

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Економічний факультет
Кафедра економіко-математичного моделювання та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістра

на тему: **«УПРАВЛІННЯ**
ПРОЄКТАМИ РОЗРОБКИ ТА ПІДТРИМКИ ТЕЛЕГРАМ-БОТІВ»

Виконав: студент 2 курсу, групи МУП-21
другого(магістерського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Управління проєктами»
Пасічник Євген Сергійович

Керівник: старший викладач Клебан Юрій Вікторович

Рецензент: кандидат технічних наук, доцент
кафедр прикладної математики та кібербезпеки
Донецького національного університету імені Василя Стуса
Загоруйко Любов Василівна

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ

Завідувач кафедри економіко-математичного моделювання та інформаційних
технологій _____ (проф., д.е.н. Кривицька О.Р.)
Протокол № ____ від « ____ » _____ 2024 р.

Острог, 2024

Міністерство освіти і науки України
Національний університет «Острозька академія»

Факультет: економічний

Кафедра: економіко-математичного моделювання та інформаційних технологій

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма: Управління проектами

ЗАТВЕРДЖУЮ

Завідувач кафедри

Ольга КРИВИЦЬКА

« ___ » _____ 20__ р.

ЗАВДАННЯ
на кваліфікаційну роботу студента

Пасічник Євген Сергійович

(прізвище, ім'я, по батькові)

1. *Тема роботи* «Управління проектами розробки та підтримки телеграм-ботів»

керівник роботи старший викладач Клебан Юрій Вікторович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджено наказом ректора НаУОА від "03" листопада 2023 року № 98

2. *Термін здачі студентом закінченої роботи:* 05.12.2024

3. *Вихідні дані до роботи:* дані про наявні освітні продукти, методології управління проектами, дані для розробки телеграм-бота.

4. *Перелік завдань, які належить виконати:*

1. Дослідити методології управління проектами для розробки Telegram-ботів.
2. Оцінити важливість розробки, орієнтованої на користувача, для поліпшення взаємодії з ботами.
3. Розглянути стратегії інтеграції Telegram-ботів із зовнішніми сервісами.
4. Визначити цілі та результати проекту Telegram-бота на етапі ініціації.
5. Провести аналіз вимог і пріоритетів функцій Telegram-бота.
6. Розробити дизайн та архітектуру для масштабованого Telegram-бота.
7. Реалізувати функціональність Telegram-бота, забезпечуючи модульність і обслуговуваність.
8. Спланувати та провести тестування Telegram-бота.
9. Провести розгортання Telegram-бота та інформувати зацікавлені сторони.

10. Впровадити систему підтримки для вирішення запитів і виправлення помилок.
11. Забезпечити аудит безпеки та конфіденційності даних.
12. Сприяти ефективній командній співпраці та обміну знаннями.

5. Перелік графічного матеріалу: рисунки, діаграми.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Клебан Ю. В., старший викладач	01.12.23	01.12.23
2	Клебан Ю. В., старший викладач	01.12.23	01.12.23
3	Клебан Ю. В., старший викладач	01.12.23	01.12.23

7. Дата видачі завдання: 01.12.23

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника.	жовтень 2023	+
2.	Початок розробки телеграм-бота.	13 березня 2023	+
3.	Аналіз вимог і постановка задач.	березня 2023	+
4.	Розробка технічного завдання і планування ресурсів.	квітень 2023	+
5.	Реалізація основних функціональних можливостей телеграм-бота.	травень 2023	+
6.	Тестування і відладка телеграм-бота.	червень - серпень 2023	+
7.	Завершення створення телеграм-бота.	8 серпня 2024	+
8.	Документування і створення кваліфікаційної роботию.	вересень 2024	+
9.	Створення презентація і попередній захист кваліфікаційної роботи.	листопад 2024	+
10.	Публічний захист кваліфікаційної роботи перед екзаменаційною комісією.	грудень 2024	+

Студент: _____ (підпис) _____ Пасічник Є. С. (прізвище та ініціали)

Керівник кваліфікаційної роботи: _____ (підпис) _____ Клебан Ю. В. (прізвище та ініціали)

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня магістра

Тема: «Управління проектами розробки та підтримки телеграм-ботів»

Автор: *Пасічник Євген Сергійович*

Науковий керівник: *старший викладач Клебан Юрій Вікторович*

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: 70 с., 17 рис., 2 табл.,
2 додатки, 35 джерел.

Ключові слова: УПРАВЛІННЯ ПРОЄКТАМИ, ТЕЛЕГРАМ-БОТ, БОТ, КОРИСТУВАЧ, ТЕСТУВАННЯ, РОЗРОБКА, АРХІТЕКТУРА.

Короткий зміст праці:

У цій праці досліджуються основні аспекти розробки та підтримки телеграм-ботів, зокрема процеси управління проектами, стратегічне планування, тестування, забезпечення якості та ефективне управління ботами. Основна увага приділяється специфіці управління проектами в контексті телеграм-ботів, що включає розгляд етапів їхнього життєвого циклу та управління ризиками. Окремо розглядаються питання аналізу вимог користувачів та функціональних вимог до телеграм-ботів, а також роль тестування, автоматизованих засобів тестування та контролю якості у забезпеченні належної роботи ботів. Метою цього дослідження є аналіз ефективного управління проектами розробки та підтримки Telegram-ботів. Об'єктом дослідження є процес управління проектами, пов'язаний з розробкою та впровадженням ботів, а предметом — методи та підходи до оптимізації цього процесу. Основні завдання включають дослідження методологій управління проектами, оцінку важливості користувацько-орієнтованої розробки, стратегії інтеграції з зовнішніми сервісами, та розробку дизайну та архітектури для масштабованих ботів.

ANNOTATION
qualification work
to obtain a master's degree

Topic: «Project management for the development and support of Telegram bots»

Author: *Yevhen Pasichnyk*

Academic supervisor: *senior lecturer Yuriy Kleban.*

Protected by «.....»..... 20__.

Explanatory note to the qualification work: 70 p., 17 figures, 2 tables, 2 appendices, 35 sources.

Keywords: PROJECT MANAGEMENT, TELEGRAM BOT, BOT, USER, TESTING, DEVELOPMENT, ARCHITECTURE.

Summary of the work:

This paper explores the main aspects of developing and maintaining Telegram bots, including project management processes, strategic planning, testing, quality assurance, and effective bot management. The main focus is on the specifics of project management in the context of Telegram bots, including the stages of their life cycle and risk management. Separate consideration is given to the analysis of user requirements and functional requirements for Telegram bots, as well as the role of testing, automated testing tools, and quality control in ensuring the proper operation of bots. The purpose of this study is to analyse the effective management of Telegram bot development and support projects. The object of the study is the project management process related to the development and implementation of bots, and the subject is the methods and approaches to optimise this process. The main tasks include researching project management methodologies, assessing the importance of user-centric development, integration strategies with external services, and developing design and architecture for scalable bots.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРОЦЕСУ КЕРУВАННЯ ПРОЄКТАМИ В РОЗРОБЦІ ТА ПІДТРИМЦІ ТЕЛЕГРАМ-БОТІВ	6
1.1. Визначення проєкту розробки та підтримки телеграм-ботів.....	6
1.2. Етапи життєвого циклу проєкту телеграм-бота.	7
1.3. Особливості управління проєктом в контексті розробки телеграм-ботів	21
РОЗДІЛ 2. СТРАТЕГІЧНЕ ПЛАНУВАННЯ ТА АНАЛІЗ ВИМОГ ДО ТЕЛЕГРАМ-БОТІВ.....	32
2.1. Стратегічне планування в розробці телеграм-ботів.....	32
2.2. Аналіз вимог користувачів та функціональних вимог до телеграм-ботів	33
2.3. Управління ризиками у розробці та експлуатації телеграм-ботів.....	36
РОЗДІЛ 3. МЕТОДИКИ ТЕСТУВАННЯ ТА ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ТЕЛЕГРАМ-БОТІВ.....	44
3.1. Види тестування та їх роль у розробці телеграм-ботів.....	44
3.2. Засоби автоматизованого тестування для телеграм-ботів.....	51
3.3. Забезпечення якості та процеси контролю якості в життєвому циклі телеграм-бота.....	53
РОЗДІЛ 4. ЕФЕКТИВНЕ УПРАВЛІННЯ ТА ПІДТРИМКА ТЕЛЕГРАМ-БОТІВ	55
4.1. Системи моніторингу та аналізу даних в управлінні телеграм-ботами..	55
4.2. Планування та впровадження оновлень та нового функціоналу телеграм-ботів.....	56

4.3. Управління взаємодією з користувачами та обробка звернень	65
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71
ДОДАТКИ.....	75
Додаток А	75
Додаток Б.....	76

ВСТУП

Telegram став провідною платформою для обміну повідомленнями, де важливу роль відіграють боти. Вони виконують різноманітні завдання, покращуючи користувацький досвід завдяки автоматизації. Telegram-боти дозволяють інтерактивно взаємодіяти через повідомлення та команди, розширюючи функціональність платформи та інтегруючи зовнішні сервіси. Їхні можливості варіюються від прийому платежів до створення ігор через HTTPS-запити. Приклади популярних ботів, таких як ImageBot, WikiBot, GmailBot, MusicBot, GitHubBot і YouTubeBot, демонструють різноманітність їхнього застосування для різних потреб користувачів. [1]

Телеграм боти мають значний потенціал у різних сферах, що підтверджується дослідженнями вчених та компаній. За словами П'єр-Іва Оллера, чат-боти можуть значно покращити користувацький досвід завдяки своїй здатності забезпечувати швидку та ефективну взаємодію з користувачами. Дослідження Microsoft показали, що боти можуть автоматизувати рутинні завдання, дозволяючи бізнесу зосередитися на більш стратегічних аспектах діяльності. Університет Гарварда вивчає потенціал ботів у сфері освіти, підкреслюючи їхню роль у персоналізованому навчанні. Компанія IBM використовує свої технології Watson для створення ботів, які допомагають у медичній діагностиці та підтримці клієнтів.

Мета дослідження управління проектами в контексті розробки Telegram-ботів є багатогранною і спрямована на розгляд ключових аспектів ефективного використання цих автоматизованих агентів в екосистемі Telegram.

Об'єктом дослідження є процес управління проектами, пов'язаний з розробкою та впровадженням Telegram-ботів.

Предметом дослідження є методи та підходи до ефективного управління проектами, які сприяють оптимізації розробки, інтеграції та використання Telegram-ботів для вирішення різних бізнес-завдань.

Основні завдання полягають у наступному:

13. Дослідити методології управління проєктами для розробки Telegram-ботів.
14. Оцінити важливість розробки, орієнтованої на користувача, для поліпшення взаємодії з ботами.
15. Розглянути стратегії інтеграції Telegram-ботів із зовнішніми сервісами.
16. Визначити цілі та результати проєкту Telegram-бота на етапі ініціації.
17. Провести аналіз вимог і пріоритетів функцій Telegram-бота.
18. Розробити дизайн та архітектуру для масштабованого Telegram-бота.
19. Реалізувати функціональність Telegram-бота, забезпечуючи модульність і обслуговуваність.
20. Спланувати та провести тестування Telegram-бота.
21. Провести розгортання Telegram-бота та інформувати зацікавлені сторони.
22. Впровадити систему підтримки для вирішення запитів і виправлення помилок.
23. Забезпечити аудит безпеки та конфіденційності даних.
24. Сприяти ефективній командній співпраці та обміну знаннями.

На кожному етапі буде використано наступні методи дослідження:

- Емпіричні методи, а саме експериментальний метод для вирішення конкретних проблем, таких як впровадження нових технологій, розробка методів локалізації та тестування Telegram-ботів, спостереження і опис для аналізу конкурентів та вивчення ефективності і слабких місць різних ботів.
- Теоретичні методи, а саме аналіз для вивчення технічних, функціональних та управлінських аспектів розробки Telegram-ботів, пояснення та узагальнення для розкриття деталей та загальних тенденції в управлінні проєктами, пояснюючи висновки дослідження.
- Системний метод для аналізу взаємодії різних компонентів управління проєктами та їх впливу на результати проєкту.
- Функціональний метод для детально розгляду функціональних аспектів Telegram-ботів та їх відповідність вимогам користувачів.

Ці методи взаємодіятимуть між собою, створюючи комплексний підхід до дослідження управління проєктами у сфері розробки та підтримки Telegram-ботів. [2, 3, 4, 5, 6].

Отже, вивчення та застосування різноманітних емпіричних та теоретичних методів дослідження у проєктному управлінні Telegram-ботами дозволяє систематично вирішувати завдання в розробці та підтримці. Застосування експериментального методу дозволяє ефективно впроваджувати нові технології, аналізувати роботу конкурентів та тестувати різні аспекти ботів. Теоретичні методи, такі як аналіз, дедукція та узагальнення, роблять можливим глибоке вивчення усіх аспектів управління проєктами, включаючи функціональні та системні складові. Використання системного та функціонального методів допомагає структурувати роботу над ботами та розглядати їхні функціональні аспекти. Загалом, комплексне застосування цих методів сприяє ефективному та системному управлінню проєктами у сфері Telegram-ботів, забезпечуючи успішний розвиток та підтримку цих інноваційних рішень.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРОЦЕСУ КЕРУВАННЯ ПРОЄКТАМИ В РОЗРОБЦІ ТА ПІДТРИМЦІ ТЕЛЕГРАМ-БОТІВ

1.1. Визначення проєкту розробки та підтримки телеграм-ботів

Це перший етап, на якому чітко формулюються основні аспекти та мета проєкту. На цьому етапі визначається обсяг робіт, визначаються конкретні завдання та цілі проєкту, а також визначається термін його виконання. Ключові етапи включають у себе розробку концепції телеграм-бота, його функціональних вимог, технічного завдання та планування процесів розробки та підтримки. Визначення проєкту є важливим етапом, що визначає вектор руху і створює базовий фундамент для подальших етапів управління проєктом.

Технології в галузі інформаційних технологій (ІТ) часто супроводжуються великою кількістю аббревіатур та технічних термінів, що може спричиняти плутанину. Порівнюючи сучасну ситуацію в ІТ з ранніми етапами розвитку таких галузей, як електрика та автомобілебудування, де нові технології також були позначені складними термінами для створення ілюзії унікальності, навіть якщо для споживачів ці питання були складними для розуміння. Така тенденція до надмірної ускладненості із застосуванням аббревіатур залишається актуальною і в наш час.

Основною проблемою в ІТ є не технології як такі, а розрив між керівництвом бізнесу та технічними рішеннями, які підтримують ці процеси. Для вирішення цієї проблеми необхідно впровадити так звані «спільні мови», які можуть бути зрозумілі як для бізнес-лідерів, так і для технічних фахівців. Це можна порівняти з використанням креслень в інженерії, де стандартні моделі допомагають різним зацікавленим сторонам розуміти та впроваджувати ідеї.

У контексті розробки та підтримки телеграм-ботів важливо застосовувати моделі, які допоможуть організувати бізнес-процеси та технологічні рішення. Це дозволяє досягти більшої гнучкості організації, роблячи її більш ефективною та швидкою в адаптації до змін. Використання таких підходів, як архітектура, орієнтована на сервіси (SOA), управління бізнес-процесами (BPM) та управління на

основі моделей (MBM), дозволяє налагодити ефективну взаємодію між бізнесом і технологіями, що сприяє успішній реалізації проєктів, таких як розробка та підтримка телеграм-ботів [29].

1.2. Етапи життєвого циклу проєкту телеграм-бота.

Життєвий цикл проєкту телеграм-бота включає в себе сукупність етапів, які необхідно пройти від ідеї до реалізації та подальшої підтримки. Кожен етап має важливе значення для досягнення кінцевої мети — створення ефективного, стабільного та зручного інструмента для користувачів. Розробка телеграм-бота є складним і багатогранним процесом, що потребує чіткої організації та виконання низки ключових завдань на різних етапах.

Розглянемо основні етапи життєвого циклу проєкту телеграм-бота, починаючи від планування та проєктування до впровадження, тестування і подальшої підтримки. Зрозуміле та ретельно сплановане проходження кожного етапу гарантує, що проєкт буде успішно реалізований у задані терміни та з належною якістю. Крім того, на кожному етапі важливо взаємодіяти з усіма зацікавленими сторонами, забезпечуючи правильне розуміння вимог та потреб, що в свою чергу сприяє безперебійній роботі телеграм-бота після його запуску.

Успішне виконання проєкту залежить від грамотного поєднання технологічних та бізнес-процесів, тому на кожному етапі важливо не лише вирішувати технічні питання, а й враховувати потреби користувачів і бізнес-цілі. Визначення чітких етапів життєвого циклу дозволяє досягти оптимального результату і уникнути можливих помилок на етапах розробки та експлуатації.

Етап ініціації проєкту є першим і одним із найважливіших етапів життєвого циклу розробки телеграм-бота. На цьому етапі визначаються основні напрямки та цілі проєкту, а також створюється фундамент для подальшої роботи. Ініціація проєкту включає в себе формулювання завдань, оцінку обсягу робіт, визначення ресурсів і строків, що необхідні для реалізації ідеї. Від цього етапу залежить, чи зможе проєкт успішно пройти всі наступні етапи розробки та підтримки.

Першим кроком є чітке формулювання мети проєкту. Це включає в себе розуміння того, для чого саме розробляється телеграм-бот, які функції він має виконувати, які проблеми користувачів він має вирішити і яку цінність принесе бізнесу або кінцевим користувачам. Важливою частиною ініціації є визначення вимог до бота, зокрема функціональних та нефункціональних. Функціональні вимоги можуть включати перелік завдань, які бот має виконувати (наприклад, обробка запитів, автоматизація комунікацій тощо), тоді як нефункціональні вимоги стосуються таких аспектів, як швидкість роботи, безпека, стабільність і масштабованість [11, 12].

Паралельно з визначенням вимог проводиться оцінка ресурсів, необхідних для реалізації проєкту. Це включає як людські ресурси (розробники, аналітики, тестувальники), так і технологічні ресурси (програмне забезпечення, сервери, інструменти для інтеграції з іншими системами). Оцінка витрат і складання бюджету також є важливою складовою цього етапу. Чітке планування бюджетних витрат дозволяє уникнути проблем на етапах розробки, коли може виникнути потреба в додаткових фінансових або матеріальних ресурсах.

Ще одним важливим аспектом етапу ініціації є визначення строків виконання проєкту (Таблиця 1.1). Під час цього етапу формуються попередні оцінки щодо тривалості виконання різних етапів розробки, що дозволяє розробити детальний графік проєкту. Також на цьому етапі відбувається формування команди проєкту та розподіл обов'язків серед учасників. [7, 8]

Таблиця 1.1

Основні етапи проєкту розробки та підтримки телеграм-ботів

Етап	Опис етапу	Тривалість	Відповідальні особи
Визначення мети проєкту	Формулювання основних цілей та задач.	1 тиждень	Менеджер проєкту, аналітик
Розробка концепції бота	Опис основних функціональних можливостей.	2 тижні	Розробники, аналітик
Технічне завдання	Визначення технічних вимог до розробки.	1 тиждень	Технічний керівник, розробники
Планування ресурсів	Оцінка необхідних ресурсів для реалізації.	1 тиждень	Менеджер проєкту
Розробка та тестування	Створення коду бота та його тестування.	4 тижні	Розробники, тестувальники
Підтримка та оновлення	Регулярна підтримка та додавання нових функцій.	Постійно	Технічна підтримка

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

З точки зору управління проєктом, підходи до ініціації можуть бути різними. Один з найпоширеніших підходів — це водоспадна модель, при якій на кожному етапі проєкту чітко визначаються завдання та послідовність їх виконання. Інший підхід — це гнучка методологія (Agile), де акцент робиться на поступовому розвитку проєкту з можливістю внесення змін і адаптації в процесі (Рис. 1.1). Вибір підходу залежить від специфіки проєкту, вимог до гнучкості та можливості адаптації до змін у вимогах клієнта або ринку [30].

