

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів інтеграції систем управління задачами та
комунікації для підвищення ефективності виконання ІТ-проектів

(тема)

Виконала:

здобувач 2 року навчання,
групи УПГІТМ-23-1

Аліна ЄМЕЛЬЯНОВА

(власне ім'я, прізвище)

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

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

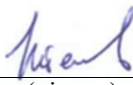
Освітня програма Управління проектами
в галузі ІТ

(повна назва освітньої програми)

Керівник: доц. каф. ІУС Віталій БРУСЕНЦЕВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ІУС


(підпис)

Костянтин ПЕТРОВ
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
 Кафедра _____ Інформаційних управляючих систем _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 122 Комп'ютерні науки _____
 (код і повна назва)
 Тип програми _____ освітньо-наукова _____
 (освітньо-професійна або освітньо-наукова)
 Освітня програма _____ Управління проектами в галузі інформаційних _____
 технологій _____
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

“ 21 “ квітня 20 25 р.

ЗАВДАННЯ**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві _____ Ємельяновій Аліні Валеріївні _____
 (прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів інтеграції систем управління задачами та для підвищення ефективності виконання ІТ-проектів _____

затверджена наказом по університету від “ 28 “ березня 2025 р. № 235Ст _____

2. Термін подання здобувачем роботи до екзаменаційної комісії “ 02 “ червня 2025 р. _____


3. Вихідні дані до роботи Науково-технічні публікації та дані Інтернет-джерел з тематики інтеграції систем управління задачами та комунікації. Результати опитування фахівців ІТ-галузі щодо особливостей використання систем управління задачами та комунікації. Технічна документація та API референси платформ Microsoft Teams, Jira. _____


4. Перелік питань, що потрібно опрацювати у роботі Аналіз сучасних підходів до управління ІТ-проектами та наявних інструментів для управління задачами та комунікації Дослідження існуючих методів інтеграції систем управління задачами та комунікації, включаючи API-інтеграцію, використання Webhooks та роботу з ботами. Аналіз результатів опитування учасників ІТ-команди щодо проблемних аспектів взаємодії між системами управління задачами та комунікації. Розробка моделі інтеграційного ішення на основі комунікаційного бота для обробки задач, надання доступу до документації та створення зустрічей. Проектування та реалізація інтеграційного рішення між системами управління задачами (Jira) та комунікації (Microsoft Teams). _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Змістовна постановка задачі призначення пакетів робіт командам виконавців ІТ-проєкту	21.04.2025 - 25.04.2025	Виконано
2	Аналіз сучасних підходів до управління ІТ-проєктами та наявних інструментів	26.04.2025 - 30.04.2025	Виконано
3	Дослідження існуючих методів інтеграції систем управління задачами та комунікації	01.05.2025 - 05.05.2025	Виконано
4	Проведення та аналіз результатів опитування фахівців ІТ-галузі	06.05.2025 - 08.05.2025	Виконано
5	Розробка моделі інтеграційного рішення між системами управління задачами та комунікації	09.05.2025 - 12.05.2025	Виконано
6	Проектування та реалізація інтеграції через додаток з каталогу Microsoft Teams	13.05.2025 - 15.05.2025	Виконано
7	Проектування та реалізація інтеграції через REST API та Webhook	06.05.2025 - 18.05.2025	Виконано
8	Розробка методики практичної перевірки запропонованих рішень	19.05.2025 - 21.05.2025	Виконано
9	Експериментальна перевірка та порівняльний аналіз ефективності розроблених методів інтеграції	22.05.2025 - 25.05.2025	Виконано
10	Оформлення пояснювальної записки	26.05.2025 - 29.05.2025	Виконано
11	Підготовка презентаційного матеріалу та перевірка роботи на плагіат	30.05.2025 - 02.06.2025	Виконано
12	Захист кваліфікаційної роботи	04.06.2025	Виконано

Дата видачі завдання 21 квітня 2025 р.

Здобувач 
(підпис)

Керівник роботи 
(підпис)

доц. каф.ІУС Віталій БРУСЕНЦЕВ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 83 сторінки, 24 рисунки, 8 таблиць, 2 додатки, 30 джерел.

ІНТЕГРАЦІЯ СИСТЕМ, КОМУНІКАЦІЯ, MICROSOFT TEAMS, JIRA, УПРАВЛІННЯ ЗАДАЧАМИ, WEBHOOK, API, ЕФЕКТИВНІСТЬ ІТ-ПРОЄКТІВ, МОНІТОРИНГ ЗАВДАНЬ, КОМАНДНА ВЗАЄМОДІЯ.

Об'єктом дослідження роботи є процес інтеграції систем управління задачами та платформ для командної комунікації при виконанні ІТ-проектів. Мета роботи – дослідження методів інтеграції систем управління задачами та комунікації, визначення їх впливу на ефективність виконання ІТ-проектів та розробка рекомендацій щодо оптимального впровадження таких інтеграційних рішень. Методи дослідження: порівняльний аналіз, емпіричне дослідження, статистичний аналіз, практичне тестування, анкетування. Результатами роботи є комплексний аналіз трьох методів інтеграції систем управління задачами та комунікації (API-інтеграція, Webhook-інтеграція та інтеграція через додаток з каталогу Microsoft Teams), визначення їх переваг, недоліків та оптимальних сфер застосування, а також розробка практичних рекомендацій щодо їх впровадження. Ефективність запропонованих рішень підтверджується результатами практичного тестування, які демонструють зменшення часу на перемикання між системами, збільшення частоти оновлення статусів завдань та зниження кількості випадків втрати контексту при обговоренні завдань. Практична значущість роботи полягає у розробці конкретних рекомендацій щодо вибору та впровадження інтеграційних рішень залежно від розміру команди, типу проекту та доступних технічних ресурсів. Запропоновані підходи можуть бути безпосередньо впроваджені в робочі процеси ІТ-команд для підвищення ефективності їх роботи.

ABSTRACT

Master's thesis: 83 pages, 24 figures, 2 tables, 2 appendices, 30 sources.

SYSTEM INTEGRATION, COMMUNICATION, MICROSOFT TEAMS, JIRA, TASK MANAGEMENT, WEBHOOK, API, IT PROJECT EFFICIENCY, TASK MONITORING, TEAM COLLABORATION.

The object of research is the process of integrating task management systems and team communication platforms in IT projects.

The purpose of the work is to study methods of integrating task management and communication systems, determine their impact on IT project efficiency, and develop recommendations for optimal implementation of such integration solutions.

Research methods: comparative analysis, empirical research, statistical analysis, practical testing, and surveying.

The results of the work include a comprehensive analysis of three methods for integrating task management and communication systems (API integration, Webhook integration, and integration through Microsoft Teams catalog application), identification of their advantages, disadvantages, and optimal application areas, as well as the development of practical recommendations for their implementation.

The effectiveness of the proposed solutions is confirmed by practical testing results, which demonstrate a reduction in time spent switching between systems, increased frequency of task status updates, and fewer instances of context loss during task discussions.

The practical significance of the work lies in developing specific recommendations for selecting and implementing integration solutions depending on team size, project type, and available technical resources.

ЗМІСТ

	С.
Скорочення та умовні позначки	8
Вступ	9
1 Аналіз предметної області та постановка задачі дослідження	11
1.1 Сучасні підходи до управління ІТ-проєктами	11
1.2 Інструменти постановки та контролю задач в ІТ-командах	12
1.2.1 Інші рішення для управління задачами: функціональні відмінності	12
1.2.2 Jira як платформа для управління робочим процесом	15
1.3 Платформи взаємодії та цифрової комунікації в ІТ-проєктах	17
1.3.1 Порівняння комунікаційних засобів (Microsoft Teams, Slack)	17
1.3.2 Microsoft Teams як середовище для командної взаємодії	19
1.4 Формулювання мети, завдань та обґрунтування актуальності дослідження	21
1.4.1 Аналіз результатів опитування учасників ІТ-команди	21
1.5 Постановка задачі дослідження	25
2 Методи інтеграції систем управління з системами комунікацій, їх прогнозований вплив на ефективність	27
2.1 Категорії інтеграційних рішень у цифрових середовищах	27
2.2 API як універсальний механізм взаємодії систем	28
2.3 Використання Webhook для реагування на події між системами	32
2.4 Інтеграція через додатки розширення	34
2.5 Порівняльний аналіз підходів: ефективність, масштабованість, застосовність	35
3 Практичне дослідження інтеграції систем управління задачами та комунікації	38
3.1 Проектування інтеграційного рішення для ІТ-команди	38

3.2	Покрокова реалізація інтеграції Jira з Microsoft Teams	40
3.2.1	Реалізація інтеграції через додаток з каталогу	40
3.2.2	Дослідження інтеграції через REST API	45
3.2.3	Дослідження інтеграції за допомогою Webhook	48
4	Аналіз результатів дослідження запропонованих методів інтеграції	52
4.1	Критерії ефективності з точки зору керування ІТ-командою	52
4.2	Методика практичної перевірки та оцінка впливу запропонованих рішень	55
4.3	Рекомендації до застосування і можливості розширення	57
	ВИСНОВКИ	61
	Перелік джерел посилання	63
	Додаток А Результати опитування	66
	Додаток Б Графічний матеріал кваліфікаційної роботи	73

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СУЗ – система управління задачами

API – Application Programming Interface – інтерфейс прикладного програмування

CI/CD – Continuous Integration/Continuous Deployment – безперервна інтеграція та безперервне розгортання

HTTP – Hypertext Transfer Protocol – протокол передачі гіпертексту

IT – Information Technology – інформаційні технології

JSON – JavaScript Object Notation – нотація об'єктів JavaScript

JWT – JSON Web Tokens – веб-токени JSON

OAuth – Open Authorization – відкритий протокол авторизації

REST – Representational State Transfer – передача репрезентативного стану

URL – Uniform Resource Locator – уніфікований локатор ресурсу

XML – eXtensible Markup Language – розширювана мова розмітки

ВСТУП

У сучасному світі цифрових технологій ефективне управління ІТ-проєктами стає ключовим фактором успіху компаній. Інструменти управління задачами та платформи для командної взаємодії відіграють вирішальну роль у плануванні та контролі виконання завдань.

Сьогодні ІТ-проєкти стають все складнішими, а команди постійно стикаються з розрізненістю інструментів: одні платформи використовуються для планування завдань (наприклад Jira), інші – для спілкування (як Microsoft Teams (MT) або Zoom). Ця роздробленість призводить до втрати часу, помилок і дублювання інформації. Приблизно 58% Agile-команд використовують Jira для управління проєктами, але при цьому 45% їх часу витрачається на ручну синхронізацію даних між системами [1]. Це яскраво демонструє, наскільки гострою є потреба в інтеграції інструментів.

Більшість ІТ-команд використовують системи управління задачами (СУЗ) та комунікації як окремі та не пов'язані між собою інструменти. Це створює розрив у робочому просторі, втрату важливої інформації та затримки в спілкуванні. Особливо гостро це відчувається при віддаленій роботі та використанні гнучких методологій (Agile, Scrum), коли важливо мати повну картину того, що відбувається з проєктом, і швидко реагувати на зміни.

Аналізуючи сучасні підходи, ми бачимо, що відсутність зв'язку між СУЗ (як Jira) та платформами для спілкування (як MT чи Zoom) робить проєктну діяльність менш прозорою, ускладнює контроль прогресу і знижує продуктивність команди.

Дослідження націлене на розробку ефективних способів інтеграції СУЗ та комунікації для покращення результативності ІТ-проєктів. Щоб досягти цієї мети заплановано:

- проаналізувати сучасні підходи до управління ІТ-проєктами та наявні інструменти для управління задачами та спілкування;

- дослідити існуючі методи інтеграції таких систем, включаючи інтеграцію через інтерфейс прикладного програмування (Application Programming Interface, API), використання Webhooks та роботу з ботами;
- розробити модель інтеграційного рішення на основі комунікаційного бота, який зможе обробляти задачі, надавати доступ до документації, створювати зустрічі та забезпечувати візуальну співпрацю;
- оцінити ефективність запропонованої моделі та розробити рекомендації з її використання.

Об'єктом дослідження є процеси управління та виконання завдань в проєктах інформаційних технологій (Information Technology, IT), а предметом дослідження – методи інтеграції СУЗ з інструментами комунікації.

Наукова новизна роботи полягає в розробці комплексного підходу до інтеграції СУЗ та комунікації, який враховує як технічні аспекти різних методів інтеграції, так і їх практичне застосування в контексті реальних IT-проєктів. Запропонований підхід до оцінки ефективності інтеграційних рішень дозволяє об'єктивно порівнювати різні методи та обирати оптимальний для конкретних умов.

Апробація результатів дослідження відбулася через публікацію наукової статті «Інтеграція систем управління задачами та комунікації як фактор підвищення ефективності виконання IT-проєктів» у збірнику матеріалів VII Міжнародної науково-практичної конференції [2]. Окремі результати практичного тестування інтеграційних рішень впроваджені в робочі процеси IT-компанії CodeIT, що підтверджує їх практичну значущість та ефективність.

Практична цінність дослідження визначається можливістю використання розроблених рекомендацій для підвищення ефективності виконання IT-проєктів. Запропоновані методи інтеграції дозволяють автоматизувати процеси контролю задач, скоротити час на щоденні зустрічі та ручну координацію, зробити виконання завдань більш прозорим та створити основу для ефективнішого управління проєктами в динамічному IT-середовищі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Сучасні підходи до управління ІТ-проєктами

Управління ІТ-проєктами в сучасному цифровому середовищі базується переважно на гнучких методологіях – Agile, Scrum, Kanban та їх гібридних варіаціях. Понад 71% ІТ-компаній в США використовують Agile-підходи як основну методологію управління проєктами [3]. Ці методології характеризуються короткими циклами розробки (спринти тривалістю 1-2 тижні), постійним зворотним зв'язком та високою адаптивністю до змін.

Ключовим елементом Scrum-методології є щоденні зустрічі (Daily Standups), які забезпечують синхронізацію в команді. Проте між 2020 та 2021 роками рівень впровадження Agile в командах розробки програмного забезпечення зріс з 37% до 86%, що призвело до збільшення тривалості щоденних зустрічей [4]. При збільшенні команди до 15+ осіб ефективність таких зустрічей знижується, оскільки втрачається фокус та зростає тривалість обговорень.

Сучасні ІТ-проєкти характеризуються розподіленими командами, віддаленою співпрацею та складними взаємозалежностями між задачами. В таких умовах критичного значення набувають інструменти управління проєктами та СУЗ з комунікаційними платформами, демонструють підвищення продуктивності команд на 47% [5].

Для структурування роботи використовуються спеціалізовані СУЗ, серед яких лідирує Jira з 42% ринку ІТ-проєктів [6]. За нею слідують Trello та Asana, які займають 14% та 11% ринку відповідно. Ці інструменти забезпечують планування спринтів, відстеження прогресу та агрегацію метрик проєкту.

Паралельно для забезпечення комунікацій застосовуються:

- Slack: платформа для миттєвого обміну повідомленнями;

- Microsoft Teams: інтегроване середовище для відеоконференцій та миттєвого обміну повідомленнями;

- Zoom: сервіс для проведення відеозустрічей [7].

Проблема полягає в тому, що ці системи функціонують як окремі «острови інформації». Команди витрачають 45% робочого часу на ручну синхронізацію даних між системами [8]. Інформація про статуси задач, коментарі та рішення розподіляється між різними платформами, що створює «інформаційні розриви» та знижує прозорість проектної діяльності.

Саме тому інтеграція СУЗ з платформами комунікації стає критичним фактором для підвищення ефективності ІТ-проектів. Використання інтегрованих платформ дозволяє скоротити час на перемикання між інструментами на 32% та підвищити швидкість виведення продуктів на ринок на 23% [5]. Такий підхід дозволяє:

- зменшити час на переключення між інструментами;
- забезпечити актуальність даних у всіх системах;
- створити єдиний контекст для прийняття рішень;
- автоматизувати рутинні процеси моніторингу та звітності.

1.2 Інструменти постановки та контролю задач в ІТ-командах

1.2.1 Інші рішення для управління задачами: функціональні відмінності

Ринок СУЗ включає різноманітні інструменти з відмінними функціональними можливостями та підходами до організації проектної роботи. Jira, розроблена компанією Atlassian, займає провідну позицію на ринку з часткою 42%, що робить її найпопулярнішим рішенням для ІТ-команд. Microsoft Project (20%), Trello (14%), Asana (11%) та Monday.com (6%) формують основну конкуренцію на ринку [6].

На рисунку 1.1 зображено частку інструментів управління задачами на ринку.

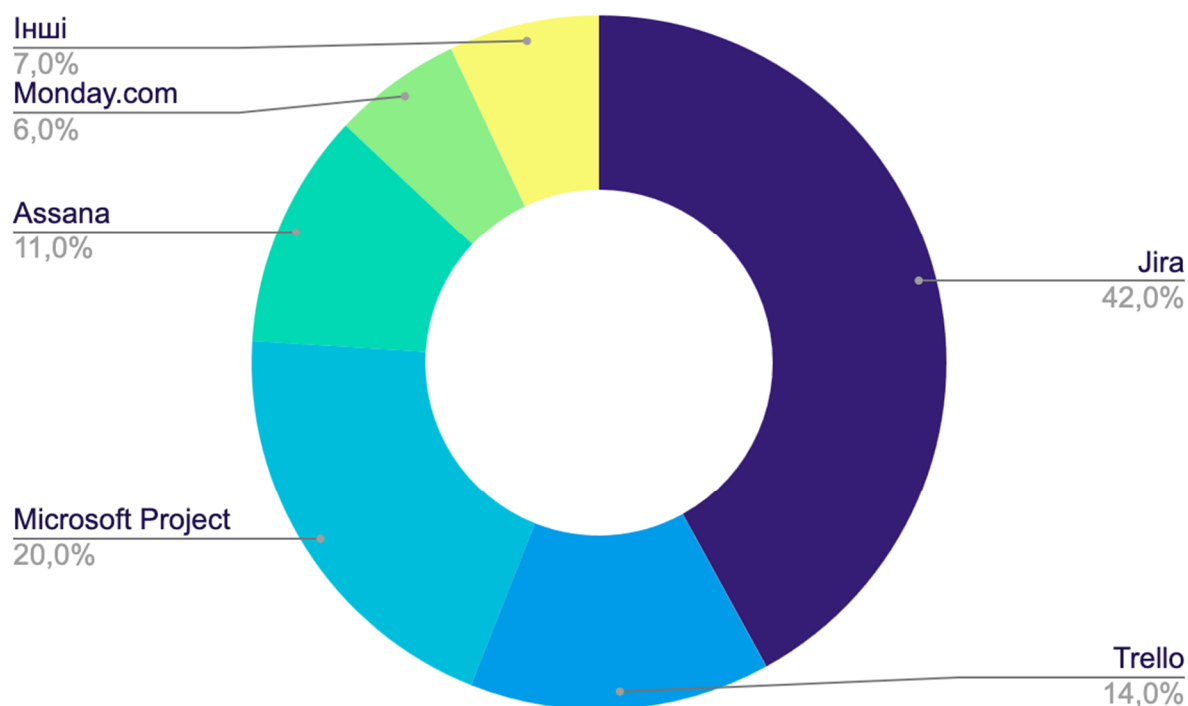


Рисунок 1.1 – Частка інструментів управління задачами на ринку

Trello представляє собою інструмент, побудований на принципах Kanban-методології з інтуїтивно зрозумілим візуальним інтерфейсом. Його головною перевагою є низький поріг входження – користувачам не потрібна спеціальна підготовка для початку роботи. Система дошок з картками дозволяє наочно відстежувати прогрес задач і швидко перерозподіляти роботу між членами команди. Основною цільовою аудиторією Trello є маркетингові, творчі та невеликі команди, для яких важлива прозорість процесів і візуальне представлення задач, але не потрібна складна аналітика чи гнучке управління спринтами.

Asana пропонує збалансований підхід до управління проєктами, поєднуючи дружній інтерфейс з розширеними можливостями для планування та відстеження роботи. Унікальною особливістю Asana є різноманітність

представлень проєкту – користувачі можуть перемикатися між списками, дошками, календарем та часовими шкалами залежно від поточних потреб. Платформа підтримує налаштування взаємозалежностей між завданнями, що дозволяє ефективно управляти складними проєктами з розгалуженою структурою задач. Інструмент орієнтований на нетехнічні команди, які потребують більшої гнучкості, ніж пропонує Trello, але не вимагають всієї потужності Jira.

Monday.com відрізняється від конкурентів високою гнучкістю настроювання робочих процесів та яскравим, інтуїтивним інтерфейсом. Система пропонує настроювані таблиці з кольоровим кодуванням статусів, що спрощує візуальне сприйняття прогресу проєкту. Основною перевагою Monday.com є можливість адаптації під різні типи робочих процесів – від маркетингових кампаній до розробки програмного забезпечення. Платформа пропонує численні інтеграції з іншими бізнес-інструментами, хоча їх кількість поступається Jira.

З погляду інтеграції з комунікаційними платформами, Jira пропонує найширші можливості завдяки розвиненій системі API та понад 3000 готових конекторів, що дозволяє безперешкодно підключатися до Slack та інших засобів спілкування. Інші інструменти також підтримують інтеграцію зі Slack, проте з меншими можливостями автоматизації робочих процесів між платформами. Ця різниця стає критичною для команд, що працюють над складними проєктами з багатьма взаємозалежними компонентами.

Ключовою відмінністю між інструментами є їхня спеціалізація та складність. Jira найкраще підходить для технічних команд, що працюють за Agile-методологіями, з глибокими можливостями налаштування робочих процесів та звітності. Trello оптимальний для невеликих команд з простими процесами, Monday.com та Asana займають проміжну позицію, пропонуючи розширені можливості при збереженні інтуїтивного інтерфейсу.

Вибір оптимального інструменту управління задачами залежить від конкретних потреб проєкту та команди. Для забезпечення максимальної

ефективності важливо не лише оцінювати функціональність самої системи, але й можливості її інтеграції з комунікаційними платформами, що дозволяє створити єдиний інформаційний простір для всіх процесів ІТ-проєкту.

1.2.2 Jira як платформа для управління робочим процесом

Jira займає провідне місце на ринку СУЗ в ІТ-сфері. Сьогодні вона є лідером серед подібних систем з часткою ринку 42% [6]. Розроблена компанією Atlassian, ця платформа пропонує все необхідне для планування, відстеження та аналізу роботи в межах проєктів. Понад 65% команд що працюють за гнучкими методологіями, використовують саме Jira для організації своїх робочих процесів [1].

Окремої уваги заслуговує аналітика в Jira. Інструменти на кшталт Burndown Chart, Velocity Chart чи Sprint Report дають змогу не просто бачити, як просувається робота, а й оцінювати продуктивність команди та прогнозувати завершення проєкту. Команди, які активно користуються цими звітами, на 27% частіше вкладаються у дедлайни [6].

Ще одна перевага Jira це її гнучкість. Платформа дозволяє налаштовувати робочі процеси відповідно до обраної методології: Scrum, Kanban або індивідуального підходу. Користувачі можуть задавати власні статуси задач і автоматизувати переходи між ними. Інтерактивні дошки задач наочно відображають прогрес, а система фільтрів полегшує навігацію в складних проєктах. Кожен етап задачі можна адаптувати, додати власні статуси та автоматизувати переходи. Завдяки інтерактивним дошкам команди мають чітке уявлення про те, на якому етапі перебуває кожна задача, а система фільтрів полегшує пошук потрібної інформації. На рисунку 1.2 наведено приклад дошки спринта з завданнями в Jira.

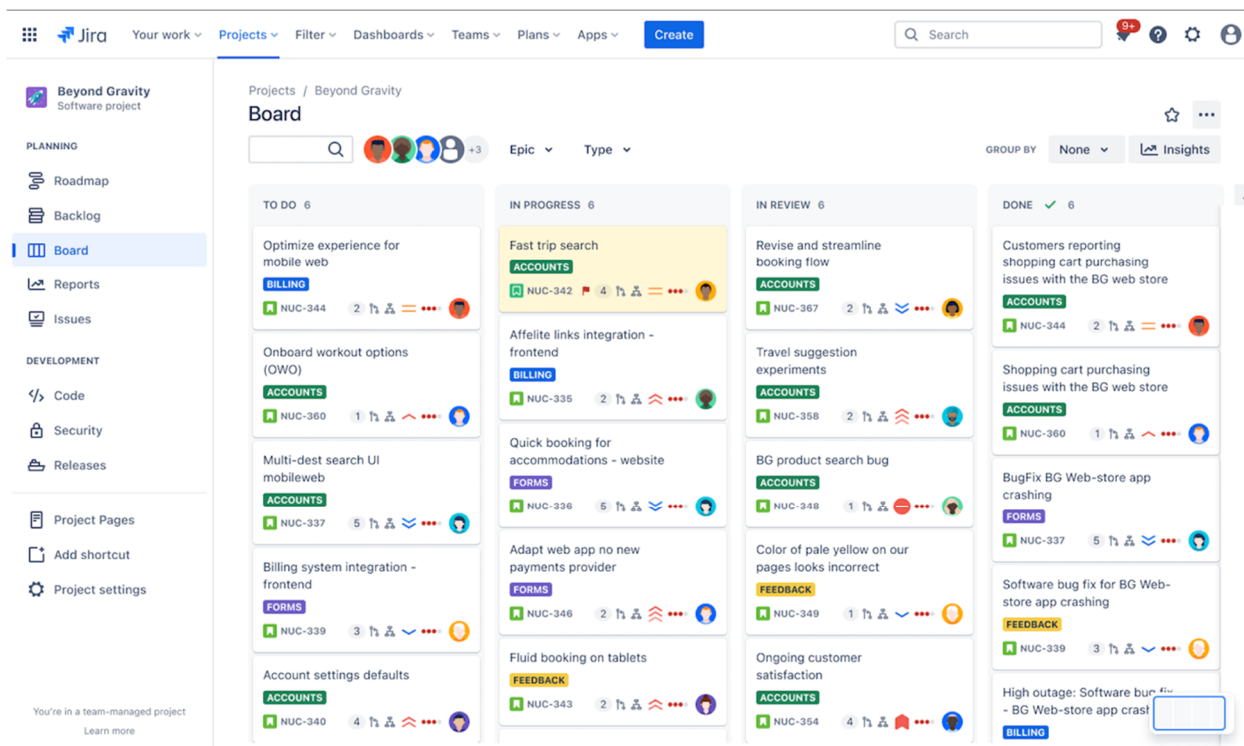


Рисунок 1.2 – Дошка завдань спринту у Jira

Проте як і будь-який інструмент, Jira має свої обмеження. Зокрема у сфері командної комунікації. Вона не є зручним інструментом для швидкого обговорення задач. Хоча платформа й підтримує коментарі, справжню ефективність команда досягає лише тоді, коли поєднує Jira з чатами на кшталт Slack чи MT.

Проблема «інформаційних розривів» між СУЗ та комунікації особливо гостро відчувається в розподілених командах, де синхронізація інформації критично важлива. Близько 69% працівників витрачають до години щодня на перемикання між комунікаційними програмами та пристроями, що призводить до втрати до 32 робочих днів на рік для кожного члена команди [9].

Саме тому інтеграція Jira з комунікаційними інструментами стає необхідним компонентом сучасних ІТ-проектів. Компанії, які інтегрують СУЗ з комунікаційними платформами, демонструють підвищення продуктивності команд на 23% [1]. Такий підхід дозволяє створити єдиний інформаційний простір, забезпечуючи повну прозорість процесів та підвищуючи ефективність командної роботи.

1.3 Платформи взаємодії та цифрової комунікації в IT-проектах

1.3.1 Порівняння комунікаційних засобів (Microsoft Teams, Slack)

В сучасному IT-середовищі існують різноманітні популярні платформи для командної комунікації та співпраці, серед яких особливу увагу заслуговують Microsoft Teams та Slack. Кожна з цих платформ має свої особливості, переваги та недоліки в контексті інтеграції з СУЗ та підтримки процесів розробки програмного забезпечення.

Microsoft Teams позиціонується як комплексне рішення для командної роботи, що тісно інтегроване з екосистемою Microsoft 365. Платформа пропонує потужні можливості для інтеграції з СУЗ через вбудований функціонал та розширення. Teams має нативну інтеграцію з Microsoft Planner та Azure DevOps, що дозволяє командам синхронізувати робочі процеси безпосередньо з комунікаційним середовищем [10]. Станом на 2025 рік Microsoft Teams демонструє значне зростання, досягнувши понад 320 мільйонів користувачів, що на 7% більше порівняно з 2022 роком [8].

Slack, з іншого боку, виник як платформа для корпоративного спілкування і здобув популярність завдяки своїй гнучкості та інтуїтивно зрозумілому інтерфейсу. Однак, з точки зору інтеграції з корпоративними системами та розширеними можливостями для командної роботи, Slack має певні обмеження порівняно з Microsoft Teams. Хоча Slack пропонує більшу кількість доступних інтеграцій з різноманітними сервісами, ці інтеграції часто не досягають такого рівня глибини та функціональності, який забезпечує екосистема Microsoft.

Порівняння цих платформ з точки зору ефективності інтеграції з системами управління задачами наведено у таблиці 1.1:

Таблиця 1.1 – Порівняння комунікаційних платформ для ІТ-проектів

Характеристика	Microsoft Teams	Slack
Інтеграція з Jira	Нативна інтеграція через офіційний додаток	Глибока двостороння інтеграція
Підтримка відеоконференцій	До 1000 учасників	До 15 учасників
Кількість доступних інтеграцій	Понад 700 додатків	Понад 2400 додатків
API для кастомних інтеграцій	Розвинений API	Розвинений API
Автоматизація процесів	Через Power Automate	Через Workflows
Вартість базового плану	\$4	\$7.25

Хоча Slack пропонує більшу кількість готових інтеграцій Microsoft Teams завдяки тісному зв'язку з продуктами екосистеми Microsoft і потужному API, забезпечує більш глибоку і функціональну інтеграцію з корпоративними системами [11]. Особливо це стосується інтеграції з Jira, де Microsoft Teams пропонує не лише перегляд задач, але й можливість їх створення, редагування та управління безпосередньо з інтерфейсу Teams.

У контексті реалізації ІТ-проектів, інтеграція інструментів комунікації з СУЗ стає вирішальною умовою для ефективного моніторингу та контролю виконання робіт. Впровадження таких інтегрованих рішень дозволяє зменшити час на пошук інформації на 35%, а продуктивність команд зростає на 20–25% [6]. Це свідчить про те що для досягнення високої ефективності важливий не лише вибір окремих сервісів, а й налагоджена взаємодія між ними, де Microsoft Teams має значну перевагу завдяки своїй інтеграції з корпоративною екосистемою та розширеними можливостями для командної роботи.

1.3.2 Microsoft Teams як середовище для командної взаємодії

Microsoft Teams є ключовим інструментом у сфері корпоративних комунікацій, особливо для IT-команд, що потребують комплексного інструменту для взаємодії. Станом на 2025 рік платформа демонструє значне зростання, досягнувши понад 320 мільйонів користувачів, що на 7% більше порівняно з 2022 роком [8]. Така популярність пояснюється не лише потужним функціоналом, але й тісною інтеграцією з екосистемою Microsoft 365, що робить Teams привабливим вибором для організацій будь-якого розміру.

Платформа Microsoft Teams виділяється на фоні конкурентів своєю комплексністю, поєднуючи різні аспекти командної роботи в єдиному інтерфейсі. На відміну від багатьох аналогів Teams пропонує не просто комунікаційний інструмент а повноцінне середовище для спільної роботи, що включає спілкування, проведення нарад, спільний доступ до документів та інтеграцію з численними сервісами. Це особливо важливо в контексті сучасних тенденцій до гібридної роботи, коли команди працюють розподілено чи взагалі віддалено.

Потужна екосистема інтеграцій є одним із найвагоміших аспектів Microsoft Teams. У магазині додатків Teams доступно понад 1,400 інтеграцій з різними бізнес-інструментами. Кількість доступних інтеграцій зростає кожного року, з найбільшою кількістю додатків у категоріях «Продуктивність» та «Управління проектами». Ця розгалужена мережа інтеграцій дозволяє командам налаштовувати робоче середовище відповідно до специфіки їхніх проектів та галузевих вимог.

Ключовою перевагою Microsoft Teams є об'єднання всіх необхідних функцій в одному місці. Замість перемикання між різними додатками, користувачі отримують доступ до чату, відеодзвінків, файлів та інших інструментів через єдиний інтерфейс. Це значно спрощує комунікацію та робить її більш природною. На рисунку 1.3 зображено інтерфейс групового

чату в Microsoft Teams, який демонструє інтуїтивно зрозумілий дизайн платформи.

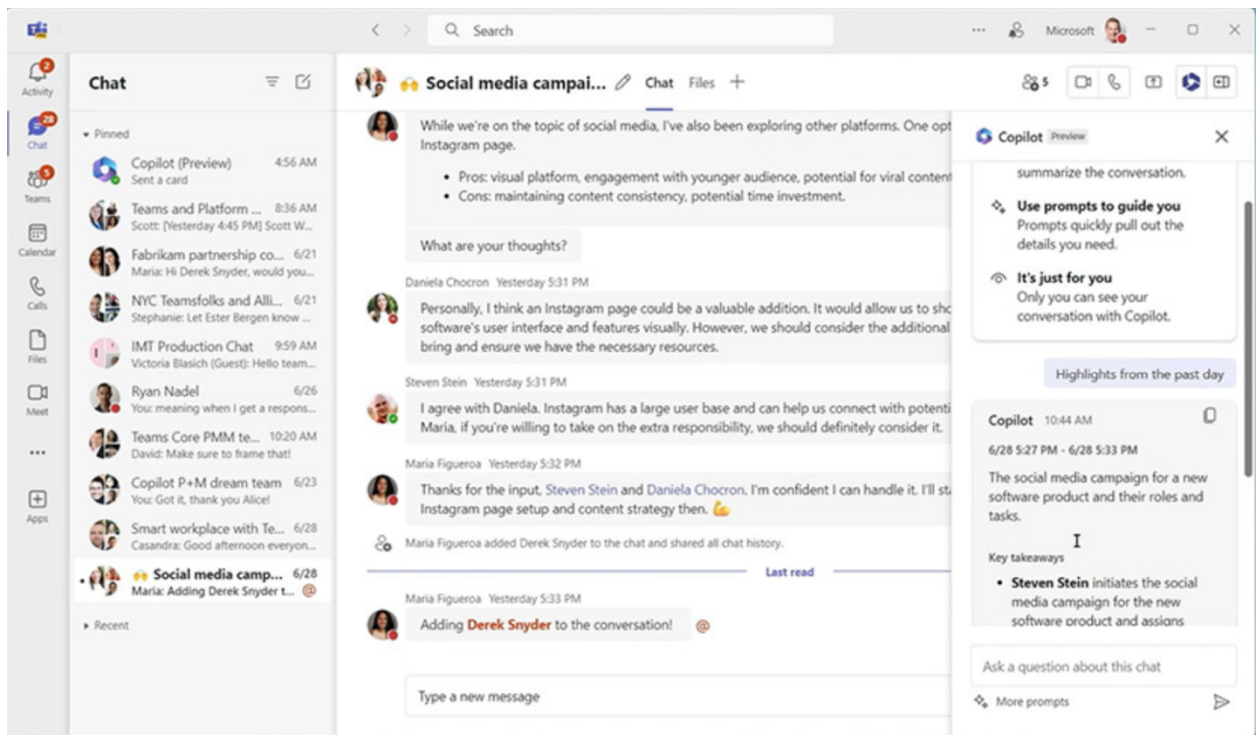


Рисунок 1.3 – Інтерфейс групового чату в Microsoft Teams

Особливо корисною для ІТ-команд є інтеграція Teams з інструментами управління проєктами, включаючи Jira. Команди, які використовують інтеграцію Teams-Jira, повідомляють про скорочення часу на перемикання між різними інструментами на 31% та підвищення ефективності відстеження задач на 24% [12]. Ця інтеграція дозволяє отримувати сповіщення про зміни статусів завдань, створювати нові задачі безпосередньо з повідомлень та швидко переглядати деталі задач, не виходячи з Teams. Для технічних команд це особливо цінно, оскільки дозволяє зменшити кількість перемикань контексту та зосередитись на вирішенні завдань.

Microsoft Teams став зручним і надійним інструментом для командної роботи, особливо в умовах віддаленого та гібридного формату. Об'єднуючи чати, дзвінки, спільні документи та інтеграції з іншими сервісами, такими як Jira, ця платформа допомагає командам ефективно спілкуватися та

координувати свої дії. Завдяки простому інтерфейсу та широким можливостям налаштування Teams добре підходить для щоденної роботи над проєктами та сприяє кращій організації робочого процесу.

1.4 Формулювання мети, завдань та обґрунтування актуальності дослідження

1.4.1 Аналіз результатів опитування учасників ІТ-команди

Інтеграція різних ІТ-систем у межах одного проєктного середовища супроводжується низкою викликів, що безпосередньо впливають на ефективність командної роботи та реалізацію проєктів. Для глибшого розуміння цих проблем було проведено опитування серед фахівців української ІТ-компанії CodeIT. Дослідження охопило 31 респондента, серед яких – 6 проєктних менеджерів, 22 розробники програмного забезпечення та 3 дизайнери.

Усі учасники працюють на проєктах, де використовуються Jira як система управління задачами, Microsoft Teams як основний канал комунікації. Опитування відбувалося у квітні 2025 року в онлайн-форматі та було спрямоване на виявлення ключових труднощів, з якими стикаються ІТ-команди під час використання кількох систем для управління задачами, комунікації та щоденної взаємодії. Учасники оцінювали зручність існуючих інтеграцій, а також потребу у нових функціональних рішеннях (рисунки А.1 – А.12). Особлива увага приділялася зв'язці Jira та Microsoft Teams – один з найпоширеніших варіантів інтеграції у сучасному проєктному середовищі. Метою було виявити ключові проблеми взаємодії між системами та оцінити потенційні переваги інтеграційних рішень.

Таблиця 1.4 – Анкета опитування щодо інтеграції систем управління задачами та комунікації

№	Питання	Варіант відповіді
1	2	3
1	Скільки учасників у вашій проєктній команді?	5–10
		11–15
		16–20
		Більше 20
2	Чи використовуєте ви інтеграційні рішення між Jira та Microsoft Teams?	Так
		Ні
3	Чи задовольняє вас поточний процес моніторингу виконання завдань на проєкті?	Повністю задовольняє
		Частково задовольняє
		Не задовольняє
4	Як часто ви забуваєте оновлювати статуси завдань у Jira?	Дуже часто
		Час від часу
		Рідко
		Ніколи
5	Чи вважали б ви корисним мати бота, який дозволяє оновлювати статуси завдань зі Microsoft Teams?	Так
		Ні
		Важко сказати
6	Скільки часу ви витрачаєте щодня на перемикання між Jira та Microsoft Teams?	До 15 хв.
		15–30 хв.
		30–60 хв.
		Більше 60 хв.

Кінець таблиці 1.4

1	2	3
7	Чи втрачаєте ви контекст під час перемикання між Jira та Microsoft Teams?	Так, постійно
		Так, інколи
		Рідко
		Ніколи
8	Які проблеми ви бачите у взаємодії між Jira та Microsoft Teams?	Неможливість оновлювати статус із Microsoft Teams
		Відсутність контексту обговорень у Jira
		Необхідність дублювати інформацію
9	Чи хотіли б ви отримувати сповіщення у Microsoft Teams про зміни в задачах, які вам призначені?	Так
		Ні
		Лише для пріоритетних завдань
10	Чи вважаєте ви, що час на перемикання між різними системами впливає на вашу загальну продуктивність?	Дуже суттєво впливає
		Помірно впливає
		Незначно впливає
		Не впливає
11	Який найчастіший сценарій, коли вам необхідно перемикатися між Microsoft Teams та Jira?	Перевірка статусу завдання
		Оновлення інформації
		Комунікація з командою про завдання
12	Яка головна проблема при роботі з розрізненими системами?	Втрата часу
		Втрата інформації
		Дублювання дій
		Інше (вказіть)

Результати опитування (рисунки А.1 – А.12) свідчать про низку

критичних обмежень у поточній інтеграції між Jira та Microsoft Teams. Зокрема, 42% респондентів вказали, що інтеграція між цими інструментами або відсутня, або не відповідає потребам команди. При цьому 13 опитаних (42%) відзначили як головну проблему неможливість оновлювати статуси задач безпосередньо з Microsoft Teams. Це змушує розробників постійно перемикатися між системами, що забирає час і знижує концентрацію.

72% учасників витрачають на перемикання між Jira та Microsoft Teams від 15 до 30 хвилин щодня, а ще 5 – понад півгодини. Більше того, 54% респондентів регулярно втрачають контекст під час перемикань. Ці дані узгоджуються з оцінкою впливу: 14 з 31 опитаного вважають, що перемикання суттєво впливає на продуктивність, ще 10 – що помірно.

Коли учасників запитали, у яких випадках їм найчастіше доводиться перемикатися між Jira та Teams, 25 із них назвали перевірку статусу та оновлення задач – сценарії, які можуть бути автоматизовані.

Також 67% розробників повідомили, що періодично забувають оновлювати статус задач у Jira, що ускладнює моніторинг і планування. У зв'язку з цим 27 з 31 респондента висловили бажання мати інтеграційного бота, який дозволяв би оновлювати статуси задач безпосередньо з Microsoft Teams. Ще 29 учасників підтримали ідею автоматичних сповіщень про зміни у задачах, що значно покращило б реактивність і командну синхронізацію.

Незважаючи на формальну наявність базових інтеграцій між Jira та Microsoft Teams, їхній поточний функціонал не забезпечує справжньої зручності. Ключові функції — оновлення статусів, фіксація обговорень, отримання сповіщень – залишаються ручними або фрагментарними. Це підтверджує потребу у створенні спеціального рішення – бота або інтеграційного модуля для синхронізації дій без виходу з Microsoft Teams.

Таке рішення здатне не лише зменшити кількість перемикань між інструментами, а й підвищити продуктивність, точність статусів задач, зменшити навантаження на команди та покращити прозорість контролю.

1.5 Постановка задачі дослідження

Метою роботи є дослідження існуючих методів інтеграції систем управління задачами та комунікації для підвищення ефективності виконання ІТ-проектів. Для виконання цієї роботи необхідно виконання наступних важливих складових дослідження:

- проведення детального аналізу предметної області;
- вивчення методів інтеграції систем управління з системами комунікацій;
- розрахунок їх прогнозованого впливу на ефективність виконання роботи;
- створення експериментальної інтеграції та вимірювання її впливу на ефективність;
- проведення аналізу та визначення оцінки результатів інтеграції.

Визначення постановки задачі та виконання всіх її складових зазначених вище є критичним етапом цієї роботи, оскільки він задає напрямок та результативність дослідження. Це надає чіткість цілей та завдань дослідження за рахунок окреслення, що саме потрібно дослідити, яку проблему потрібно вирішити та які питання слід вивчати, що також забезпечує структурований більш підхід.

Додатково така структура виконання допомагає впевнитись в ефективності використання ресурсів завдяки уникненню витрати часу та ресурсів на нерелевантні аспекти дослідження, а також додає певну методологічну точність дослідницької задачі через чіткий вибір відповідних методів аналізу, експериментів і збору даних, додаючи підґрунтя та об'єктивність отриманих результатів.

Вивчення існуючих методів інтеграції систем управління та комунікацій дає змогу оцінити їх переваги й недоліки, розробити рекомендації для оптимального налаштування інтеграції, а також прогнозувати можливі

покращення у взаємодії інструментів. Інтеграція дозволяє автоматизувати передачу інформації між системами, усуваючи потребу в ручному оновленні, зменшуючи ризик помилок та покращуючи бізнес процеси. Дослідження інтеграції допоможе визначити найкращі практики та методи, що дозволяють впроваджувати інтеграцію з найменшими витратами ресурсів та максимальним позитивним впливом на робочий процес.

Створення експериментальної інтеграції та вимірювання її впливу на ефективність виконання роботи допоможе не лише оцінити важливість інтеграції, але й глибше зрозуміти її вплив на динаміку роботи команди, якість виконання проєктів та довгострокові переваги. Під час такого експериментального можна виявити проблеми сумісності між системами, затримки в обміні даними та інші технічні бар'єри, які заважають оптимальному впровадженню.

Аналіз та оцінка є фундаментальними для обґрунтованого прийняття рішень щодо впровадження або модифікації інтеграційних підходів у сфері управління ІТ-проєктами. Цей етап також дозволяє порівняти ефективність інтегрованих систем з їхніми окремими компонентами, а також визначити, чи покращує інтеграція управління задачами та комунікацію в ІТ-проєктах. На основі результатів можна виробити корисні висновки та рекомендації щодо правильного застосування та впровадження інтеграцій.

2 МЕТОДИ ІНТЕГРАЦІЇ СИСТЕМ УПРАВЛІННЯ З СИСТЕМАМИ КОМУНІКАЦІЙ, ЇХ ПРОГНОЗОВАНИЙ ВПЛИВ НА ЕФЕКТИВНІСТЬ

2.1 Категорії інтеграційних рішень у цифрових середовищах

У сучасних ІТ-проєктах технологічна інтеграція відіграє ключову роль, забезпечуючи ефективне з'єднання різних цифрових систем, сервісів і платформ. Вона дозволяє автоматизувати процеси, зменшити ручну роботу та підвищити продуктивність команд, працюючи як міст між інструментами управління задачами та комунікаційними платформами.

Впровадження інтеграцій між різними платформами та програмними застосунками може надати значні переваги та підвищити ефективність виконання роботи. До основних позитивних аспектів можна віднести:

- покращення оперативності комунікації за допомогою автоматичного надсилання оновлень між системами;
- автоматизація процесів та зменшення необхідності ручного введення даних. Наприклад, зміна статусу задачі у Jira автоматично сповіщає команду в Microsoft Teams;
- підвищена актуальність даних через можливість двосторонніх синхронізованих оновлень (наприклад, зміна статусу задачі в Microsoft Teams буде відображена в Jira і навпаки);
- спрощене управління проєктами через централізований контроль над завданнями та їхнім прогресом, що також дозволяє більш ефективно розподіляти ресурси;
- більша ефективність та зручність для учасників команди, оскільки немає потреби перемикатися між платформами завдяки тому, що інтегровані системи(наприклад, система управління задачами), дозволяють взаємодіяти зі своїми компонентами завданнями безпосередньо в інших програмних застосунках, зокрема Microsoft Teams.

Тим не менш, будь-яка технологічна інтеграція, зв'язок Jira з іншими

системами має свої недоліки, зокрема складність налаштування (деякі плагіни потребують складної конфігурації, яка може бути незрозумілою для користувачів без технічного досвіду); обмеження функціональності (інтеграція може не підтримувати всі можливості, які доступні у рідних інструментах Jira або інших сервісах); продуктивність (занадто висока кількість інтегрованих плагінів може сповільнювати роботу системи, особливо при обробці великого обсягу даних); безпека даних (певні інтеграції передають дані між системами, і це може створювати ризики, якщо немає достатнього рівня захисту або доступу).

2.2 API як універсальний механізм взаємодії систем

Одним з найбільш поширених підходів є використання API (Application Programming Interface), що дозволяє різним програмним рішенням взаємодіяти безпосередньо. API-інтеграція дає змогу автоматизувати передачу даних між системами, об'єднуючи функціональність різних платформ, зокрема таких як, системи управління задачами та комунікації.

API (або «Application Programming Interface») був створений як ключовий механізм інтеграції між різними програмними системами та сервісами, а його головна мета це забезпечення стандартизованого способу взаємодії між різними платформами, щоб вони могли ефективно обмінюватися даними та функціональністю. Тож API представляє собою набір правил та механізмів, які дозволяють одній програмі взаємодіяти з іншою. Такий підхід дозволяє обмінюватися даними та виконувати функції без необхідності надання прямого доступу до внутрішнього коду кожної з них. Механізми роботи API будуються на таких основних принципах як: комунікація за допомогою запиту та відповіді API (одна система надсилає запити, наприклад, щоб отримати дані, а інша система обробляє цей запит і повертає відповідь);

визначений стандартизований формат обміну даними, який легко читається як машинами, так і людьми, наприклад, нотація об'єктів JavaScript (JavaScript Object Notation, JSON).

Масштабованість та автоматизація API дозволяє автоматизувати процеси між сервісами, забезпечуючи швидку інтеграцію без ручного втручання. