завдання, яке було надано науковим керівником

Update scientific director task

URL: /scientificdirectortask/{TaskId}

Method: PUT

Expected result: Оновлює завдання наукового керівника (підтягується оновлення інформація)

Delete scientific director task

URL: /scientificdirectortask/{TaskId}

Method: DELETE

Expected result: Видаляється завдання (або певні фрагменти завдання) наукового керівника

View scientific director employee pool

URL: /scientificdirectoremployeeepool/{TaskId}

Method: GET

Expected result: Показує прив'язку ід викладача до кількості студентів (task ід)

Add scientific director employee pool

URL: /scientificdirectoremployeeepool/{TaskId}

Method: POST

Expected result: Додає прив'язку ід викладача до кількості студентів (task ід)

Update scientific director employee pool

URL: /scientificdirectoremployeeepool/{TaskId}

Method: PUT

Expected result: Оновлює прив'язку ід викладача до кількості студентів (task ід)

Delete scientific director employee pool

URL: /scientificdirectoremployee/ {TaskId}

Method: DELETE

Expected result: Видаляє прив'язку ід викладача до кількості студентів (task ід)

View scientific director students pool

URL: /scientificdirectorstudentspool/ {TaskId}

Method: GET

Expected result: Показує перелік студентів, який має викладач

Add scientific director students pool

URL: /externalprofiletype/ {Id}

Method: POST

Expected result: Додає студентів до переліку, який має викладач

Update scientific director students pool

URL: /externalprofiletype/ {Id}

Method: PUT

Expected result: Оновлює перелік студентів, який має викладач

Delete scientific director students pool

URL: /externalprofiletype/ {Id}

Method: DELETE

Expected result: Видаляє певного студента з переліку викладачів

Add departments type

URL: /scientificdirectortask/ {TaskId}

Method: POST

Expected result: Додає назву(тип) факультету (Кафедра, лабораторія, навчальний відділ)

Update departments type

URL: /scientificdirectortask/{TaskId}

Method: PUT

Expected result: Оновлює назву(тип) факультету (Кафедра, лабораторія, навчальний відділ)

Delete departments type

URL: /scientificdirectortask/{TaskId}

Method: DELETE

Expected result: Видаляє назву(тип) факультету (Кафедра, лабораторія, навчальний відділ)

Опитування щодо вибору наукового керівника

Опитування для студентів

anketaonline.fakholonkiv.org.ua/anketa/2022

[Самостійний доступ опитування](#)

Обов'язковий вибір

Ваш статус на момент опитування *

☐ студент бакалавріату
☐ студент магістратури
☐ отримав диплом бакалавра/магістра/спеціаліста і зараз не навчається
☐ Інше: _____

Вкажіть назву навчального закладу, в якому ви навчаєтесь/навчалися *

Мій вибір: _____

Яким чином ви отримали свого наукового керівника? *

☐ Ще не обрав
☐ З особистого знайомства
☐ За рекомендацією викладачів/друзів/знайомих
☐ На основі інформації з сайту університету
☐ Інше: _____

Наскільки прозорим на ваш погляд є процес обрання наукового керівника у вашому університеті?

0 1 2 3 4 5 6 7 8 9 10

Визначте, прозора чи непрозора процедура

З якими труднощами ви стикалися при виборі наукового керівника? *

☐ Невідомість наукових інтересів/досліджень викладачів
☐ Непрозрачність критеріїв вибору
☐ Відсутність доступної інформації про викладачів
☐ Інше: _____

Які фактори для вас найважливіші при виборі наукового керівника? *

☐ Науковий інтерес/дослідження
☐ Довід роботи/стаж
☐ Вміння налагодити комунікацію та роботу
☐ Інше: _____

Які фактори для вас найважливіші при виборі наукового керівника? *

☐ Наукові інтереси/дослідження

☐ Довід роботи/стаж

☐ Вміння налагодити комунікацію та роботу

☐ Друге:

Чи вважаєте ви, що запропонована система допомогла б вам прийняти більш обґрунтоване рішення щодо вибору керівника? *

☐ Так

☐ Ні

Чи допомогла б вам наявність розширеного портфоліо викладачів у процедурі вибору керівника? *

☐ Так

☐ Ні

З якими проблемами зустрілися під час написання роботи (курсової, бакалаврської, магістрської, дипломної)? *

Моя стаття:

Чи хотіли ви використувати додаток для вибору наукового керівника? *

☐ Так

☐ Ні

Які функції ви б хотіли бачити в системі вибору наукового керівника? *

Моя стаття:

[Опитування](#)

[Очистити форум](#)

ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загальноновизнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;
що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)_____
(підпис)