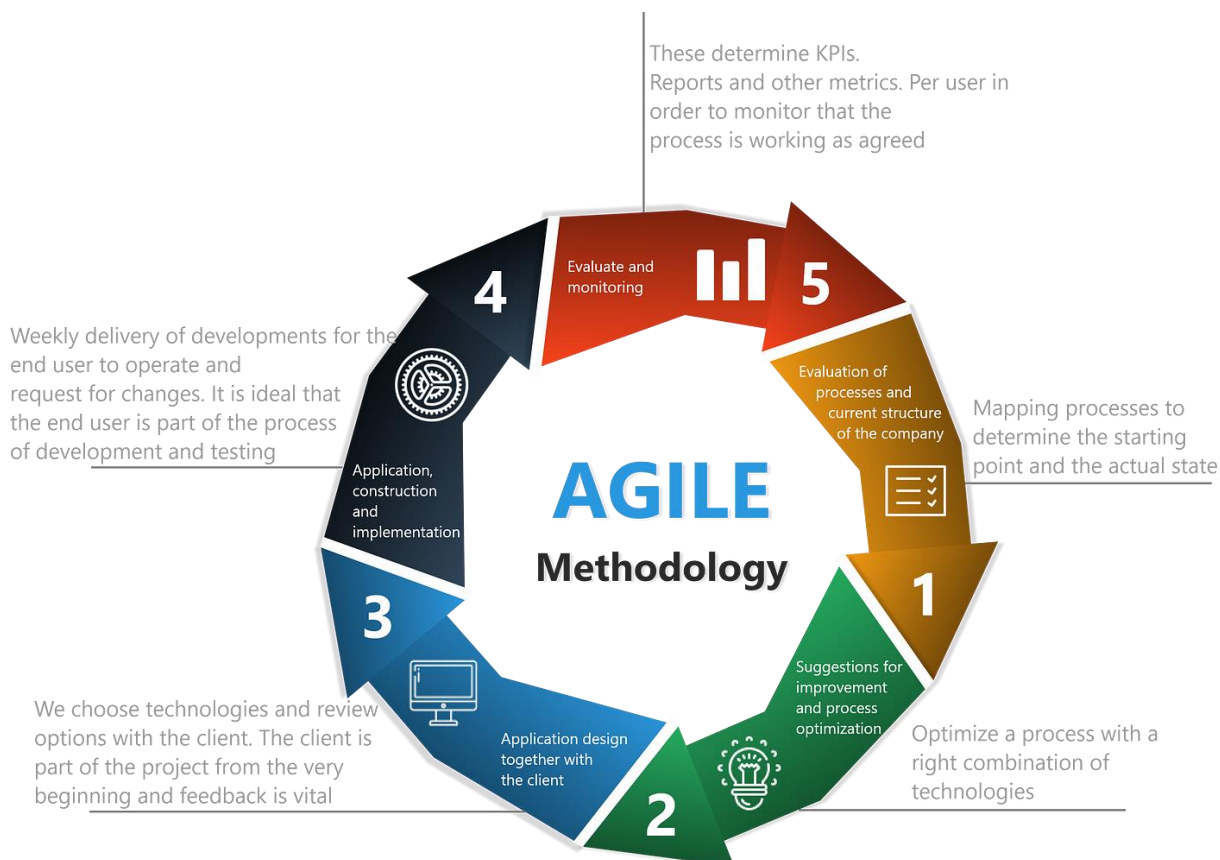


Рис. 1.1. Схема методології Agile

Джерело: [30]

Під час ініціації важливим є також визначення всіх зацікавлених сторін, таких як замовники, користувачі, розробники та інші учасники проєкту. Визначення їх ролей і очікувань дозволяє забезпечити ефективну комунікацію протягом усіх етапів проєкту і мінімізувати ризики, пов'язані з непорозуміннями або зміною вимог.

Етапи життєвого циклу проєкту телеграм-бота можна описати на прикладі проєкту "Вартовий" — бота для проведення навчань по евакуації. Цей проєкт є чудовим прикладом, оскільки він ілюструє весь процес розробки та реалізації від початкових ідей до підтримки бота після запуску. Розглянемо, як кожен етап життєвого циклу проєкту виглядає на прикладі "Вартового".

Ініціація проєкту.

На цьому етапі ставляться основні цілі та завдання для бота. Визначається, що "Вартовий" буде призначений для навчання користувачів основам евакуації під час надзвичайних ситуацій, таких як повітряні тривоги та інші. Встановлюються

функціональні вимоги до бота, наприклад, наявність різних сценаріїв евакуації, тестів для перевірки знань, а також можливість отримання сповіщень у разі тривоги. Важливими є також оцінка необхідних ресурсів, формулювання бюджету та термінів реалізації проєкту.

Терміни для реалізації проєкту становили 4 місяці від 13 березня по 8 серпня. Це нам дало зрозуміти, що ми можемо зробити, а на що в нас не вистачить часу. Таким чином сформувавши список пріоритетних задач (Рис. 1.2).

Матриця Ейзенхауера



Рис. 1.2. Матриця Ейзенхауера для встановлення пріоритетності задач

Джерело: [31]

Розробник бекенд частини відповідає за створення серверної архітектури додатка, реалізуючи взаємодію між базами даних, серверами та іншими компонентами. Його завдання включали проєктування та розробку API, налаштування серверних процесів, а також розгортання додатків на хмарних платформах, таких як Azure (Рис. 1.7), для забезпечення масштабованості та

стабільності. Крім того, він здійснював інтеграцію клієнтських запитів із серверною частиною, займаючись усіма технічними аспектами зв'язку між клієнтом і сервером [25, 26].

Розробник фронтенд частини займався створенням інтерфейсу веб-додатку (Рис. 1.5), що дозволило користувачам взаємодіяти з сервісом. Водночас він виконував роль керівника проєкту, координуючи роботу всієї команди та контролюючи терміни, ресурси і виконання задач. Він слідкував за тим, щоб розробка йшла за планом, забезпечуючи взаємодію між розробниками, а також міг вирішувати проблеми на етапах тестування та запуску продукту.

Розробник телеграм-бота спеціалізувався на створенні ботів для платформи Telegram (Рис. 1.6), програмуючи їх для виконання конкретних функцій або автоматизації певних процесів. Він працював з API Telegram, реалізував логіку роботи бота та інтеграцію з іншими системами, щоб забезпечити користувачам комфортну та ефективну взаємодію з ботом.

Планування проєкту.

На етапі планування створюється детальний план проєкту, що включає в себе розподіл задач, визначення ключових етапів та створення технічного завдання для розробників (Рис. 1.3). Планується час і ресурси, необхідні для розробки бота, включаючи програмістів, тестувальників, а також програмне забезпечення та серверні потужності. Крім того, на цьому етапі визначаються ризики проєкту, наприклад, можливі труднощі при інтеграції з іншими системами або нестабільність роботи при високих навантаженнях.

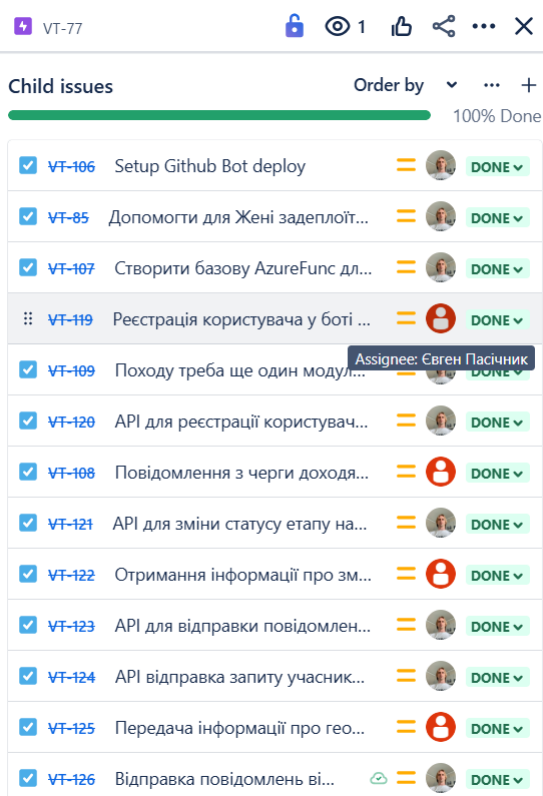


Рис. 1.3. Список розподілених задач у Jira

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

На рисунку 1.4 зображено архітектуру проєкту, що базується на хмарній платформі Azure та технології .NET Core для розробки веб-додатків.

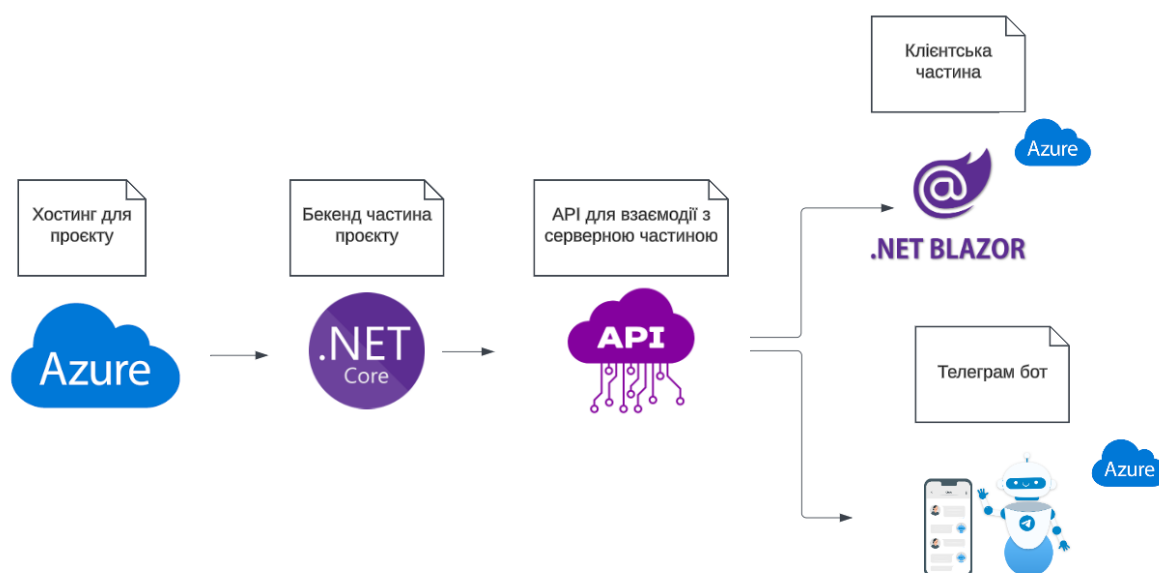


Рис. 1.4. Архітектура проєкту на платформі .NET Core та Azure

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

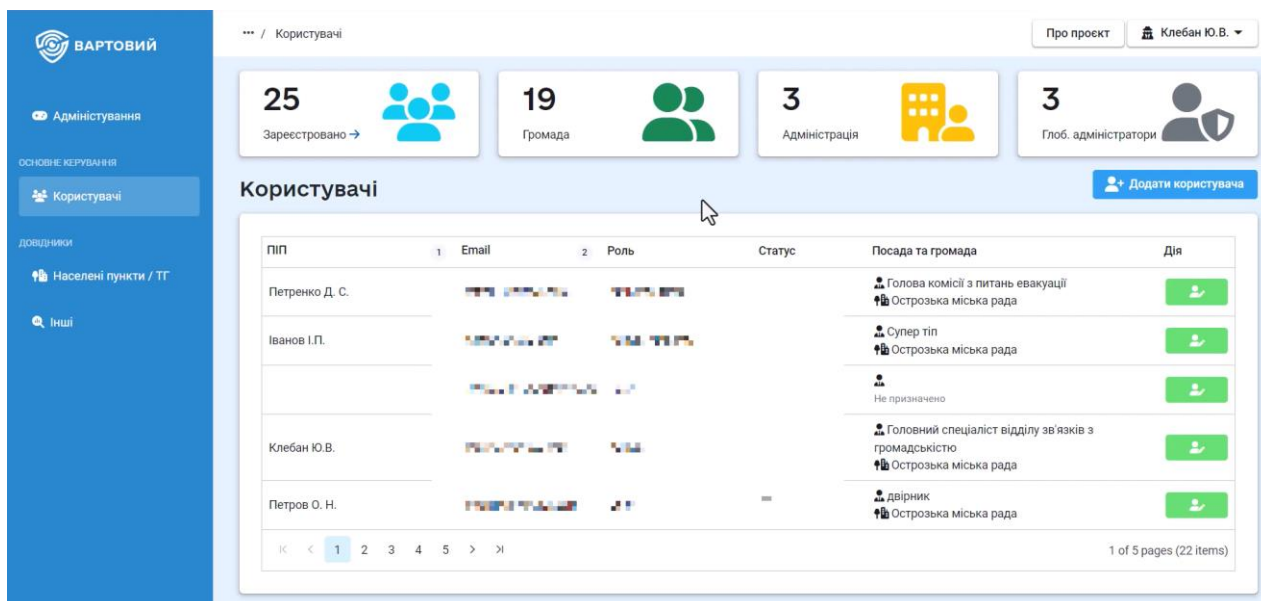


Рис. 1.5 Клієнтська частина веб-додатку «Вартовий»

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

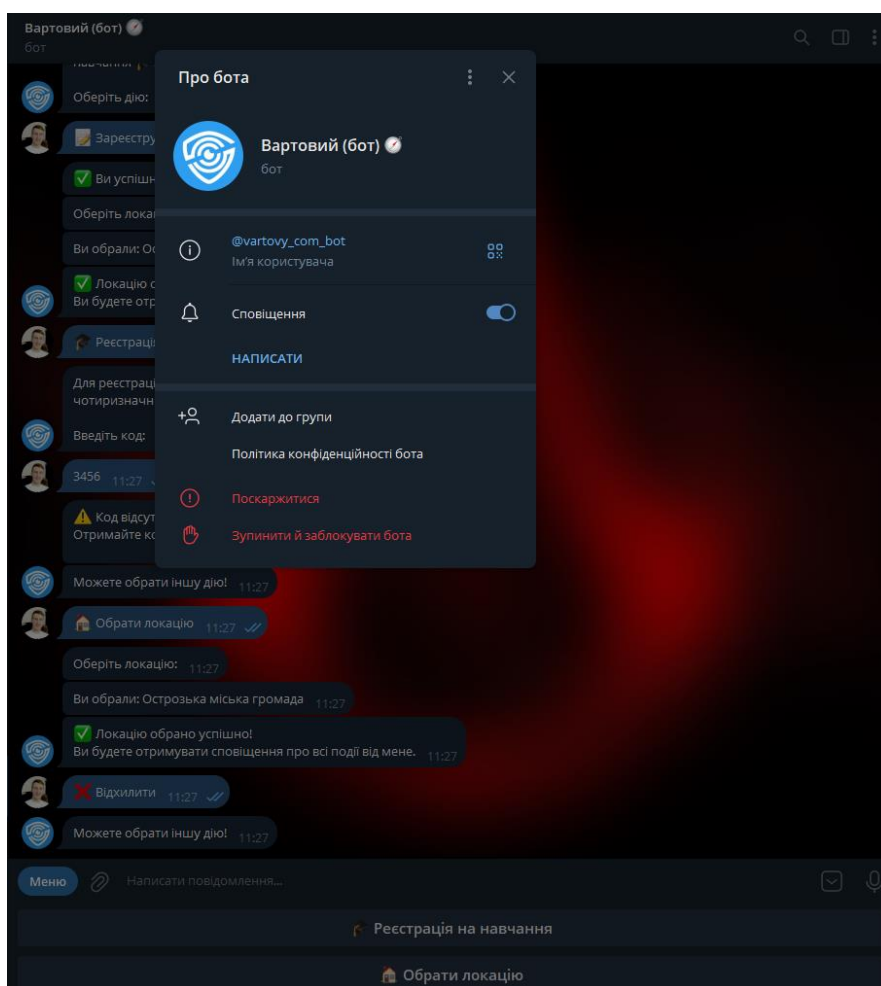


Рис. 1.6 Телеграм-бот для проєкту «Вартовий»

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

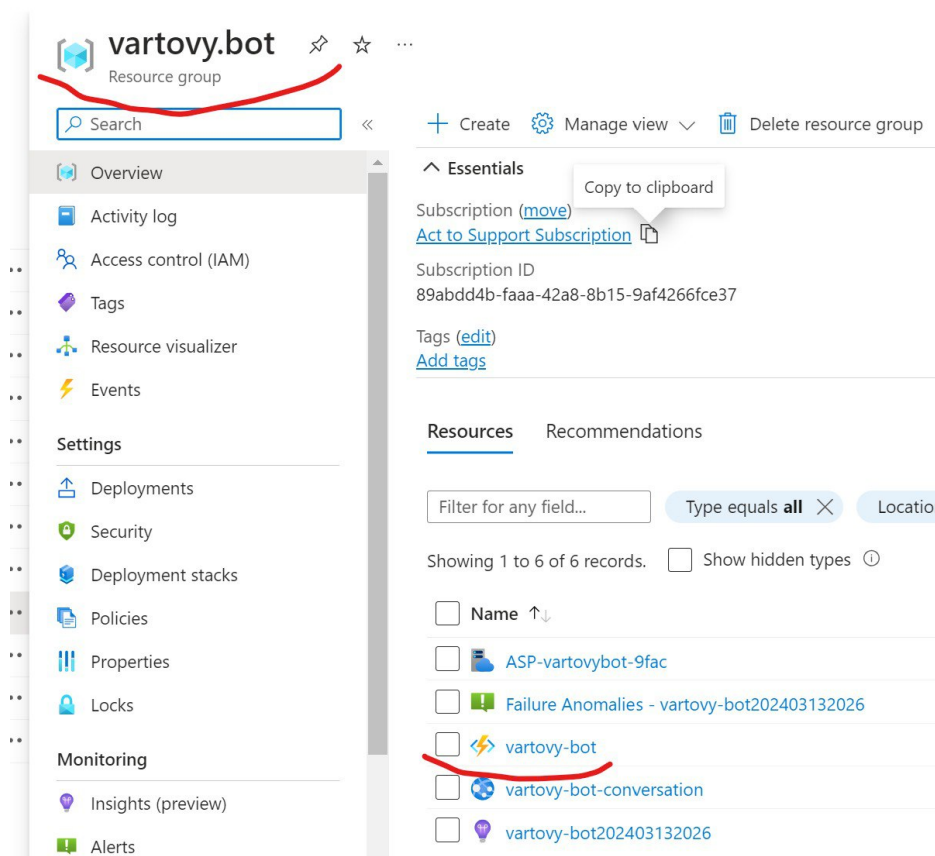


Рис. 1.7 Розміщення бота на платформі Azure

Джерело: розроблено автором на основі розробки проекту "Вартовий" у 2024 р.

Опис основних компонентів і технологій:

Хостинг для проекту (Azure) - це основна хмарна платформа, на якій розміщується весь проект. Вона забезпечує інфраструктуру для розгортання як серверної частини проекту, так і клієнтської частини. Azure надає можливість масштабування, високої доступності та безпеки, що робить її ідеальним вибором для хмарних рішень [27].

Бекенд частина проекту (.NET Core) - це серверна частина проекту, реалізована за допомогою .NET Core, крос-платформенного фреймворку для розробки веб-додатків. .NET Core забезпечує високу продуктивність, безпеку і гнучкість. Це основний компонент, який обробляє запити від клієнтської частини і відповідає за логіку бізнес-процесів.

API для взаємодії з серверною частиною - тут реалізований інтерфейс для комунікації між серверною частиною (на базі .NET Core) і клієнтськими

компонентами. API дозволяє виконувати різноманітні операції, такі як отримання даних, обробка запитів, а також забезпечує зв'язок між сервером та іншими компонентами, такими як Telegram-бот і Blazor.

Клієнтська частина (.NET Blazor) - клієнтська частина проєкту розроблена за допомогою .NET Blazor. Blazor дозволяє створювати інтерактивні веб-додатки з використанням C# замість JavaScript. Це забезпечує єдину платформу для серверної та клієнтської частини, полегшуючи розробку і підтримку коду. Клієнтська частина взаємодіє з сервером через API для отримання необхідних даних.

Telegram-бот - взаємодіє з користувачами через Telegram, виконує функції для обробки запитів користувачів. Він підключається до серверної частини через API, що дозволяє користувачам взаємодіяти з системою, наприклад, отримувати повідомлення чи оновлення в реальному часі.

Архітектура телеграм-боту складається з декількох основних модулів і файлів, які організовані для забезпечення функціональності бота. Нижче наведено короткий опис структури проєкту та його компонентів:

1. api

Цей модуль відповідає за роботу з API сервісами та містить декілька підмодулів:

- Axios: обробляє HTTP запити.
 - axiosClient.ts: конфігурує та ініціалізує axios клієнт.
 - index.ts: експорт всіх необхідних функцій.
- identity: працює з сервісами ідентифікації.
 - identityServices.ts: сервіси для аутентифікації та авторизації.
 - index.ts: експорт сервісів ідентифікації.
- locality: робота з локалізацією.
 - getLocalityConfirmed.ts: отримання підтверджених даних про локацію.
 - index.ts: експорт функцій локації.
- sendData: відповідає за відправлення даних.

- `sendDataForTrainingParticipant.ts`: відправка даних учасника тренінгу.
- `actionCommands`: обробка команд.
- `index.ts`: експорт всіх функцій.
- `sendImage.ts`: відправка зображень.

2. bot

Модуль, що реалізує логіку роботи бота.

- `actionCommands`: команди дій для бота.
 - `index.ts`, `sendImage.ts`, `setActionCommands.ts`: налаштування команд.
- `commands`: команди, які розпізнає бот.
 - `commands.ts`: визначення команд.
 - `index.ts`: експорт команд.
- `conversations`: сценарії діалогів.
 - `createAccount.ts`: створення облікового запису.
 - `getLocation.ts`: отримання місцезнаходження.
 - `locationAppLogin.ts`: авторизація в додатку за місцезнаходженням.
 - `registrationForTraining.ts`: реєстрація на тренінг.
 - `index.ts`: експорт всіх сценаріїв.
- `types`: визначення типів.
 - `index.ts`: експорт типів.
 - `bot.ts`: основний файл конфігурації бота.
 - `index.ts`: головний файл для запуску бота.
 - `keyboardCommand.ts`: обробка команд з клавіатури.
 - `limitsConfig.ts`: конфігурація лімітів.

3. constants

Містить константи, які використовуються в проєкті.

- `index.ts`: експорт всіх констант.
- `openAi.ts`: константи для роботи з OpenAI.

4. server

Файли, пов'язані з серверною частиною проєкту.

5. controllers

Контролери для обробки запитів.

`bot-controller.ts`: основний контролер для роботи з ботом.

6. middleware

Проміжні програмні засоби для обробки запитів.

- `upload.js`: обробка завантаження файлів.

7. routes

Маршрутизація запитів.

- `bot-routes.ts`: визначення маршрутів для бота.

8. types

Визначення типів, які використовуються в проєкті.

- `index.ts`: експорт типів.

Інші файли:

- `.env-example`: приклад конфігураційного файлу середовища.
- `.gitignore`: список файлів і директорій, які ігноруються Git.
- `.prettierrc.json`: конфігурація Prettier.
- `README.md`: документація проєкту.
- `index.ts`: головний файл для запуску проєкту.
- `package-lock.json` та `package.json`: залежності проєкту.
- `tsconfig.json`: конфігурація TypeScript.
- `validateEnvs.ts`: валідація змінних середовища.

Ця архітектура забезпечує модульність та гнучкість, дозволяючи легко масштабувати та підтримувати бота.

Важливою частиною розробки Telegram-бота є розробка якісного технічного завдання (ТЗ) вимагає врахування багатьох аспектів. Правильно складене ТЗ забезпечує зрозуміле бачення проєкту як для замовника, так і для розробника [13, 14, 15, 16].

Спочатку необхідно чітко визначити мету та завдання бота. Це дозволить окреслити загальну концепцію проєкту і зрозуміти, які функції має виконувати бот. Наприклад, якщо бот призначений для підтримки користувачів, потрібно вказати, які саме види підтримки він має надавати (інформаційні, технічні тощо).

Далі слід детально описати цільову аудиторію. Визначення, хто буде користуватися ботом, допоможе адаптувати його функціонал і інтерфейс під конкретні потреби користувачів. Варто також врахувати можливі сценарії використання, що дасть змогу передбачити всі варіанти взаємодії з ботом.

Далі потрібно перейти до функціональних вимог. Тут необхідно описати всі можливості бота. Кожна функція повинна бути детально прописана, включаючи можливі відповіді бота на певні команди користувачів, алгоритми обробки запитів, а також інтеграцію з іншими сервісами чи базами даних. Наприклад, якщо бот буде відповідати на запити щодо залишків товарів на складі, варто вказати, як і звідки він буде отримувати цю інформацію. Для спрощення сприйняття того який має бути функціонал доречно буде використати UML-діаграми (Додатки А, Б).

Інтерфейс користувача також заслуговує уваги. Необхідно описати, як буде виглядати взаємодія з ботом: які кнопки чи меню будуть використовуватися, які повідомлення бот буде надсилати у відповідь на команди користувачів. Цей розділ ТЗ дозволить розробникам зрозуміти, як зробити взаємодію з ботом інтуїтивно зрозумілою та зручною.

Потрібно також розглянути питання безпеки. Врахування аспектів захисту даних користувачів та безпеки комунікацій між ботом та серверами є критично важливими. У ТЗ варто включити вимоги щодо автентифікації та авторизації користувачів, захисту від можливих атак та методи шифрування даних [17].

Необхідно визначити вимоги до продуктивності та масштабованості. Вказати, скільки одночасних користувачів бот повинен обслуговувати без втрати продуктивності, та як буде забезпечуватися його робота при збільшенні навантаження.

Потрібно ще дещо додати стосовно планування ресурсів для створення телеграм-бота. Ми вже знаємо що, воно включає такі аспекти як людські, фінансові та часові ресурси. Для цього необхідно чітко визначити всі етапи проєкту, розподілити завдання між членами команди та ефективно використовувати інструменти управління проєктами, такі як Jira (Рис. 1.3).

1. Планування людських ресурсів.

Важливо визначити ролі та обов'язки кожного члена команди. Це може включати розробників, тестувальників, дизайнерів та менеджерів проєкту. Кожен учасник команди повинен чітко розуміти свої завдання та терміни їх виконання. Відповідно до компетенцій і досвіду, завдання можуть бути розподілені таким чином, щоб максимізувати продуктивність та забезпечити високу якість кінцевого продукту [18].

2. Планування фінансових ресурсів.

Оцінка бюджету проєкту є критично важливою. Потрібно врахувати витрати на оплату праці, ліцензії на програмне забезпечення, серверні ресурси та інші можливі витрати. Ретельне планування фінансових ресурсів допоможе уникнути неочікуваних витрат і забезпечить фінансову стабільність проєкту.

3. Планування часових ресурсів.

Визначення чітких дедлайнів для кожного етапу проєкту є основою успішного планування. Важливо встановити реалістичні терміни виконання завдань, враховуючи можливі затримки та ризики. Використання методології Agile може бути корисним для гнучкого управління часом і адаптації до змін [19].

4. Створення графіка робіт та визначення ключових етапів.

Після планування ресурсів необхідно створити детальний графік робіт. Починається все з визначення основних етапів розробки телеграм-бота, таких як аналіз вимог, дизайн, розробка, тестування та запуск. Кожен з цих етапів можна розбити на більш дрібні завдання, що дозволить краще контролювати процес виконання проєкту [20].

Визначення ключових етапів включає:

1. Аналіз вимог та планування.
2. Розробка технічного завдання та дизайн системи.
3. Програмування основних функцій та інтеграція з телеграм API.
4. Тестування та виправлення помилок.
5. Запуск бота та технічна підтримка.
6. Використання Jira

Jira є потужним інструментом для управління проєктами, який дозволяє ефективно планувати, відслідковувати прогрес і контролювати виконання завдань. У Jira можна створювати задачі для кожного члена команди, встановлювати пріоритети, дедлайни і відстежувати статус виконання. За допомогою дашбордів і звітів можна отримати повну картину про хід проєкту, що сприяє своєчасному виявленню проблем і їх вирішенню.

1.3. Особливості управління проєктом в контексті розробки телеграм-ботів

Особливості управління проєктом в контексті розробки телеграм-ботів визначаються специфічними вимогами та характеристиками цієї області. До ключових аспектів можна віднести:

Динамічність змін в управлінні проєктом особливо актуальна в контексті розробки телеграм-ботів. Швидка зміна технологій та вимог користувачів створює значні виклики, що вимагають від розробників високого рівня гнучкості та здатності адаптуватися до нових тенденцій [21].

Постійне оновлення технологій означає, що команди розробників повинні бути готові інтегрувати нові функціональні можливості та адаптуватися до змін у платформі Telegram. Наприклад, нещодавнє впровадження підтримки Payments API в Telegram (Рис. 1.5) створило можливості для розробників інтегрувати платіжні системи безпосередньо в боти. Це потребувало від розробників швидкої реакції та адаптації їхніх проєктів до нових можливостей, щоб залишатися конкурентоспроможними на ринку [22].

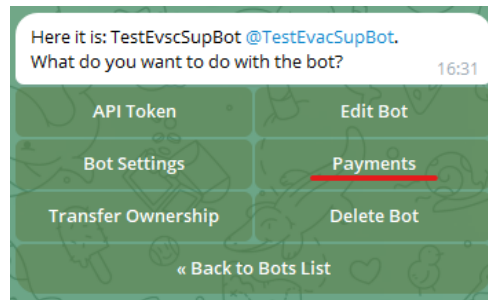


Рис. 1.5. Додання оплати для телеграм-бота

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

Вимоги користувачів також можуть змінюватися швидко, що вимагає від команди розробників постійного аналізу зворотного зв'язку та оперативного внесення змін. Наприклад, користувачі можуть вимагати нових функцій, таких як підтримка мультимедійних повідомлень або інтеграція з іншими популярними сервісами. В таких випадках команда має бути готова швидко реалізувати нові функції, не порушуючи загальної стабільності системи.

Крім того, важливо зазначити, що успішна адаптація до змін вимагає ефективного управління проєктом. Це включає регулярні зустрічі команди для обговорення поточного стану проєкту, пріоритизацію задач та швидку реакцію на будь-які проблеми, що можуть виникнути. Гнучкість також передбачає використання сучасних методологій розробки, таких як Agile або Scrum, що дозволяють ефективно планувати і здійснювати роботу в умовах постійних змін.

Наприклад, якщо команда працює над ботом для підтримки клієнтів, і з'являється нова технологія розпізнавання голосу, яка може значно покращити користувацький досвід, управління проєктом повинно швидко оцінити можливості її впровадження, перепланувати задачі та реалізувати інтеграцію з мінімальними затримками.

Інтеграція з зовнішніми сервісам.

Розробка телеграм-ботів часто включає інтеграцію з різноманітними зовнішніми сервісами, що потребує докладного планування та управління інтерфейсами.

Потрібно зважати на інтеграцію з зовнішніми сервісами при розробці телеграм-ботів, адже вони часто потребують взаємодії з іншими платформами для виконання своїх функцій. Це може включати інтеграцію з платіжними системами, базами даних, хмарними сервісами для зберігання даних, а також зовнішніми API для отримання або обробки інформації, наприклад, для погодних даних або новин.

Планування таких інтеграцій має враховувати різні аспекти, починаючи від визначення необхідних функцій, які повинні бути реалізовані через зовнішні сервіси, і до детальної розробки інтерфейсів для комунікації з цими сервісами. Наприклад, при інтеграції платіжної системи, як-от Stripe чи PayPal, важливо забезпечити належну безпеку транзакцій, обробку помилок та взаємодію з API сервісу для отримання підтверджень або відмов.

Такі інтеграції потребують не лише технічної реалізації, але й правильного управління версіями API, щоб забезпечити безперебійну роботу бота навіть при оновленнях зовнішніх сервісів. Водночас треба враховувати можливість змін у зовнішніх сервісах, тому важливо мати чіткий план щодо обробки таких змін, щоб мінімізувати вплив на роботу бота.

Прикладом може служити інтеграція з зовнішнім сервісом для створення чат-бота підтримки клієнтів, який отримує дані з бази знань компанії або API для реальних запитів. Якщо сервіс змінює структуру відповіді або API, необхідно оперативно коригувати інтерфейс і логіку бота. Тому інтеграція з зовнішніми сервісами вимагає гнучкості та планування на етапі проектування, щоб забезпечити адаптацію до змін і мінімізувати ризики збоїв.

Контингентність на користувачів. Орієнтація на вимоги та зручність використання для кінцевих користувачів, враховуючи їхні очікування від взаємодії з ботом.

В управлінні проектом з розробки телеграм-ботів потрібно врахувати контингентності на користувачів, тобто орієнтація на вимоги та зручність використання для кінцевих користувачів. Це означає, що кожен аспект проекту

повинен бути спланований і виконаний з урахуванням потреб, очікувань і поведінки користувачів.

Початковим етапом є глибоке розуміння цільової аудиторії. Наприклад, якщо бот розробляється для підтримки клієнтів інтернет-магазину, важливо розуміти, які питання найчастіше задають користувачі, якого роду інформацію вони шукають і які функції їм найбільш потрібні. Це може бути відстеження замовлення, інформація про товар або підтримка у вирішенні проблем.

На основі цього аналізу формуються вимоги до бота. Важливо, щоб бот мав інтуїтивно зрозумілий інтерфейс і надавав швидкі та точні відповіді. Якщо користувачі очікують миттєвих відповідей, варто інтегрувати можливості машинного навчання та штучного інтелекту для забезпечення високого рівня автоматизації та якості відповідей.

Зручність використання бота також включає простоту його налаштування і взаємодії. Наприклад, бота можна зробити більш дружнім до користувачів, використовуючи природну мову для комунікації, замість складних команд. Якщо бот використовується для бронювання, можна надати можливість користувачам вибирати дати і час через інтерактивні меню, що знижує ймовірність помилок і робить процес більш зручним. Хорошим прикладом є меню в боті «Вартовий» (Рис. 1.6, 1.9).

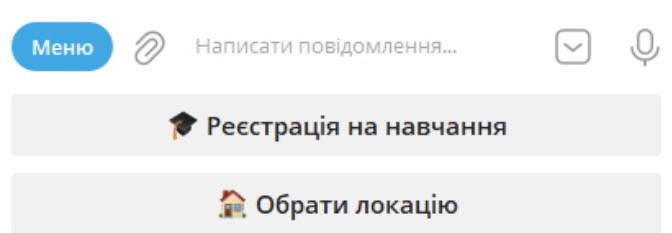


Рис. 1.6. Приклад кнопкового меню в телеграм-боті

Джерело: розроблено автором на основі розробки проекту "Вартовий" у 2024 р.

Варто також враховувати зворотний зв'язок від користувачів. Після запуску бота слід регулярно збирати і аналізувати відгуки, щоб виявляти слабкі місця і можливості для покращення. Наприклад, якщо користувачі скаржаться на повільність відповідей або на те, що бот не завжди правильно розуміє їхні запити, це сигнал до

того, що потрібно вдосконалити алгоритми обробки природної мови або оптимізувати швидкість роботи сервера.

Крім того, важливо адаптувати бота під різні платформи та пристрої, щоб забезпечити доступність для широкого кола користувачів. Наприклад, бот повинен однаково добре працювати як на мобільних пристроях, так і на десктопах.

Безпека та конфіденційність: Забезпечення високого рівня безпеки, оскільки боти можуть обробляти конфіденційні дані користувачів.

Управління проєктом з розробки телеграм-ботів потребує уваги до безпеки та конфіденційності даних користувачів. Це питання критичне, оскільки боти часто обробляють чутливу інформацію, яка може включати особисті дані, фінансову інформацію або інші конфіденційні дані. Для забезпечення високого рівня безпеки необхідно реалізувати кілька ключових аспектів.

По-перше, слід використовувати сучасні методи шифрування для зберігання та передачі даних. Наприклад, дані, які бот отримує від користувачів, повинні бути зашифровані під час передачі через мережу. Це можна забезпечити за допомогою протоколів TLS (Transport Layer Security). Також важливо зашифровувати дані, які зберігаються в базі даних, щоб унеможливити їх використання навіть у разі компрометації серверів [23].

По-друге, необхідно реалізувати строгі заходи автентифікації та авторизації. Наприклад, використовувати двофакторну автентифікацію (2FA) для доступу до адміністративних функцій бота. Це допоможе забезпечити, що лише авторизовані користувачі мають доступ до конфіденційної інформації та критичних функцій бота.

Також важливо регулярно проводити аудит безпеки та тестування на проникнення. Це включає виявлення вразливостей у коді бота та його інфраструктурі, а також впровадження виправлень для усунення виявлених проблем. Наприклад, залучення зовнішніх експертів для проведення тестування на проникнення може допомогти ідентифікувати вразливості, які можуть бути пропущені внутрішньою командою.

Ще одним важливим аспектом є обмеження доступу до даних на рівні необхідності. Це означає, що кожен користувач чи система повинні мати доступ лише до тих даних, які їм необхідні для виконання своїх завдань. Наприклад, якщо бот інтегрований з іншими сервісами, важливо забезпечити, щоб кожен сервіс мав доступ лише до мінімально необхідного набору даних.

Окрім того, необхідно забезпечити резервне копіювання даних та плани відновлення після збоїв. Регулярні резервні копії допоможуть відновити дані у разі їх втрати або компрометації. Наприклад, автоматизовані системи резервного копіювання можуть забезпечити регулярне створення копій даних та їх зберігання у безпечному місці.

Забезпечення високого рівня безпеки та конфіденційності в розробці телеграм-ботів є комплексним завданням, яке включає технічні, організаційні та процедурні заходи. Реалізація цих заходів допоможе мінімізувати ризики та забезпечити захист даних користувачів, що є ключовим фактором успіху проєкту [24].

Постійна підтримка: Надання системи підтримки для оперативного вирішення проблем, оновлення та забезпечення стабільної роботи бота.

Постійна підтримка є не менш важливим аспектом управління проєктом з розробки телеграм-ботів. Вона включає в себе декілька ключових компонентів: оперативне вирішення проблем, регулярне оновлення та забезпечення стабільної роботи бота.

Оперативне вирішення проблем є необхідним для забезпечення безперебійного функціонування телеграм-бота. Наприклад, якщо бот раптово перестає відповідати на запити користувачів або виникають інші технічні проблеми, команда підтримки повинна мати чітко налагоджений процес для швидкого виявлення і виправлення цих неполадок. Це може включати моніторинг роботи бота в режимі реального часу, використання систем автоматичних повідомлень про збої та організацію чіткого ланцюжка комунікації між розробниками та технічною підтримкою.

Регулярне оновлення також є невід'ємною частиною постійної підтримки. Технології та вимоги користувачів постійно змінюються, тому телеграм-бот повинен

відповідати новим стандартам і очікуванням. Це може включати впровадження нових функцій, поліпшення інтерфейсу користувача або адаптацію до змін в API Telegram. Наприклад, якщо Telegram випускає нову версію API, команда розробників повинна оперативно оновити бота, щоб забезпечити його сумісність і використання всіх нових можливостей.

Забезпечення стабільної роботи бота вимагає постійного моніторингу і тестування. Це означає, що потрібно регулярно перевіряти бота на наявність багів, проводити навантажувальні тести для оцінки його продуктивності під час пікових навантажень і забезпечувати безперервний цикл тестування та вдосконалення. Наприклад, команда може використовувати інструменти автоматизованого тестування для перевірки коректності роботи бота після кожного оновлення або зміни в коді. Також важливо мати резервні копії та плани аварійного відновлення на випадок серйозних збоїв або втрати даних.

Постійна підтримка включає і взаємодію з користувачами для отримання зворотного зв'язку. Наприклад, команда може збирати відгуки через опитування або аналізувати дані про використання бота для виявлення найчастіших проблем і запитів. Це дозволяє швидко реагувати на потреби користувачів і покращувати бот, роблячи його більш корисним та ефективним.

Монетизація та комерційні аспекти: Якщо бізнесові цілі включають генерацію прибутку, необхідно розглядати стратегії монетизації та ефективне управління комерційними аспектами.

Монетизація та комерційні аспекти є ключовими елементами в управлінні проектом з розробки телеграм-ботів, якщо бізнесові цілі включають генерацію прибутку. Важливо розробити стратегії, що дозволяють не лише створити якісний продукт, але й забезпечити стабільний грошовий потік.

Однією з поширених стратегій монетизації є впровадження підписок. Бот може надавати базовий набір функцій безкоштовно, а за розширені можливості користувачі можуть сплачувати щомісячну чи річну плату. Наприклад, бот для управління

завданнями може надавати можливість створення обмеженої кількості проєктів безкоштовно, а для необмеженого використання вимагати платну підписку.

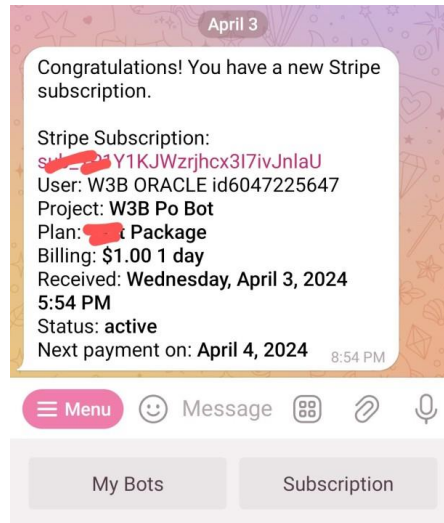


Рис. 1.7. Приклад згенерованого повідомлення бота про підписку

Джерело: [32]

Інший підхід полягає у використанні мікротранзакцій, коли за окремі функції або додатковий контент користувачі сплачують разово. Це може бути актуально для ігрових ботів, де користувачі купують віртуальну валюту чи предмети, або для ботів, що надають унікальний контент, як-от тренувальні програми чи ексклюзивні статті.

Реклама є ще одним ефективним методом монетизації. Телеграм-бот може показувати рекламу або надсилати спонсорвані повідомлення. Для цього потрібно розробити чітку стратегію таргетингу, щоб реклама була релевантною та корисною для користувачів. Наприклад, бот для планування подорожей може співпрацювати з туристичними компаніями, показуючи користувачам спеціальні пропозиції та знижки.

Продаж даних також може бути джерелом доходу, хоча цей метод викликає питання щодо конфіденційності. Дані про поведінку користувачів можуть бути цінними для маркетингових досліджень та аналізу ринку. Однак важливо забезпечити дотримання всіх норм щодо захисту персональних даних і отримати згоду користувачів на обробку їхньої інформації.

Ефективне управління комерційними аспектами вимагає також ретельного планування витрат. Необхідно враховувати витрати на розробку, підтримку,

маркетинг та інші операційні витрати. Важливо забезпечити фінансову стабільність проєкту, маючи резервні кошти на випадок непередбачених обставин або затримок у генерації прибутку.

Відкритий API Telegram: Використання та врахування можливостей, що надає відкритий API Telegram для взаємодії бота з платформою.

Відкритий API Telegram є важливим елементом для розробки телеграм-ботів, надаючи розробникам можливість інтегрувати свої рішення безпосередньо з платформою Telegram. Цей API дозволяє ботам взаємодіяти з користувачами та виконувати широкий спектр функцій, від надсилання повідомлень до обробки платежів. Розглянемо детальніше використання та врахування можливостей цього API в управлінні проєктом.

Telegram Bot API забезпечує простий і зрозумілий інтерфейс для розробників, що дозволяє швидко створювати боти, які можуть відповідати на повідомлення користувачів, надсилати сповіщення, виконувати команди та багато іншого. Важливо враховувати, що кожен бот створюється через офіційний @BotFather, який генерує унікальний токен доступу. Цей токен використовується для автентифікації ботів під час звернення до API.

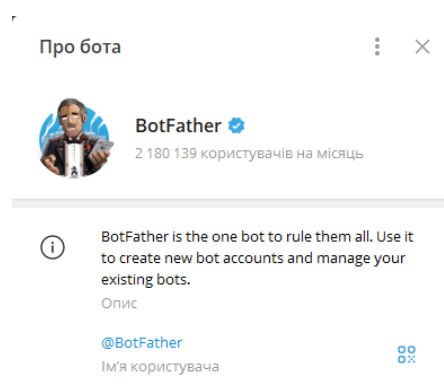


Рис. 1.8. Телеграм-бот для створення власних ботів

Джерело: [https://t.me/BotFather]

Наприклад, при створенні телеграм-бота для підтримки клієнтів, розробник може налаштувати бота таким чином, щоб він автоматично відповідав на часті запитання або перенаправляв складніші запити до живих операторів. Це досягається

за допомогою обробки команд та текстових повідомлень, що надсилаються користувачами.

API дозволяє ботам обробляти різні типи медіа, включаючи зображення, відео та файли. Це відкриває можливості для створення багатофункціональних ботів, які можуть, наприклад, приймати та обробляти зображення документів для подальшої обробки або верифікації.

Крім того, Telegram API підтримує вебхуки, що дозволяє створювати боти, які миттєво реагують на дії користувачів. Це забезпечує більш інтерактивний досвід, оскільки боти можуть оперативно обробляти запити та надсилати відповіді. Наприклад, бот, створений для фінансових консультацій, може використовувати вебхуки для швидкої відповіді на запити щодо поточних курсів валют або стану рахунку користувача.

Важливою особливістю є підтримка опитувань та квізів, що дозволяє створювати інтерактивні опитування або навчальні квізи прямо в чаті з ботом. Це може бути корисним для компаній, які хочуть отримувати зворотній зв'язок від клієнтів або навчати їх за допомогою інтерактивних інструментів.

Telegram API дозволяє інтегрувати платіжні системи, що робить можливим створення ботів для електронної комерції. Бот може обробляти платежі, надаючи користувачам можливість купувати товари або послуги безпосередньо через Telegram. [28]

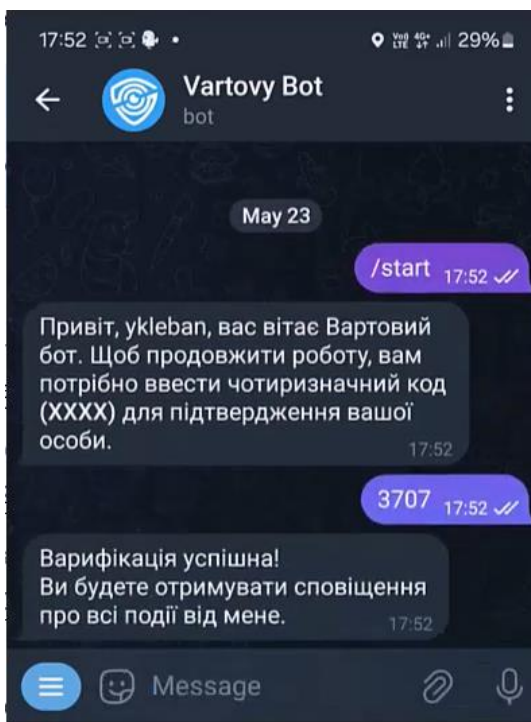


Рис. 1.9. Приклад використання текстової команди телеграм-боті

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

РОЗДІЛ 2. СТРАТЕГІЧНЕ ПЛАНУВАННЯ ТА АНАЛІЗ ВИМОГ ДО ТЕЛЕГРАМ-БОТІВ

2.1. Стратегічне планування в розробці телеграм-ботів

Цей пункт включатиме в себе формулювання ключових стратегічних цілей та тактик для досягнення успіху проєкту. Цей етап передбачає визначення мети бота, аналіз цільової аудиторії, ідентифікацію конкурентної переваги та розробку плану взаємодії з користувачами. Стратегічне планування також враховує фактори, такі як безпека, співпраця з іншими сервісами, масштабованість та довгострокову життєздатність бота. Цей пункт визначає стратегічні кроки, які спрямовані на досягнення успішного розвитку, впровадження та підтримку телеграм-бота на протязі тривалого періоду.

Стратегічне планування в розробці телеграм-ботів є ключовим етапом, що визначає успіх проєкту на всіх його етапах. Воно охоплює довгострокове бачення проєкту, визначення його цілей та завдань, а також розробку шляхів досягнення цих цілей.

По-перше, стратегічне планування починається з аналізу ринку та конкурентного середовища. Це включає вивчення потреб і очікувань користувачів, аналіз існуючих рішень на ринку, а також визначення основних трендів і нових технологій. Знання про ринок дозволяє команді проєкту правильно визначити цільову аудиторію та створити продукт, що максимально відповідає її потребам.

Наприклад, при створенні бота «Вартовий» на той час не було жодних аналогів, що дозволило бути лідером у наданні послуг евакуації.

Другим важливим аспектом стратегічного планування є визначення ключових цілей проєкту. Цілі можуть бути різноманітними: від підвищення рівня автоматизації обслуговування клієнтів до створення інноваційного маркетингового інструменту. Визначення чітких, досяжних і вимірюваних цілей є основою для подальшого планування і прийняття рішень.

Наступним кроком є розробка детального плану реалізації проєкту. План має включати розробку технічного завдання, визначення архітектури системи, вибір технологій та інструментів, а також планування ресурсів. Важливо передбачити всі можливі ризики та шляхи їх мінімізації, щоб уникнути непередбачених затримок і проблем у процесі розробки.

Особлива увага приділяється управлінню ресурсами. Це стосується як людських ресурсів, так і фінансових, технічних та часових. Успішне стратегічне планування передбачає залучення до проєкту кваліфікованих спеціалістів, розподіл ролей і відповідальності між членами команди, а також визначення бюджетів і строків виконання робіт.

Комунікація і взаємодія в команді є ще одним критичним аспектом стратегічного планування. Важливо забезпечити ефективний обмін інформацією, щоб всі учасники проєкту мали чітке уявлення про свої завдання і загальну мету. Регулярні зустрічі, звіти про виконану роботу і коригування планів допомагають тримати проєкт на правильному шляху.

Останнім, але не менш важливим, елементом стратегічного планування є контроль і оцінка результатів. Постійний моніторинг прогресу, аналіз досягнутих результатів і їх порівняння з поставленими цілями дозволяють вчасно виявляти відхилення від плану і вносити необхідні корективи [9, 10].

2.2. Аналіз вимог користувачів та функціональних вимог до телеграм-ботів

В рамках цього пункту проводитиметься систематичне вивчення очікувань та потреб користувачів у контексті Telegram-ботів. Здійснюється детальний розгляд функціональних вимог, які визначають необхідні можливості та операції, які має виконувати бот. Цей етап дослідження спрямований на забезпечення відповідності розроблених ботів потребам користувачів та гармонійне влитання їхніх функціональних очікувань у процес розробки та експлуатації Telegram-ботів.

Аналіз вимог користувачів та функціональних вимог є одним із найважливіших етапів у процесі розробки телеграм-ботів. Він визначає, що саме повинен робити бот,

як він має взаємодіяти з користувачами, а також які функції та можливості повинні бути реалізовані для досягнення цілей проєкту. Це дозволяє не лише забезпечити необхідну корисність та ефективність бота, але й створити зручний, інтуїтивно зрозумілий інтерфейс, який буде відповідати потребам цільової аудиторії.

Першим кроком у цьому процесі є вивчення потреб і очікувань користувачів. Це можна зробити за допомогою різноманітних методів: опитувань, інтерв'ю, тестування прототипів або аналізу відгуків про подібні продукти. Наприклад, якщо телеграм-бот призначений для автоматизації обслуговування клієнтів в інтернет-магазині, потрібно з'ясувати, які саме функції будуть найбільш корисними для користувачів. Це може бути, наприклад, швидкий доступ до інформації про наявність товару, відстеження статусу замовлення, консультації щодо товарів або навіть оформлення замовлень без необхідності звертатися до живого консультанта.

У результаті цього аналізу формується перелік основних вимог до бота, який має відображати всі бажані функції та можливості. Наприклад, функціональні вимоги до бота можуть включати автоматичну обробку запитів користувачів, інтеграцію з платіжними системами, а також можливість налаштування повідомлень або нагадувань.

Функціональні вимоги стосуються технічних характеристик бота, визначаючи його здатність виконувати певні дії (Таблиця 2.2). До таких вимог можуть входити: здатність надсилати текстові або мультимедійні повідомлення, обробляти команди користувачів, інтеграція з базами даних або сторонніми API, забезпечення безпеки даних користувачів тощо. Важливо, щоб ці вимоги були чітко визначені і детально описані, оскільки вони впливають на вибір технологій та інструментів для реалізації бота.

Таблиця 2.2

Функціональні вимоги до телеграм-бота

Функція	Опис	Важливість	Пріоритет
Автентифікація користувача	Перевірка прав доступу користувача.	Висока	Обов'язкова
Взаємодія з користувачем	Обробка запитів та відповідей.	Середня	Обов'язкова
Інтеграція з API	Зв'язок з іншими сервісами (наприклад, для доставки повідомлень).	Висока	Обов'язкова
Збір аналітики	Збір статистики використання бота.	Низька	Опціональна

Джерело: розроблено автором на основі розробки проекту

Не менш важливою складовою є врахування типу користувачів, які будуть взаємодіяти з ботом. Наприклад, якщо бота планується використовувати в навчальному середовищі, то він може включати функції тестування, надання навчальних матеріалів та можливість відстеження прогресу. Якщо ж бот орієнтований на бізнес-користувачів, функціональність може включати створення звітів, аналіз даних або навіть автоматизоване ведення документації.

Крім того, важливим аспектом є також врахування обмежень у часі та ресурсах на етапі розробки. Наприклад, функціональні вимоги мають бути реалістичними з точки зору доступних технологій та часу на реалізацію. У разі необхідності можна зробити компроміси між кількістю функцій та якістю їх реалізації, обираючи пріоритетні задачі для початкових версій бота, з подальшим розширенням функціоналу.

Прикладом успішного аналізу вимог може бути ситуація з розробкою бота для фітнес-центру. Після проведення опитувань серед потенційних користувачів виявилось, що найбільш важливими функціями є запис на тренування, перегляд розкладу, а також можливість отримання персоналізованих рекомендацій щодо

тренувань. Оскільки для реалізації цих вимог потрібно інтегрувати систему управління розкладом і базу даних користувачів, ці функції стали основою для розробки телеграм-бота.

2.3. Управління ризиками у розробці та експлуатації телеграм-ботів

Цей пункт включає в себе ідентифікацію, аналіз та керування потенційними загрозами, які можуть виникнути на різних етапах життєвого циклу проєкту. Зокрема:

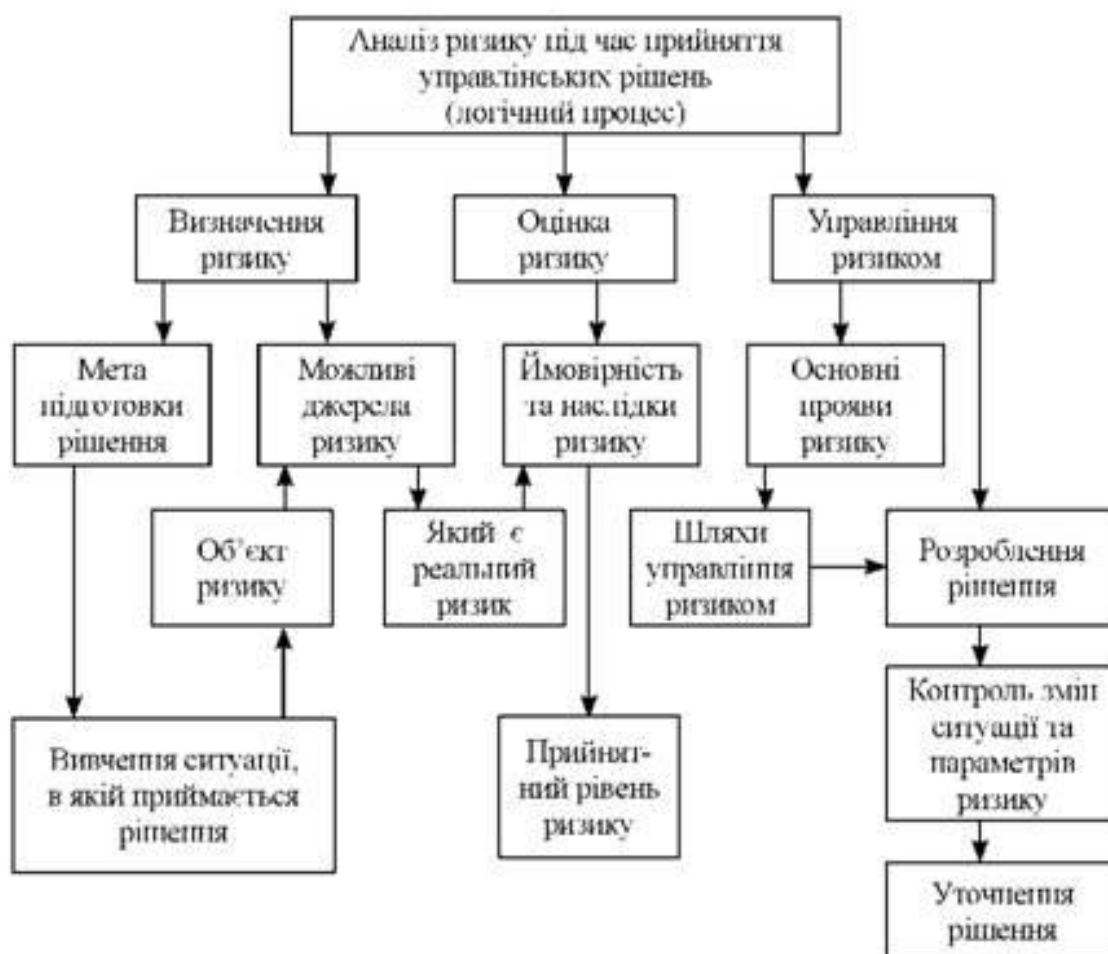


Рис. 2.1. Логічний процес аналізу ризику під час прийняття управлінських рішень

Джерело:

[https://pidru4niki.com/1920042161278/menedzhment/tsili_osnovni_etapi_analizu_riziku]

Ідентифікація ризиків є першочерговим етапом у процесі управління ризиками при розробці та експлуатації телеграм-ботів. Вона включає виявлення потенційних загроз, що можуть негативно вплинути на проєкт, а також аналіз їх можливих

наслідків. Процес ідентифікації ризиків (Рис. 2.1) є безперервним і вимагає систематичного підходу, щоб забезпечити своєчасне виявлення нових ризиків на різних етапах життєвого циклу телеграм-бота.

Спочатку необхідно зібрати інформацію з різних джерел, таких як досвід попередніх проєктів, думки експертів, а також аналіз існуючих практик і стандартів у розробці телеграм-ботів. Наприклад, на початковому етапі розробки можна провести серію мозкових штурмів з командою для визначення можливих технічних та організаційних ризиків.

Одним із ключових ризиків може бути технічний ризик, пов'язаний з обраними технологіями та інструментами. Наприклад, використання нових або недостатньо перевірених технологій може призвести до технічних проблем або затримок у розробці. Для ідентифікації такого ризику необхідно проаналізувати технологічний стек, оцінити його стабільність та підтримку з боку спільноти розробників.

Ще одним важливим ризиком є ризик, пов'язаний з безпекою даних. Телеграм-боти можуть обробляти конфіденційну інформацію користувачів, і будь-який витік даних може мати серйозні наслідки. Ідентифікація цього ризику включає аналіз можливих загроз, таких як хакерські атаки, фішинг або вразливості в програмному забезпеченні. Наприклад, можна провести аудит безпеки системи, щоб виявити потенційні вразливості і розробити стратегії їх усунення.

Ризики, пов'язані з людським фактором, також є критичними. Наприклад, недостатня кваліфікація або досвід команди розробників може призвести до помилок в коді або затримок у проєкті. Для ідентифікації цього ризику слід провести оцінку компетенцій команди, визначити необхідність додаткового навчання або залучення зовнішніх експертів [14].

Експлуатаційні ризики, такі як недоступність сервісу або проблеми з масштабуванням, можуть виникати під час використання телеграм-бота. Ідентифікація цих ризиків включає аналіз можливих сценаріїв збоїв та розробку планів відновлення. Наприклад, можна передбачити резервні сервери або реалізувати механізми автоматичного відновлення роботи системи у випадку збою.

Аналіз ризиків є невід'ємною частиною процесу управління ризиками в розробці та експлуатації телеграм-ботів. Цей процес передбачає виявлення, оцінку і розробку стратегій для мінімізації впливу потенційних ризиків на проєкт (Рис. 2.2).

Перший крок у аналізі ризиків – це ідентифікація можливих ризиків. Розробники повинні визначити всі можливі загрози, що можуть вплинути на проєкт. Наприклад, ризик може бути пов'язаний з технічними аспектами, такими як вразливості в безпеці, нестабільність платформи або помилки в коді. Інші ризики можуть стосуватися організаційних питань, як-от затримки у постачанні ресурсів, відсутність кваліфікованих кадрів чи недостатнє фінансування.

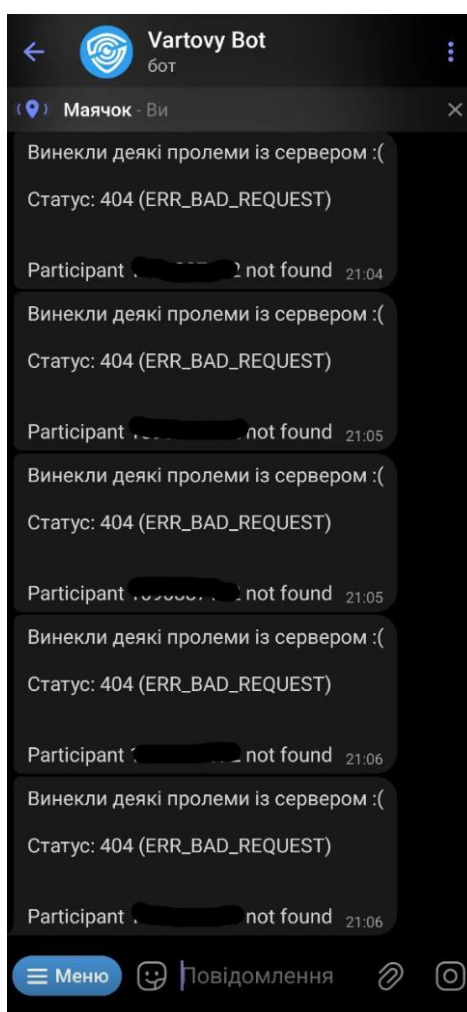


Рис. 2.2. Виявлення критичних помилок у телеграм-боті

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

Наступний етап – це оцінка ризиків, що включає аналіз ймовірності виникнення кожного ризику та його потенційного впливу на проєкт. Наприклад, якщо команда

визначає, що існує ймовірність того, що телеграм-бот може піддатися кібератаці, слід оцінити, наскільки ця ймовірність висока і якими можуть бути наслідки такої атаки, як-от компрометація даних користувачів або збої в роботі бота. Для оцінки ризиків можуть використовуватися різні методи, зокрема аналіз на основі експертних оцінок або статистичних даних.

Після оцінки ризиків, важливо розробити стратегії для їх мінімізації або уникнення. Наприклад, для зниження ризику кібератак можна впровадити додаткові заходи безпеки, такі як двофакторна аутентифікація, регулярні оновлення програмного забезпечення та проведення аудитів безпеки. Якщо ризик пов'язаний з можливими затримками в розробці через нестачу кадрів, можна планувати залучення додаткових фахівців або підготовку резервних співробітників.

В процесі аналізу ризиків важливо також передбачити плани дій на випадок виникнення тих чи інших ризиків. Наприклад, якщо виявлено, що існує високий ризик збою в роботі сервера, слід розробити план резервного копіювання даних і відновлення системи для мінімізації часу простою та втрат інформації.

Планування заходів з управління ризиками є невід'ємною частиною процесу розробки та експлуатації телеграм-ботів. Воно передбачає ідентифікацію можливих ризиків, оцінку їхнього впливу та ймовірності виникнення, а також розробку стратегій для їх мінімізації чи усунення.

На першому етапі управління ризиками визначаються всі потенційні загрози для проекту. Наприклад, одним із ризиків може бути невідповідність розробленого бота очікуванням користувачів. Це може статися через недостатнє вивчення потреб цільової аудиторії або неправильне формулювання вимог до функціональності. Щоб уникнути цього ризику, варто провести глибокий аналіз ринку та залучити до процесу розробки представників цільової аудиторії.

Інший приклад ризику – технічні проблеми, пов'язані з інтеграцією бота з іншими системами або платформами. Це можуть бути непередбачені проблеми з API або зміни в політиці використання платформ, на яких працює бот. Для мінімізації цього ризику слід регулярно оновлювати знання про платформи та підтримувати

зв'язок з їхніми розробниками. Забезпечення гнучкості архітектури бота також допомагає швидко адаптуватися до змін.

Окрім технічних, існують також організаційні ризики, такі як недостатнє фінансування або проблеми з командою розробників. Наприклад, якщо ключовий розробник покине проєкт, це може суттєво затримати його завершення. Для зменшення цього ризику важливо мати чіткий план заміщення кадрів і підтримувати добрі умови праці для команди. Також доцільно залучати кількох розробників, щоб уникнути залежності від одного спеціаліста.

Після ідентифікації ризиків необхідно оцінити їхню ймовірність та потенційний вплив. Наприклад, ризик невідповідності бота очікуванням користувачів може мати високий вплив на успіх проєкту, тоді як ймовірність його виникнення можна зменшити за допомогою додаткових досліджень і тестувань. Таким чином, планування заходів з управління ризиками включає не лише оцінку ймовірності та впливу ризиків, але й розробку конкретних дій для їхнього уникнення або пом'якшення.

Важливим аспектом є регулярний перегляд та оновлення плану управління ризиками. Це дозволяє враховувати нові ризики, що можуть виникнути в процесі розробки та експлуатації бота. Наприклад, після запуску бота можуть з'явитися нові вимоги користувачів, що вимагатимуть змін у функціональності. Відповідно, план управління ризиками має бути динамічним документом, що відображає поточний стан проєкту.

На практиці планування заходів з управління ризиками може включати такі кроки, як створення резервних копій даних, регулярне тестування системи на наявність вразливостей, проведення тренінгів для команди розробників, а також підготовка альтернативних варіантів розвитку подій. Наприклад, для мінімізації ризику втрати даних варто впровадити систему автоматичного резервного копіювання на різні фізичні та хмарні сховища. планування заходів з управління ризиками [21].

Впровадження заходів з управління ризиками є наступним елементом успішного управління проектом розробки та експлуатації телеграм-ботів. Цей процес включає виявлення потенційних ризиків, оцінку їх впливу та розробку стратегій для їх мінімізації або усунення.

Першим кроком у впровадженні заходів з управління ризиками є ідентифікація ризиків. Команда проекту повинна визначити всі можливі загрози, які можуть вплинути на розробку та експлуатацію телеграм-бота. Наприклад, це можуть бути технічні проблеми, такі як несправності серверів, або безпекові ризики, пов'язані з потенційними атаками на систему.

Після ідентифікації ризиків необхідно провести їх оцінку. Це включає визначення ймовірності виникнення кожного ризику та його потенційного впливу на проект. Наприклад, якщо ймовірність збою сервера висока і це може призвести до значних втрат даних, то цей ризик слід оцінити як високий.

Далі слід розробити стратегії мінімізації ризиків. Один з підходів до зменшення технічних ризиків може включати впровадження резервних копій та систем відновлення даних. Наприклад, регулярне створення резервних копій бази даних телеграм-бота допоможе уникнути втрати важливої інформації у разі збою.

Також важливо розглянути питання кібербезпеки. Для мінімізації ризиків, пов'язаних із зломом або несанкціонованим доступом, слід впровадити сучасні методи шифрування даних та автентифікації користувачів. Наприклад, використання двофакторної автентифікації може значно знизити ризик несанкціонованого доступу до облікових записів користувачів.

Комунікаційні ризики, пов'язані з поганою взаємодією між членами команди або зацікавленими сторонами, можна мінімізувати шляхом впровадження ефективних каналів комунікації. Регулярні зустрічі, використання інструментів для управління проектами, таких як Trello або Asana, дозволяють підтримувати прозорість і своєчасність обміну інформацією.

Окрім технічних і комунікаційних ризиків, варто звернути увагу на ризики, пов'язані з вимогами користувачів. Постійний збір зворотного зв'язку від

користувачів і впровадження необхідних змін у функціональність телеграм-бота дозволить адаптувати продукт до потреб ринку і уникнути негативних відгуків.

Моніторинг та контроль є важливими складовими управління ризиками на всіх етапах життєвого циклу телеграм-бота. Під моніторингом розуміється постійне спостереження за функціонуванням бота та системи в цілому, а під контролем — прийняття оперативних заходів у разі виявлення проблем або відхилень від плану. Це дозволяє своєчасно виявляти неполадки та реагувати на них, мінімізуючи негативний вплив на користувачів [22].

Процес моніторингу починається ще до запуску бота. На етапі розробки важливо налаштувати систему логування, яка дозволить відслідковувати всі помилки, збої або нестандартні ситуації під час тестування. Це дозволяє розробникам виявити можливі слабкі місця в коді та усунути їх до того, як продукт потрапить до кінцевих користувачів. Наприклад, якщо під час тестування з'ясується, що бот не обробляє певний тип запиту користувача, це сигналізує про необхідність внесення змін до коду.

Після запуску бота в експлуатацію моніторинг стає ще більш критичним. Основними аспектами є контроль за доступністю бота, швидкістю обробки запитів і своєчасністю відповіді на запити користувачів. Наприклад, якщо бот починає затримувати відповіді або зовсім не відповідає на запити, це може свідчити про перевантаження серверів або про помилки в базі даних. Виявивши таку проблему, адміністратор може оперативно перейти до оптимізації системи або масштабування ресурсів для відновлення нормальної роботи.

Важливим елементом моніторингу є аналіз поведінки користувачів і взаємодії з ботом. Використання інструментів аналітики дозволяє зібрати дані про те, як часто користувачі взаємодіють з ботом, які функції використовують найчастіше, а які не викликають інтересу. Це дає змогу робити висновки про те, чи відповідає продукт потребам аудиторії, і, якщо необхідно, вносити зміни у функціонал або інтерфейс бота.

Контроль здійснюється через регулярні перевірки працездатності системи. Це можуть бути як автоматизовані тести, так і ручні перевірки, які допомагають

переконалися в тому, що всі функції працюють згідно з вимогами. Наприклад, якщо у процесі моніторингу виявлено, що бот не може обробляти запити певного типу, то важливо не лише зафіксувати цю помилку, але й вжити заходів для її виправлення. Це може включати оновлення програмного забезпечення, виправлення кодових помилок або внесення змін у базу даних.

Крім того, моніторинг і контроль охоплюють й інші аспекти безпеки. Одним з основних ризиків є спроби несанкціонованого доступу або атак на систему. У такому разі використання систем моніторингу для виявлення аномальних поведінкових патернів, наприклад, великої кількості запитів з одного IP-адреси, дозволяє вчасно реагувати на потенційні загрози. Механізми контролю дозволяють блокувати шкідливі запити та запобігати втручанню злоумисників.

Управління ризиками у розробці та експлуатації телеграм-ботів спрямоване на максимізацію можливостей успіху та зниження впливу негативних факторів, що можуть виникнути під час роботи з цими інноваційними технологіями [11].

РОЗДІЛ 3. МЕТОДИКИ ТЕСТУВАННЯ ТА ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ТЕЛЕГРАМ-БОТІВ

3.1. Види тестування та їх роль у розробці телеграм-ботів

Види тестування грають ключову роль у розробці телеграм-ботів, забезпечуючи їх надійність, ефективність та відповідність вимогам користувачів. Декілька основних видів тестування та їх ролі в цьому контексті включають:

1. Одиничне тестування

Одиничне тестування, або Unit testing, є першою сходинкою в процесі тестування програмного забезпечення. Воно полягає у перевірці окремих компонентів, таких як функції або класи, на коректність та працездатність. Основна мета одиничного тестування — гарантувати, що кожна частина бота функціонує належним чином, незалежно від інших частин системи.

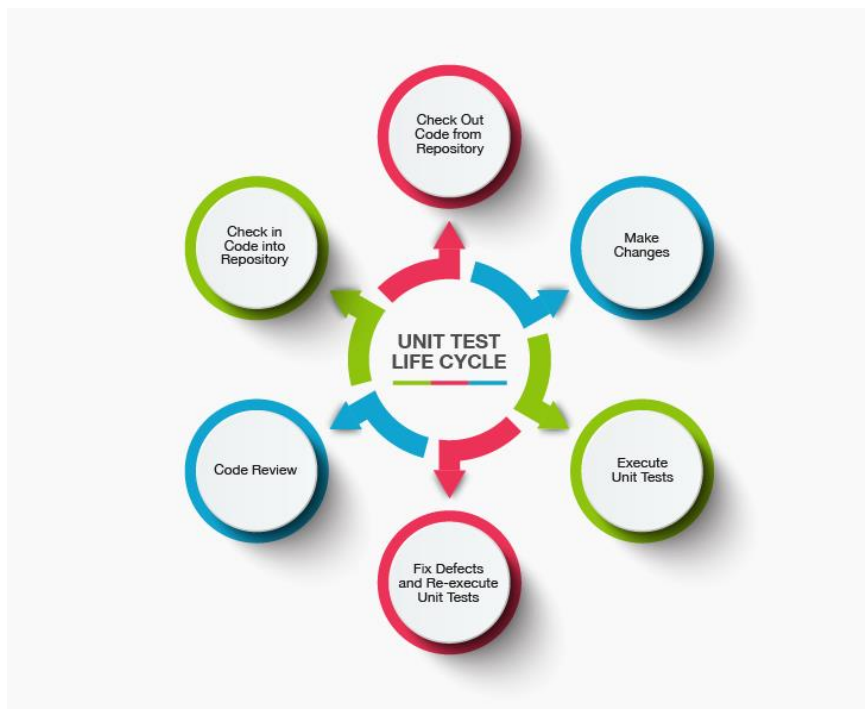


Рис. 3.1. Логічний процес аналізу ризику під час прийняття управлінських рішень
Джерело: [33]

Цей вид тестування проводиться розробниками під час написання коду і допомагає знайти та виправити помилки на ранніх стадіях розробки. Одиничне тестування дозволяє ізолювати кожен функцію або клас, забезпечуючи перевірку на

відповідність очікуваному результату. Це забезпечує надійність та стабільність коду, а також спрощує процес відладки.

У розробці телеграм-ботів одиничне тестування допомагає перевірити базову функціональність окремих модулів, таких як обробка команд користувача, взаємодія з API телеграма, обробка даних та інші ключові компоненти. Наприклад, перевіряючи функцію, яка відповідає за відправлення повідомлення користувачу, тестування може гарантувати, що повідомлення буде надіслано правильно за всіх можливих умов.

Окрім цього, одиничне тестування сприяє підтриманню коду, оскільки дозволяє легко виявити помилки, які можуть виникнути під час внесення змін до коду. Це особливо важливо для проєктів, які активно розвиваються і де регулярно додається новий функціонал. Таким чином, одиничне тестування є критично важливим елементом процесу розробки телеграм-ботів, забезпечуючи високу якість програмного забезпечення та стабільну роботу всіх його компонентів.

2. Інтеграційне тестування

Інтеграційне тестування є наступним етапом у процесі розробки телеграм-ботів, яке перевіряє взаємодію між різними модулями або компонентами системи. Цей вид тестування проводиться після того, як окремі модулі пройшли одиничне тестування, і фокусується на виявленні дефектів, які можуть виникнути внаслідок інтеграції різних частин системи.

Основна мета інтеграційного тестування полягає в тому, щоб забезпечити коректну роботу всіх компонентів при їх об'єднанні. Це включає перевірку взаємодії між модулями, зокрема, передачу даних, виклики функцій та обробку помилок. Наприклад, для телеграм-бота інтеграційне тестування може включати перевірку правильності обробки повідомлень, які надходять від користувачів, взаємодію з зовнішніми API, збереження та отримання даних з бази даних.

Під час інтеграційного тестування важливо враховувати різні сценарії використання, які можуть виникнути під час роботи бота. Це можуть бути як стандартні сценарії, так і випадки, коли користувач вводить некоректні дані або виконує незаплановані дії. Наприклад, тестування може включати перевірку того, як

бот реагує на некоректні команди, чи коректно обробляє великі обсяги даних, а також як він взаємодіє з різними сторонніми сервісами.

Інтеграційне тестування допомагає виявити проблеми, які не можна було помітити під час одиничного тестування окремих модулів. Це можуть бути проблеми з сумісністю між різними компонентами, помилки у форматі даних, що передаються, або невідповідності у логіці обробки даних. Наприклад, може виявитися, що один модуль передає дані у форматі, який не підтримується іншим модулем, що призводить до помилок у роботі системи [33].

Для проведення інтеграційного тестування можуть використовуватися як ручні, так і автоматизовані підходи. Ручне тестування дозволяє детально перевірити специфічні сценарії, які важко автоматизувати, тоді як автоматизоване тестування забезпечує швидке і ефективне виявлення помилок при кожній інтеграції нових змін у код. Використання тестових фреймворків та інструментів для автоматизації інтеграційного тестування значно прискорює процес і підвищує його ефективність.

3. Системне тестування

Системне тестування є ключовим етапом у забезпеченні якості телеграм-ботів, під час якого бот перевіряється як єдине ціле. Метою цього тестування є комплексна оцінка, наскільки система відповідає вимогам і виконує очікувані функції.

На цьому етапі бот перевіряється у повному обсязі, включаючи всі інтегровані компоненти та функціональні можливості. Це дозволяє оцінити, як окремі частини системи взаємодіють між собою і чи немає конфліктів або збоїв у роботі. Системне тестування охоплює всі аспекти функціонування бота, включаючи взаємодію з користувачами, обробку даних, інтеграцію з іншими сервісами та API.

Одним з основних завдань системного тестування є перевірка коректності виконання всіх функцій бота. Наприклад, якщо бот призначений для автоматизації відповіді на запити користувачів, тестування повинно підтвердити, що він правильно обробляє ці запити, відправляє коректні відповіді і виконує усі функції відповідно до специфікації.

Додатково, системне тестування включає перевірку на відповідність вимогам щодо продуктивності та стабільності роботи бота. Це означає, що бот повинен витримувати певне навантаження, обробляти великий обсяг даних та забезпечувати стабільну роботу під час тривалої експлуатації. Тестування також дозволяє виявити, як система реагує на різні помилки або некоректні дії з боку користувачів.

Важливою складовою системного тестування є перевірка зручності використання бота. Це означає оцінку інтерфейсу взаємодії з користувачами, логічності та зрозумілості команд, а також загального користувацького досвіду.

4. Функціональне тестування

Функціональне тестування є важливим етапом у процесі забезпечення якості телеграм-ботів, під час якого перевіряється, наскільки функціональність бота відповідає визначеним вимогам. Основна мета цього виду тестування полягає в тому, щоб переконатися, що всі функції бота виконуються правильно та відповідно до специфікацій [33].

Під час функціонального тестування бот перевіряється на здатність виконувати всі передбачені функції. Це включає в себе перевірку коректності обробки команд користувачів, правильності генерації відповідей, а також взаємодії з іншими системами та сервісами. Наприклад, якщо бот призначений для автоматичного відповідання на запити користувачів, функціональне тестування забезпечує перевірку того, що всі можливі запити обробляються належним чином і на них надсилаються коректні відповіді.

Особлива увага приділяється сценаріям використання, які описують різні ситуації, в яких користувач може взаємодіяти з ботом. Це дозволяє оцінити, наскільки бот відповідає визначеним бізнес-вимогам і чи забезпечує він необхідний рівень користувацького досвіду. Кожна функція тестується окремо, щоб гарантувати, що всі компоненти працюють правильно як в нормальних умовах, так і в умовах високого навантаження або при виникненні помилок.

Функціональне тестування також включає перевірку інтеграції з іншими системами та сервісами. Наприклад, якщо бот використовує зовнішні API для

отримання даних або виконання певних дій, тестування перевіряє, чи коректно відбувається взаємодія з цими API, чи отримуються правильні дані, і чи правильно обробляються відповіді.

Важливою частиною функціонального тестування є регресійне тестування, яке проводиться після внесення змін до коду або додавання нових функцій. Це дозволяє переконатися, що зміни не вплинули негативно на вже існуючі функції бота і що вся система продовжує працювати стабільно.

5. Тестування продуктивності

Тестування продуктивності є важливим етапом у забезпеченні якості телеграм-ботів. Воно спрямоване на оцінку швидкодії та стабільності бота при великому обсязі взаємодій та запитів. Основна мета цього тестування — гарантувати, що бот буде ефективним і надійним в умовах реального використання.

На етапі тестування продуктивності бот піддається симуляції різних сценаріїв, що моделюють реальні умови його експлуатації. Це включає перевірку роботи бота при високому навантаженні, коли велика кількість користувачів одночасно взаємодіє з системою. Важливо оцінити, наскільки швидко бот реагує на запити, як він обробляє дані і чи зберігається стабільність його роботи під час пікових навантажень.

Під час тестування оцінюється затримка відповіді на запити користувачів, а також загальна пропускна здатність системи. Це дозволяє визначити максимальну кількість запитів, яку бот може обробити за одиницю часу без втрати продуктивності. Важливим аспектом є також перевірка ефективності використання ресурсів, таких як процесор, пам'ять та мережеві ресурси, щоб уникнути можливих перевантажень або збоїв у роботі.

Крім того, тестування продуктивності дозволяє виявити потенційні "вузькі місця" в системі, які можуть стати причиною проблем при високих навантаженнях. Наприклад, якщо бот використовує зовнішні API або бази даних, тестування допомагає виявити, чи не виникає затримок або збоїв при взаємодії з цими сервісами.

Результати тестування продуктивності надають цінну інформацію для оптимізації коду та інфраструктури бота. Вони дозволяють внести необхідні зміни

для підвищення ефективності та надійності системи, забезпечуючи тим самим готовність бота до роботи в умовах реального використання з великою кількістю користувачів.

6. Тестування безпеки

Тестування безпеки є критично важливим етапом у розробці телеграм-ботів. Воно спрямоване на виявлення та усунення потенційних загроз та вразливостей, щоб забезпечити захист системи від небажаних втручань та атак. Мета цього тестування — гарантувати, що бот здатен протистояти різним типам загроз і забезпечити безпеку даних користувачів.

Під час тестування безпеки проводиться комплексна перевірка системи на предмет різних вразливостей. Це включає аналіз коду бота на наявність помилок або слабких місць, які можуть бути використані зловмисниками. Особлива увага приділяється перевірці механізмів аутентифікації та авторизації, щоб переконатися, що доступ до конфіденційних даних мають лише уповноважені користувачі.

Важливою частиною тестування безпеки є перевірка на наявність вразливостей, пов'язаних з обробкою даних користувачів. Це включає перевірку захисту від SQL-ін'єкцій, міжсайтового скриптингу (XSS), та інших типів атак, які можуть призвести до витоку даних або компрометації системи. Проводяться тести, що симулюють різні сценарії атак, такі як спроби несанкціонованого доступу, перехоплення даних, та введення шкідливого коду.

Особлива увага приділяється захисту комунікацій між ботом та користувачами. Перевіряється використання шифрування для передачі даних, щоб запобігти їх перехопленню або модифікації під час передачі через мережу. Також аналізується безпека з'єднань з зовнішніми сервісами та API, які використовуються ботом, щоб переконатися, що вони не є потенційним джерелом загроз.

Тестування безпеки включає також перевірку наявності механізмів логування та моніторингу, які дозволяють вчасно виявляти та реагувати на підозрілі дії або спроби атак. Це забезпечує можливість оперативного реагування на інциденти безпеки та мінімізації їх наслідків.

Результати тестування безпеки надають цінну інформацію для покращення захисту системи. Вони дозволяють виявити та усунути вразливості на ранніх етапах розробки, забезпечуючи тим самим високу стійкість бота до потенційних загроз та атак.

7. Тестування користувацького інтерфейсу

Тестування користувацького інтерфейсу (UI) є важливим етапом у розробці телеграм-ботів, який спрямований на оцінку зручності та естетичності інтерфейсу. Мета цього тестування полягає в тому, щоб переконатися, що взаємодія з ботом є інтуїтивно зрозумілою для користувачів, а також забезпечити позитивний досвід користування.

На етапі тестування користувацького інтерфейсу проводиться детальний аналіз всіх елементів інтерфейсу бота, таких як кнопки, меню, форми та повідомлення. Особлива увага приділяється зручності навігації, тобто наскільки легко користувач може знайти потрібні функції та виконати необхідні дії. Перевіряється логічність розташування елементів, їх розмір, колірна гамма та відповідність загальному стилю додатку.

Важливим аспектом тестування користувацького інтерфейсу є перевірка його адаптивності до різних пристроїв та екранів. Бот повинен коректно відображатися як на мобільних телефонах, так і на планшетах або комп'ютерах. Тестування включає перевірку масштабованості інтерфейсу, чи правильно відображаються всі елементи при зміні розміру екрану, а також перевірку наявності та коректності мобільних версій інтерфейсу.

Під час тестування користувацького інтерфейсу проводиться оцінка естетичної складової, тобто наскільки привабливим виглядає інтерфейс для користувачів. Важливо, щоб дизайн був сучасним і відповідним до останніх тенденцій, а також забезпечував приємний користувацький досвід. Естетичний вигляд інтерфейсу впливає на загальне враження від використання бота та може стати вирішальним фактором для його популярності серед користувачів.

Окремо перевіряється інтерактивність інтерфейсу, тобто реакція бота на дії користувачів. Важливо, щоб всі кнопки та інші інтерактивні елементи мали чіткий і зрозумілий зворотний зв'язок. Наприклад, при натисканні кнопки повинна відобразитися відповідна реакція, що підтверджує виконання дії. Це допомагає користувачам зрозуміти, що їхні команди були прийняті і оброблені ботом.

Результати тестування користувацького інтерфейсу надають важливу інформацію для покращення взаємодії з ботом. Вони дозволяють виявити та виправити недоліки в дизайні та зручності використання, що в кінцевому результаті сприяє створенню інтуїтивно зрозумілого та привабливого інтерфейсу. Застосування цих видів тестування в розробці телеграм-ботів сприяє створенню надійних та функціональних продуктів, які відповідають вимогам користувачів і ефективно функціонують в реальних умовах використання.

3.2. Засоби автоматизованого тестування для телеграм-ботів

Засоби автоматизованого тестування для телеграм-ботів включають інструменти та програмні засоби, спеціально розроблені для виконання автоматизованих тестів з метою перевірки функціональностей та ефективності Telegram-ботів. Ці засоби дозволяють автоматизувати процеси тестування, забезпечуючи швидке виявлення помилок, вдосконалення якості програмного продукту та прискорення циклу розробки. Використання таких засобів спрощує тестування взаємодії ботів з користувачами, перевірку реакцій на різні команди та забезпечує стабільність та надійність роботи ботів у різних умовах використання.

Одним із основних інструментів автоматизованого тестування є тестові фреймворки, такі як Mocha (Рис. 3.2), Jest або Jasmine. Ці фреймворки надають можливість створювати тестові сценарії, які автоматично виконуються для перевірки коректності роботи бота. Вони дозволяють визначити очікуваний результат для кожної функції або методу, що дає змогу швидко виявити та локалізувати помилки.

Для автоматизованого тестування взаємодії з API телеграм-ботів широко використовуються інструменти на кшталт Postman або Newman. Вони дозволяють

створювати набори тестових запитів до API, перевіряючи коректність відповідей на різні сценарії. Це особливо важливо для перевірки обробки запитів від користувачів та взаємодії з іншими сервісами.

Для забезпечення стабільності та надійності роботи телеграм-ботів важливим є використання інструментів для безперервної інтеграції та доставки (CI/CD), таких як Jenkins, Travis CI або GitHub Actions. Вони дозволяють автоматично запускати тестові сценарії при кожному зміні коду, що забезпечує швидке виявлення і виправлення помилок на ранніх етапах розробки. CI/CD інструменти також забезпечують можливість автоматичного розгортання нових версій бота, що знижує ризик помилок при ручному оновленні системи.

Ще одним важливим аспектом є тестування користувацького інтерфейсу. Для цього використовуються інструменти на кшталт Selenium або Puppeteer, які дозволяють автоматизувати взаємодію з інтерфейсом бота. Вони симулюють дії користувача, такі як натискання кнопок або введення тексту, перевіряючи коректність відображення і роботи інтерфейсу.

Не менш важливим є використання засобів для навантажувального тестування, таких як Apache JMeter або Gatling. Вони дозволяють симулювати високе навантаження на систему, перевіряючи її стабільність та продуктивність при великій кількості одночасних запитів. Це забезпечує впевненість у тому, що бот буде працювати надійно навіть в умовах пікового навантаження.

GETTING STARTED

```
$ npm install mocha
$ mkdir test
$ $EDITOR test/test.js # or open with your favorite editor
```

In your editor:

```
var assert = require('assert');
describe('Array', function () {
  describe('#indexOf()', function () {
    it('should return -1 when the value is not present', function () {
      assert.equal([1, 2, 3].indexOf(4), -1);
    });
  });
});
```

Back in the terminal:

```
$ ./node_modules/mocha/bin/mocha.js

Array
  #indexOf()
    ✓ should return -1 when the value is not present

1 passing (9ms)
```

Рис. 3.2. Приклад створення автоматизованого тестування за допомогою Моча
Джерело: [<https://mochajs.org/>]

3.3. Забезпечення якості та процеси контролю якості в життєвому циклі телеграм-бота

Забезпечення якості та процеси контролю якості є також важливими етапами в життєвому циклі телеграм-бота, що спрямовані на досягнення високої надійності, ефективності та задоволення потреб користувачів. Ці процеси включають кілька аспектів, які гарантують, що бот працює відповідно до визначених вимог та стандартів.

Першим етапом у забезпеченні якості є планування, яке включає визначення вимог до бота, встановлення критеріїв якості та розробку тестових сценаріїв. Вимоги повинні бути чітко задокументовані, що дозволяє розробникам і тестувальникам мати

однакове розуміння очікуваних результатів. На цьому етапі також визначаються методи тестування, які будуть використовуватися для перевірки функціональності, продуктивності та безпеки бота.

Розробка бота відбувається з урахуванням принципів забезпечення якості, таких як модульність, повторне використання коду та дотримання стандартів кодування. Важливо, щоб код був зрозумілим і добре документованим, що полегшує його тестування та подальше обслуговування. У процесі розробки використовуються інструменти для автоматизованого тестування, що дозволяють виявляти помилки на ранніх етапах.

Після завершення розробки проводиться ретельне тестування, яке включає як ручне, так і автоматизоване тестування. Тестування охоплює перевірку функціональності, продуктивності, безпеки та користувацького інтерфейсу. Результати тестування аналізуються, і виявлені помилки фіксуються та виправляються. Цей процес повторюється до тих пір, поки бот не досягне необхідного рівня якості.

Окрім тестування, важливу роль відіграє процес перевірки та валідації, який забезпечує відповідність бота визначеним вимогам та стандартам. Перевірка включає аналіз документації, код-рев'ю та оцінку тестових звітів. Валідація, у свою чергу, передбачає оцінку реальної роботи бота в умовах, близьких до експлуатаційних.

Після успішного тестування та валідації бот готовий до впровадження та експлуатації. На цьому етапі важливо забезпечити безперервний моніторинг роботи бота, щоб своєчасно виявляти та усувати потенційні проблеми. Використання інструментів для моніторингу дозволяє відстежувати продуктивність, стабільність та безпеку бота в режимі реального часу.

Зворотній зв'язок від користувачів також відіграє важливу роль у забезпеченні якості. Отримані відгуки дозволяють виявляти недоліки та потенційні покращення, що сприяє постійному вдосконаленню бота. Регулярне оновлення та додавання нового функціоналу, базуючись на відгуках користувачів, забезпечує актуальність та відповідність бота потребам користувачів.

РОЗДІЛ 4. ЕФЕКТИВНЕ УПРАВЛІННЯ ТА ПІДТРИМКА ТЕЛЕГРАМ-БОТІВ

4.1. Системи моніторингу та аналізу даних в управлінні телеграм-ботами

Системи моніторингу та аналізу даних є важливими інструментами для ефективного управління телеграм-ботами, оскільки вони забезпечують реальний огляд роботи бота, дозволяючи своєчасно реагувати на виникаючі проблеми та оптимізувати його продуктивність. Використання таких систем дозволяє отримати точні дані про те, як бот взаємодіє з користувачами, виявляти можливі помилки, покращувати функціональність і забезпечувати високу якість обслуговування.

Моніторинг бота в реальному часі дозволяє виявляти будь-які аномалії або несправності в роботі системи. Це може включати відстеження часу відповіді бота на запити, його доступність, рівень завантаженості серверів, а також продуктивність у різних умовах. Системи моніторингу зазвичай працюють за допомогою спеціальних інструментів, таких як Prometheus, Grafana або Zabbix, які надають інформацію про стан сервісів, що підтримують бота, та дозволяють оперативно реагувати на технічні проблеми.

Крім того, моніторинг охоплює не лише технічні аспекти, а й відслідковування поведінки користувачів. Для цього використовуються аналітичні інструменти, такі як Google Analytics, Mixpanel або інші спеціалізовані сервіси. Вони дозволяють збирати інформацію про те, як часто користувачі взаємодіють з ботом, які функції є найбільш популярними, скільки часу проводять на певних етапах взаємодії та які запити виконуються найчастіше. Це допомагає краще розуміти потреби користувачів і адаптувати функціонал бота відповідно до їх вимог.

Аналіз даних із систем моніторингу дозволяє виявляти корисні тренди та патерни. Наприклад, можна оцінити, чи є які-небудь збої, що часто повторюються, або які частини бота потребують оптимізації для покращення ефективності. Данні про час відгуку на запити або використання конкретних функцій можуть допомогти в плануванні покращень, підвищуючи задоволеність користувачів [10].

Також важливим аспектом є забезпечення безпеки даних, які передаються через бота. Системи моніторингу можуть включати функції виявлення спроб несанкціонованого доступу, атак або інші вразливості. Це дозволяє миттєво реагувати на потенційні загрози, запобігаючи їхньому розвитку.

Моніторинг також включає звітність та аналіз, що дозволяє команді розробників отримувати регулярні дані про стан бота та ефективність його функціонування. Це можуть бути як автоматичні звіти, що генеруються системами моніторингу, так і спеціально підготовлені аналізи, що надають більш глибоке розуміння взаємодії користувачів з ботом.

4.2. Планування та впровадження оновлень та нового функціоналу телеграм-ботів

У цьому пункті вивчення визначається стратегія планування та впровадження оновлень та нового функціоналу для телеграм-ботів. Включає в себе:

1. Стратегічне планування.

Стратегічне планування є основою для ефективного впровадження оновлень та нового функціоналу в телеграм-боти, оскільки воно визначає загальний напрямок розвитку бота і допомагає зберегти його релевантність на ринку. Правильно розроблена стратегія дозволяє оптимально поєднати технічні можливості, потреби користувачів та зовнішні фактори, які можуть вплинути на функціонування бота.

Першим етапом стратегічного планування є чітке визначення цілей, які повинні бути досягнуті завдяки оновленням. Це може бути покращення продуктивності бота, розширення його функціоналу, адаптація до змін у вимогах користувачів чи інтеграція з новими платформами. Визначення цих цілей дозволяє створити чітке бачення того, як бот повинен розвиватися, що в свою чергу дозволяє скласти дорожню карту для розробки та впровадження оновлень.

Один із важливих аспектів стратегічного планування — це врахування потреб користувачів. Для цього проводяться дослідження, аналітика використання бота та зворотний зв'язок від користувачів. Це дає змогу зрозуміти, які функції користувачі

вважають найбільш корисними, а які, можливо, потребують покращення чи оновлення. Також важливо враховувати очікування користувачів щодо нових можливостей бота, зокрема інтеграцію з іншими платформами, поліпшення інтерфейсу або додавання нових функцій, які допомагають вирішувати їхні завдання.

Окрім цього, стратегічне планування також передбачає вивчення ринкових тенденцій, тобто аналіз конкурентів та новітніх технологій. Важливо враховувати, як інші телеграм-боти та подібні технології розвиваються, які функціональні можливості з'являються на ринку. Це дозволяє підтримувати конкурентоспроможність бота та забезпечувати його актуальність серед користувачів. Наприклад, якщо з'являються нові функції або інтеграції, які активно використовуються іншими ботами, це може стати сигналом для необхідності їх реалізації у вашому продукті [18].

Стратегію також слід адаптувати до зміни умов на ринку та технологічного середовища. Зміни в алгоритмах Telegram, нові вимоги до безпеки чи оновлення програмного забезпечення можуть вимагати швидкої реакції та внесення коригувань у стратегію розгортання оновлень. Тому важливо, щоб стратегічне планування включало механізми для гнучкого реагування на зміни зовнішнього середовища, а також для коригування пріоритетів.

Створення стратегії для оновлень та нового функціоналу є комплексним процесом, який вимагає узгодження технічних, комерційних і користувацьких аспектів. Завдяки стратегічному плануванню можна оптимізувати процеси розробки та впровадження, забезпечуючи максимальну ефективність і відповідність потребам як користувачів, так і ринку в цілому.

2. Процес визначення функціоналу

Процес визначення функціоналу для оновлень та нових можливостей телеграм-ботів є важливим етапом у розробці, який передбачає глибокий аналіз потреб користувачів, а також ретельне врахування технічних обмежень і можливостей системи. Це дозволяє створити функціональні можливості, які будуть корисні та зручні для користувачів, одночасно враховуючи обмеження, з якими стикається команда розробників під час реалізації нових ідей.

Першим етапом процесу є аналіз потреб користувачів. Це передбачає активне збирання зворотного зв'язку від тих, хто регулярно взаємодіє з ботом. Для цього використовуються різноманітні методи: опитування користувачів, вивчення статистики використання бота, аналіз часто задаваних питань та виявлення найбільш запитуваних функцій. Важливо, щоб команда розробників мала точне уявлення про те, які проблеми користувачі стикаються під час роботи з ботом, а також що саме їм необхідно для поліпшення досвіду використання.

Збір відгуків може також здійснюватися через безпосередню взаємодію з користувачами у самій програмі, наприклад, через спеціальні команди або форми зворотного зв'язку, що дозволяє отримати точні пропозиції та побажання. Ключовим моментом є правильний підхід до відбору запитів користувачів — не кожен відгук є рівнозначно важливим. Тому важливо оцінювати відгуки в контексті загальної стратегії бота та його цілей.

Після збору даних про потреби користувачів, наступним етапом є врахування технічних обмежень і можливостей. Наприклад, деякі функції можуть потребувати значних ресурсів на сервері, що може бути неприпустимим у рамках поточного технічного середовища. Іноді можуть виникати питання сумісності з іншими платформами або обмеження API Telegram, що обмежують можливість реалізації певних ідей. Також слід врахувати обмеження, пов'язані з безпекою даних або захистом конфіденційності, особливо якщо мова йде про нові способи обробки особистої інформації користувачів [35].

Врахування цих технічних моментів дозволяє уникнути ситуацій, коли пропоновані функції виявляються неможливими для реалізації через наявні обмеження. При плануванні нового функціоналу також важливо враховувати масштабованість рішення, тобто здатність бота працювати при великій кількості одночасних користувачів без втрати продуктивності.

Визначення функціоналу має включати взаємодію з усіма зацікавленими сторонами: розробниками, користувачами, аналітиками та маркетологами. Після того, як буде проведено аналіз і визначено бажаний функціонал, важливо за

допомогою прототипів чи описів перевірити, чи відповідають ці функції потребам користувачів і чи не суперечать вимогам системи. Тільки після цього можна приступити до створення детального плану впровадження оновлень.

3. Планування Релізів

Планування релізів передбачає ефективний розподіл робіт, визначення чітких часових рамок для кожного етапу та встановлення пріоритетів для релізів. Такий підхід дозволяє забезпечити своєчасне та безпечне розгортання оновлень без негативних впливів на роботу бота.

Процес планування починається з аналізу всіх функцій та змін, які повинні бути реалізовані. Після цього весь новий функціонал розподіляється на логічні релізи, кожен з яких містить певну групу змін. Релізи можуть бути спрямовані на впровадження конкретних можливостей або ж на покращення вже існуючих функцій. Важливо, щоб кожен реліз мав чітко визначену мету і не включав більше змін, ніж це необхідно для досягнення поставлених цілей. Це дозволяє зменшити ризик виникнення помилок і полегшує тестування.

Визначення часового графіка для кожного релізу є критично важливим, оскільки невірно розраховані терміни можуть призвести до затримок в оновленні бота або ж до неконтрольованого зростання складності задач. Для кожного релізу має бути встановлений чіткий графік, що включає всі етапи розробки, тестування та впровадження оновлень. Кожен етап має свою тривалість і визначені відповідальні особи, що дає змогу команді чітко орієнтуватися в строках виконання завдань.

Пріоритети для кожного релізу встановлюються на основі важливості та терміновості кожної зміни. Це дозволяє зосередити ресурси на найбільш критичних функціях і забезпечити їх реалізацію перш за все. При цьому варто враховувати як технічні, так і бізнес-потреби. Наприклад, якщо певні оновлення необхідні для того, щоб відповідати новим вимогам користувачів або технологічним стандартам, вони повинні отримати високий пріоритет. Водночас менш важливі функції або вдосконалення можуть бути відкладені на наступні етапи [35].

Крім того, при плануванні релізів необхідно враховувати можливі залежності між різними частинами функціоналу. Наприклад, деякі зміни можуть вимагати попереднього оновлення інфраструктури або інтеграцій з іншими системами, і ці залежності слід врахувати при визначенні пріоритетів і строків. Якщо для успішного впровадження певної функції необхідно виконати інші кроки, це має бути включено в загальний план релізу.

4. Тестування та QA

Тестування та забезпечення якості (QA) є невід'ємною частиною процесу планування та впровадження оновлень і нового функціоналу для телеграм-ботів. Перед тим як нові функції або оновлення будуть інтегровані в основну систему бота, їх необхідно ретельно протестувати, щоб гарантовано виявити і виправити можливі помилки або недоліки (Рис. 4.1). Це дає змогу забезпечити, що зміни не порушать стабільність роботи бота і відповідатимуть вимогам користувачів.

Перш за все, важливо розробити чіткий тестовий план, що включає всі аспекти нововведень, які потрібно перевірити. Тестування повинно охоплювати всі рівні функціональності, починаючи від окремих компонентів і закінчуючи повним тестуванням системи, що включає новий функціонал. У цьому контексті велике значення має автоматизація тестів, що дозволяє прискорити перевірку, зменшити людський фактор і підвищити ефективність тестувальних заходів. Крім того, автоматизовані тести допомагають постійно перевіряти роботу бота після кожного оновлення, мінімізуючи ризик виникнення непередбачених помилок.

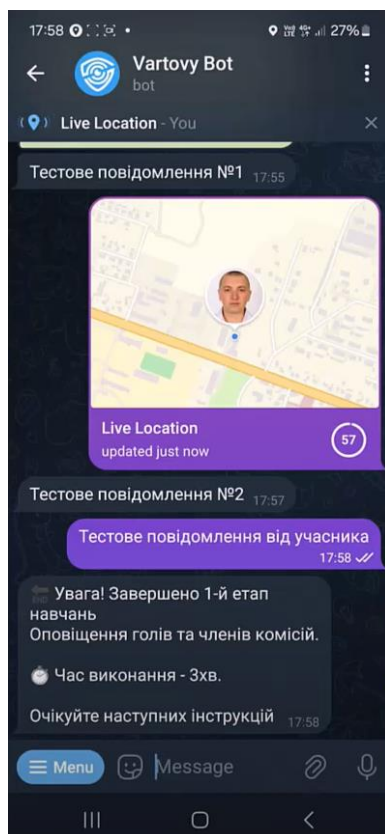


Рис. 4.1. Проведення функціонального тестування Telegram-бота

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

Важливим етапом є виконання регресійного тестування, яке допомагає виявити потенційні проблеми, що можуть виникнути через інтеграцію нового функціоналу з існуючими частинами бота. Такий підхід гарантує, що нові зміни не порушують роботу вже наявних функцій, і вони будуть працювати коректно після оновлення.

Окрім цього, тести мають включати перевірку сумісності нових функцій з різними версіями операційних систем і пристроїв, оскільки телеграм-бот може взаємодіяти з користувачами через різні платформи та пристрої. Також необхідно враховувати тестування безпеки, щоб переконатися в захищеності нововведень від можливих атак.

Забезпечення високої якості та надійності оновлень передбачає не лише їх тестування, але й регулярний процес оцінки результатів тестування. Якщо виявлено будь-які недоліки чи помилки, їх необхідно виправити до впровадження в основну систему. Для цього проводиться аналіз помилок, який допомагає зрозуміти причини їх виникнення, і вживаються заходи для уникнення подібних проблем у майбутньому.

Завдяки ефективному тестуванню та суворому контролю якості забезпечується не лише функціональність, але й стабільність і безпека бота після кожного оновлення. Це також дозволяє задовольняти вимоги користувачів, підтримуючи високий рівень задоволеності та мінімізуючи ризики збоїв у роботі системи.

5. Комунікація з Користувачами

Успішне планування та впровадження оновлень і нового функціоналу для телеграм-ботів вимагає ретельного підходу до комунікації з користувачами. Користувачі повинні бути чітко проінформовані про зміни, щоб знати, що саме було оновлено, які нові можливості з'явилися і як вони можуть скористатися новим функціоналом. Це важливо для забезпечення безперебійної роботи бота та збереження довіри користувачів.

Підготовка ефективної комунікаційної стратегії є ключовим етапом, який включає розробку чітких повідомлень та визначення каналів комунікації. Стратегію слід розробляти таким чином, щоб вона охоплювала різні аспекти: інформування користувачів про нововведення, надання детальних інструкцій, роз'яснення змін у функціонуванні бота. Повідомлення мають бути зрозумілими, стиснутими та в той же час досить інформативними, щоб користувачі не відчували себе розгубленими перед новими функціями або змінами.

Інформація про оновлення може бути надана через різні канали в межах самого бота, включаючи автоматичні повідомлення, спеціальні сповіщення або інтерактивні меню, які дозволяють користувачам ознайомитися з новими функціями. Важливим елементом є також надання користувачам доступу до інструкцій щодо використання нового функціоналу. Це можуть бути короткі текстові інструкції, посилання на детальніші гайди або відео, що пояснюють, як користуватися новими можливостями. Такі ресурси допомагають користувачам швидко освоїти зміни та не витратити час на пошук відповіді на питання.

Також слід враховувати можливість отримання зворотного зв'язку від користувачів. Це дозволяє оперативно реагувати на питання або проблеми, що виникають у процесі використання оновлених функцій. Користувачі повинні мати

можливість без труднощів звертатися з запитаннями або пропозиціями через чати підтримки, а також бути впевненими, що їхні звернення будуть розглянуті.

Забезпечення доступу до інформації про зміни та інструкцій має допомогти зменшити кількість помилок при використанні нового функціоналу, а також підвищити загальне задоволення користувачів. Вчасне і зрозуміле інформування допомагає створити позитивне враження від оновлень, заохочує до використання нових можливостей і підтримує користувачів на всіх етапах взаємодії з ботом [35].

6. Впровадження

Планування та впровадження оновлень і нового функціоналу для телеграм-ботів вимагає ретельної організації, щоб процес змін не порушував стабільність роботи бота та забезпечував безперервність сервісу для користувачів. Плавне впровадження є важливим для підтримки високої якості обслуговування користувачів та запобігання негативним наслідкам, таким як затримки чи помилки в роботі бота.

Одним із ключових аспектів є організація такого впровадження, яке мінімізує перерви у роботі. Це може включати виконання оновлень у позаробочий час або в періоди з низькою активністю користувачів, коли кількість запитів є найменшою. Також часто використовується поетапне впровадження, коли новий функціонал або оновлення поступово інтегруються, що дає змогу вчасно виявити потенційні проблеми і знизити ризик виникнення серйозних збоїв.

Щоб гарантувати, що оновлення не призведуть до значних збоїв або неполадок, важливо регулярно моніторити процес впровадження. Це включає відслідковування системних показників, таких як завантаження серверів, швидкість обробки запитів, а також функціонування нових компонентів бота. У разі виявлення проблем необхідно швидко реагувати, зупиняти або коригувати оновлення, якщо це необхідно, щоб уникнути більш серйозних наслідків для користувачів.

Моніторинг процесу впровадження допомагає також виявляти проблеми, які можуть виникнути лише в умовах реального використання, що може бути не передбачено під час попередніх тестів. Завдяки постійному спостереженню можна

своєчасно вносити корективи, оптимізувати роботу нових функцій або виявити неочікувані збої.

Для досягнення успішного впровадження оновлень також важливо забезпечити комунікацію з користувачами щодо будь-яких можливих затримок або неполадок, що можуть виникнути під час впровадження. Інформування користувачів про поточний стан оновлення або його завершення дозволяє зберегти довіру до бота та його функціональності.

7. Збір та аналіз зворотнього зв'язку

Після впровадження оновлень або нового функціоналу для телеграм-бота потрібно забезпечити систематичний збір зворотного зв'язку від користувачів. Це дозволяє не тільки оцінити, наскільки вдало було виконано впровадження, але й виявити потенційні недоліки або проблеми, які могли не бути помічені під час тестування. Зворотний зв'язок може бути зібраний через різні канали: через опитування, через прямі комунікації з користувачами в чатах, через автоматичні анкети або за допомогою спеціальних інтерфейсів для збору фідбеку всередині самого бота.

Важливим аспектом є не тільки кількість отриманих відгуків, а й їх зміст. Необхідно звертати увагу на ті моменти, де користувачі стикаються з труднощами або висловлюють незадоволеність, оскільки ці сигнали можуть вказувати на проблеми, які потребують негайного виправлення. Крім того, важливо враховувати загальний тон відгуків, оскільки це дозволяє зрозуміти, як новий функціонал або оновлення сприймаються аудиторією в цілому.

Аналіз зворотного зв'язку включає в себе не лише обробку негативних коментарів, але й виділення позитивних аспектів, що підтверджують правильність впровадження. Часто користувачі можуть вказувати на зручність нового інтерфейсу або ефективність нових функцій, що є важливими критеріями для подальшої роботи.

На основі зібраної інформації від користувачів проводиться аналіз, що дозволяє оцінити ефективність впровадженого оновлення та його відповідність первинним вимогам. Це також допомагає зрозуміти, чи є потреба в додаткових корективках або

удосконаленнях. Зворотний зв'язок може стати основою для прийняття рішень щодо подальших оновлень або зміни функціоналу.

У деяких випадках, якщо зворотний зв'язок показує серйозні недоліки або труднощі, виникає необхідність у швидкому внесенні коректив. Наприклад, це може бути вдосконалення інтерфейсу або зміна логіки роботи функціоналу для полегшення взаємодії користувачів. Такий підхід не тільки допомагає усунути помилки, але й забезпечує позитивний досвід користувачів, що підвищує їх лояльність і задоволення від використання бота.

В кінцевому рахунку, збір і аналіз зворотного зв'язку після впровадження оновлень або нового функціоналу є важливою частиною процесу управління життєвим циклом бота. Це дозволяє постійно вдосконалювати продукт, відповідаючи на потреби користувачів і підтримуючи високу якість обслуговування.

Цей пункт дослідження спрямований на розробку ефективних стратегій для планування, тестування та впровадження оновлень у телеграм-ботах з метою забезпечення їхньої надійності та високої якості функціонування [35].

4.3. Управління взаємодією з користувачами та обробка звернень

Взаємодія з користувачами охоплює не тільки забезпечення нормальної роботи бота, а й підтримку зручного, зрозумілого і ефективного каналу комунікації. Це включає взаємодію в процесі надання послуг, допомоги або інформації користувачам, а також своєчасну обробку будь-яких запитів або проблем, що виникають.

Забезпечення високої якості взаємодії з користувачами полягає в тому, щоб кожне звернення було вирішене швидко та ефективно. Успішна взаємодія починається з чіткої та простого в користуванні інтерфейсу, де користувачі легко можуть знайти необхідну функцію або отримати допомогу. Важливо також, щоб бот був здатний відповідати на різноманітні запити користувачів, адаптуючи свої відповіді до контексту.

Обробка звернень користувачів включає в себе як автоматизовані, так і вручну оброблювані запити. Автоматизація може бути здійснена через систему шаблонних

відповідей або спеціалізованих функцій, які розпізнають тип звернення і надають відповідну відповідь. Водночас для складніших або більш специфічних запитів потрібно забезпечити можливість їх передачі до відповідних фахівців або підтримки, яка займається розв'язанням проблем. Це може включати функціонал ескалації запитів до адміністратора чи відповідної служби підтримки.

Важливим аспектом є надання зворотного зв'язку після обробки звернення, що допомагає користувачам переконатися у вирішенні їхнього питання або проблеми. Користувачі повинні отримати інформацію про те, як їхнє звернення було оброблено, а також можливість для подальших уточнень або запитів. Висока швидкість і ефективність реагування на звернення часто стають визначальними факторами для користувачів, які використовують телеграм-бота в повсякденному житті.

Окрім того, управління взаємодією з користувачами включає моніторинг процесу комунікації, аналіз частоти та типів запитів, виявлення найбільш поширених проблем і швидке реагування на них. Це дозволяє своєчасно вносити зміни в функціональність бота або розробляти нові інструменти для покращення взаємодії.

Налагодження ефективної обробки звернень також дозволяє підтримувати високу задоволеність користувачів і забезпечує позитивний досвід взаємодії. Це, в свою чергу, підвищує лояльність користувачів до бота та сприяє популяризації сервісу. Тому взаємодія з користувачами повинна бути не лише швидкою, але й ефективною, зрозумілою і комфортною [34].

ВИСНОВКИ

Детальний розгляд ключових аспектів, що визначають специфіку управління проектами в розробці та підтримці телеграм-ботів показав, що проєкт розробки та підтримки телеграм-бота має низку характеристик, таких як унікальність, тимчасові обмеження та орієнтація на специфічні вимоги користувачів або компаній, для яких створюється бот. Важливу роль у розвитку проєкту відіграють етапи його життєвого циклу, які варіюються від ініціації до завершення, включаючи етапи тестування, оптимізації та підтримки.

Управління проєктом у сфері розробки телеграм-ботів має особливості, зумовлені специфікою технологій, таких як інтеграція з API Telegram, обробка запитів у реальному часі, а також потреба в швидкому реагуванні на зміни та оновлення API, що можуть вплинути на функціональність бота. Крім того, з огляду на обмеження часу і ресурсів, важливою є адаптивність управлінських підходів, здатність швидко реагувати на непередбачувані ситуації, а також ефективна комунікація між розробниками, тестувальниками та замовниками проєкту.

Розглядаються ключові аспекти стратегічного планування та аналізу вимог до телеграм-ботів, що є невід'ємною частиною успішної розробки та експлуатації цих цифрових інструментів.

Стратегічне планування в розробці телеграм-ботів відіграє вирішальну роль у процесі створення телеграм-ботів. Ефективне стратегічне планування включає в себе визначення мети та завдань бота, аналіз ринку та цільової аудиторії, а також розробку стратегії розвитку та підтримки бота. Важливо враховувати потенційні виклики та можливості, що можуть виникнути під час реалізації проєкту. Ретельне планування дозволяє мінімізувати ризики та забезпечити довгострокову життєздатність бота.

Детально розглянуто аналіз вимог користувачів та функціональних вимог до телеграм-ботів. Для забезпечення успішної взаємодії з користувачами необхідно провести глибоке дослідження їх потреб і очікувань. Визначення функціональних вимог включає ідентифікацію основних функцій та можливостей, які бот повинен мати для задоволення цих потреб. Це включає визначення сценаріїв використання,

проектування інтерфейсу користувача, а також врахування зручності та доступності для різних категорій користувачів. Аналіз вимог є критично важливим етапом, який визначає, наскільки бот буде корисним та ефективним для кінцевих користувачів.

Розробка бота супроводжується різноманітними ризиками, такими як технічні проблеми, недоліки в безпеці, непередбачені зміни у вимогах користувачів та інші. Для успішного управління ризиками необхідно розробити план управління ризиками, який включає ідентифікацію можливих ризиків, оцінку їх впливу та розробку стратегій для їх мінімізації. Регулярний моніторинг та оцінка ризиків дозволяють своєчасно реагувати на виникаючі проблеми та забезпечувати безперебійну роботу бота.

Зокрема, ми детально зупинилися на видах тестування, засобах автоматизованого тестування, а також процесах забезпечення якості на всіх етапах життєвого циклу розробки телеграм-ботів.

Перш за все, було визначено, що тестування відіграє ключову роль у розробці телеграм-ботів. Воно забезпечує виявлення і виправлення помилок на ранніх етапах, що дозволяє мінімізувати ризики та уникнути проблем у майбутньому. Було розглянуто різні види тестування, включаючи модульне тестування, інтеграційне тестування, системне тестування та тестування продуктивності. Кожен з цих видів тестування має свої особливості та цілі, але всі вони спрямовані на забезпечення стабільності та надійності роботи телеграм-бота.

Особливу увагу було приділено засобам автоматизованого тестування, які значно спрощують процес тестування та забезпечують більш високу ефективність. Було розглянуто різні інструменти та фреймворки, які можна використовувати для автоматизації тестування телеграм-ботів. Зокрема, були описані такі інструменти, як Selenium, Pytest та Telethon. Вони дозволяють автоматизувати різні аспекти тестування, включаючи перевірку функціональності, навантажувальне тестування та тестування інтерфейсу користувача.

Процеси забезпечення якості та контролю якості також є невід'ємною частиною життєвого циклу розробки телеграм-ботів. Забезпечення якості включає в себе як

превентивні заходи, так і контрольні дії, що здійснюються на всіх етапах розробки. Це дозволяє забезпечити відповідність розробленого продукту встановленим вимогам і стандартам. Контроль якості передбачає проведення регулярних перевірок та аудиту для виявлення можливих недоліків та їх подальшого усунення.

Таким чином, в результаті проведеного дослідження було встановлено, що ефективне тестування та забезпечення якості є критично важливими для успішної розробки телеграм-ботів. Використання різноманітних методик тестування та інструментів автоматизації дозволяє досягти високого рівня надійності та стабільності роботи телеграм-ботів. Забезпечення якості та контроль якості на всіх етапах життєвого циклу розробки гарантують відповідність кінцевого продукту вимогам користувачів та стандартам галузі.

У розділі 4 детально розглянуто різні аспекти ефективного управління та підтримки телеграм-ботів, що є важливим елементом їх успішного функціонування. Основний акцент було зроблено на системах моніторингу та аналізу даних, плануванні та впровадженні оновлень, а також на управлінні взаємодією з користувачами.

Перш за все, розглянуто системи моніторингу та аналізу даних в управлінні телеграм-ботами. Вони є необхідними для забезпечення безперервної роботи та підвищення ефективності ботів. Зокрема, моніторинг дозволяє виявляти технічні проблеми на ранніх етапах, що сприяє швидкому їх усуненню. Аналіз даних дає змогу оцінювати ефективність ботів, визначати найбільш популярні функції та виявляти можливі точки росту. Такі інструменти, як Grafana, Prometheus та Google Analytics, є важливими компонентами цього процесу.

Далі розглянуто питання планування та впровадження оновлень і нового функціоналу. Це важливий аспект, оскільки регулярне оновлення бота не тільки покращує його функціональні можливості, але й забезпечує відповідність поточним потребам користувачів. Планування оновлень включає аналіз зворотного зв'язку від користувачів, тестування нових функцій та підготовку до їх впровадження.

Важливою складовою цього процесу є також забезпечення безперервної інтеграції та доставки (CI/CD), що дозволяє мінімізувати ризики під час впровадження оновлень.

Останній пункт розділу присвячено управлінню взаємодією з користувачами та обробці звернень. Цей аспект має ключове значення для підтримки позитивного іміджу бота та забезпечення його популярності серед користувачів. Важливим є створення системи обробки звернень, що дозволяє швидко і ефективно відповідати на питання та вирішувати проблеми користувачів. Використання спеціалізованих платформ для обробки звернень, таких як Zendesk чи Freshdesk, може значно підвищити ефективність цього процесу.

Підсумовуючи, можна зазначити, що ефективне управління та підтримка телеграм-ботів потребують комплексного підходу, що включає використання сучасних систем моніторингу та аналізу даних, планування оновлень і взаємодію з користувачами. Застосування цих принципів дозволяє забезпечити високу якість роботи телеграм-ботів, задоволеність користувачів та стійкий розвиток проєкту в цілому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jayesh Bairagi, Telegram Bot Development 101, 2019 URL: <https://medium.com/gammastack/telegram-bot-development-101-1f64edbff252>
2. Методологія та організація наукових досліджень : навч. посібник / О.І. Гупорів; Харк. нац. аграр. ун-т ім. В.В. Докучаєва – Х.: ХНАУ, 2017. – 272 с.
3. Методика та організація наукових досліджень : Навч. посіб. / С. Е. Важинський, Т. І. Щербак. – Суми: СумДПУ імені А. С. Макаренка, 2016. – 260 с.
4. Бірта Г. О. Методологія і організація наукових досліджень. [текст] : навч. посіб. / Г. О. Бірта, Ю.Г. Бургу– К. : «Центр учбової літератури», 2014. – 142 с.
5. Зацерковний В. І. Методологія наукових досліджень : навч. посіб. / В. І. Зацерковний, І. В. Тішаєв, В. К. Демидов. – Ніжин : НДУ ім. М. Гоголя, 2017. – 236 с.
6. Основи методології та організації наукових досліджень: Навч. посіб. для студентів, курсантів, аспірантів і ад'юнтів / за ред. А. Є. Конверського. — К.: Центр учбової літератури, 2010. — 352 с
7. While Web Production, 2023 URL: <https://whileweb.com/uk/blog/efektivne-upravlinnya-proektami-metodologiyi-ta-instrumenti/> (дата звернення: 12.02.2024)
8. Т. Осетрова, Еволюція розвитку та сучасні тренди в управлінні проектами, 2018 URL: <https://www.econa.org.ua/index.php/econa/article/download/1479/6565656603> (дата звернення: 12.02.2024)
9. Р. Олійник, Проектний менеджмент: особливості ключових концептів та застосування у публічному секторі, 2019 URL: http://www.pdu-journal.kpu.zp.ua/archive/2_2019/tom_2/24.pdf (дата звернення: 12.02.2024)
10. Діма Полежаєв, Технології керування проектами, 2023 URL: <https://lemon.school/blog/tehnologiyi-upravlinnya-proyektamy> (дата звернення: 12.02.2024)
11. Gareis R., Happy Projects [2019] URL: <https://gretchenrubin.com/wp-content/uploads/2017/09/The-Happiness-Project-Sample-Chapter.pdf> (дата звернення: 12.02.2024)

12. Астрід Тран, Що таке управління проєктами? [2023?] URL: <https://ahaslides.com/uk/blog/what-is-project-management/> (дата звернення: 12.02.2024)
13. Nurul Ashikin Abd Samad, Zalihar Embong, Maizatul Akhmar Mohamad Nor, “Development of Telegram Bot in Supporting the Learning Process of Object-Oriented Programming”, 2022 URL: https://www.researchgate.net/publication/363857573_Development_of_Telegram_Bot_in_Supporting_the_Learning_Process_of_Object-Oriented_Programming (дата звернення: 12.02.2024)
14. Nicolas Modrzyk, “Building Telegram Bots”, 2019 URL: 2 (дата звернення: 12.02.2024)
15. Darya I. Suntsova, Viktor A. Pavlov, Zinaida V. Makarenko, Petr P. Bakholdin, Alexander S. Politsinsky, Artem S. Kremlev, Alexey A. Margun, “Development of a Telegram Bot to Determine the Level of Technological Readiness”, 2022 URL: https://link.springer.com/chapter/10.1007/978-981-19-3444-5_9 (дата звернення: 12.02.2024)
16. Kevin Payne, “The Role of Chatbots in Project Management Strategy”, 2021 URL: <https://botsify.com/blog/the-role-of-chatbots-in-project-management-strategy/> (дата звернення: 12.02.2024)
17. Reggie Baker, “The 33 Best Project Management Blogs to Follow in 2024”, 2018 URL: <https://www.ntaskmanager.com/blog/best-project-management-blogs/> (дата звернення: 12.02.2024)
18. Jamina Nasrin Mohaideen, “A THEORETICAL LITERATURE REVIEW INVESTIGATION ON THE CRITICAL FACTORS FOR SUCCESSFUL AGILE PROJECT MANAGEMENT IN IT PROJECTS”, 2024 URL: https://www.researchgate.net/publication/347443708_A_THEORETICAL_LITERATURE_REVIEW_INVESTIGATION_ON_THE_CRITICAL_FACTORS_FOR_SUCCESSFUL_AGILE_PROJECT_MANAGEMENT_IN_IT_PROJECTS_JAMINA_NAS

- RIN_MOHAIDEEN_Research_Methods_and_Professional_Issues (дата звернення: 12.02.2024)
19. United Nations Development Programme, “Terms of Reference for the Programme Management Support Unit (PMSU)”, 2024 URL: <https://erc.undp.org/evaluation/managementresponses/keyaction/documents/download/331> (дата звернення: 02 Mar 2024)
20. Michaela Rollings, “18 Best Project Management Blogs & Influencers To Follow In 2024”, 2023 URL: <https://hive.com/blog/best-project-management-blogs/> (дата звернення: 12.02.2024)
21. Anya Brui, “How to Develop a Telegram Chatbot Builder and Grow to ...” [2022] URL: <https://sendpulse.com/blog/developing-telegram-bot-builder> (дата звернення: 13.02.2024)
22. Іванов О.В., Петрова К.С., "Стратегії управління проектами при розробці та підтримці Telegram-ботів", 2023 URL: https://gerabot.com/ru/article/prosuvannya_telegram_chatbotu_klyuchovi_aspekti_ta_efektivni_strategii (дата звернення: 13.02.2024)
23. node-telegram-bot-api - npm, 2024 URL: <https://core.telegram.org/bots/samples> (дата звернення: 13.02.2024)
24. Eightify, Build a Telegram Bot with Node JS: Step-by-Step Guide, 2024 URL: <https://eightify.app/summary/web-development/build-a-telegram-bot-with-node-js-step-by-step-guide> (дата звернення: 13.02.2024)
25. The Developer's Guide to Azure, 2024 URL: https://download.microsoft.com/download/2/C/F/2CF7401A-B9D7-4828-917D-199E0896BFE5/Azure_Developer_Guide_eBook.pdf (дата звернення: 13.02.2024)
26. Thomas Mitchell, Deploy Applications to a Windows Virtual Machine in Azure with the ..., 2024 URL: <https://thomasmitchell.net/wp-content/uploads/2018/05/Deploy-applications-to-a-Windows-virtual-machine-in-Azure-with-the-Custom-Script-Extension.pdf> (дата звернення: 13.02.2024)

- 27.R. Akhromieiev, Building a Telegram Bot with Azure Functions and Node.js - CodeProject, 2020 URL: <https://www.codeproject.com/Tips/5274291/Building-a-Telegram-Bot-with-Azure-Functions-and-N> (дата звернення: 13.02.2024)
- 28.Dmitry Rogozhny, Running Telegram Bot with Node.js and Azure Functions, 2019 URL: <https://dmitryrogozhny.com/blog/running-nodejs-telegram-bot-with-azure-functions> (дата звернення: 13.02.2024)
- 29.Morgan Kaufmann, Building the Agile Enterprise, 2009 URL: <http://ppdi.stmik-banjarbaru.ac.id/data.bc/12.%20Enterprise%20Architecture/2009%20Building%20the%20Agile%20Enterprise%20with%20SOA%2C%20BPM%20and%20MBM.pdf> (дата звернення: 13.02.2024)
- 30.Udara Abeythilake, Agile Methodology, 2022 URL: <https://medium.com/@abeythilakeudara3/agile-methodology-106270809c99> (дата звернення: 13.02.2024)
- 31.Світлана Васильченко, Що таке матриця Ейзенхауера та як її використовувати у роботі та житті, 2023 URL: <https://happymonday.ua/matrytsya-ejzenhauera> (дата звернення: 13.02.2024)
- 32.Jovin Mwendwa, Telegram Subscription Bot: How to scale your business or service on Telegram group or Channel, 2024 URL: <https://www.linkedin.com/pulse/telegram-subscription-bot-how-scale-your-business-service-mwendwa-idjve> (дата звернення: 13.02.2024)
- 33.Diane Wong, Unit Testing. What it is, How it Works, Types & Top Benefits, 2024 URL: <https://testsigma.com/blog/unit-testing/> (дата звернення: 13.02.2024)
- 34.Jorge Fernández Rodríguez, If You Want to Deliver Fast, Your Tests Have the Last Word, 2023 URL: <https://www.infoq.com/articles/delivering-fast-testing-strategies/> (дата звернення: 13.02.2024)
- 35.Coursera Staff, What Is Implementation Planning? And How to Write Your Plan, 2024 URL: <https://www.coursera.org/articles/implementation-planning> (дата звернення: 13.02.2024)

ДОДАТКИ

Додаток А

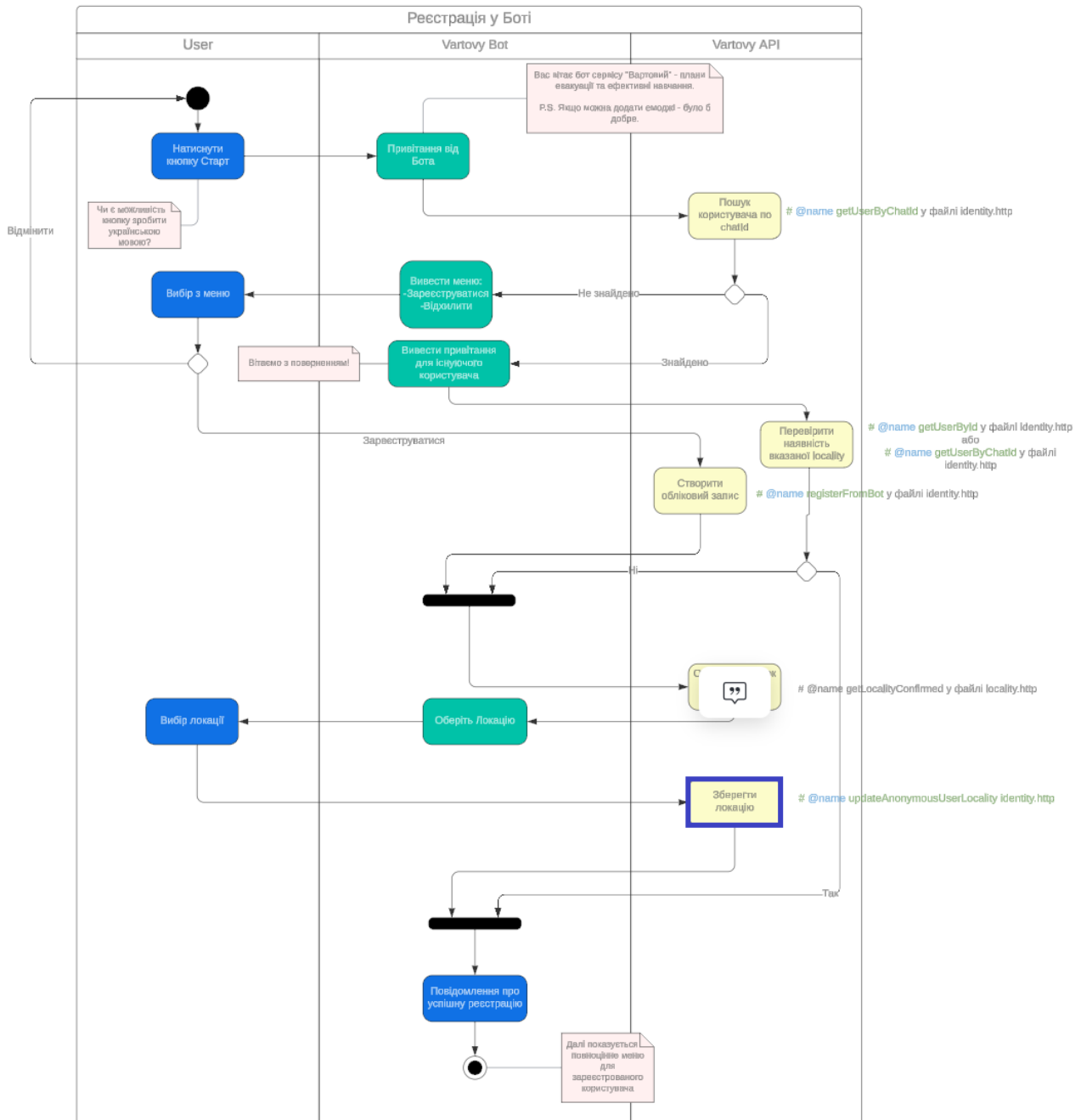


Рис. 1.10. UML-діаграма представлення процесу реєстрації у боті

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.

Додаток Б

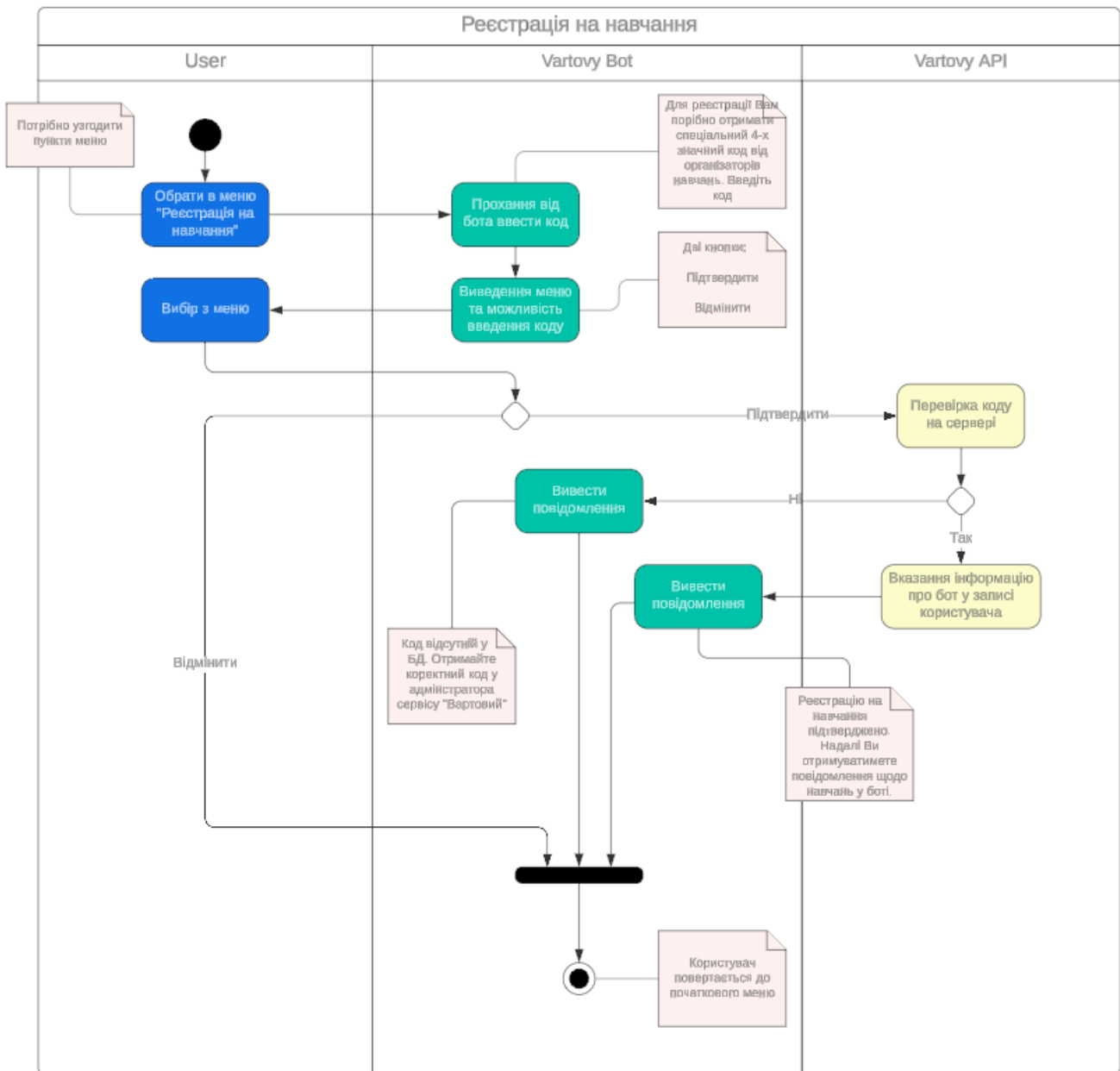


Рис. 1.11. UML-діаграма представлення процесу реєстрації на навчання

Джерело: розроблено автором на основі розробки проєкту "Вартовий" у 2024 р.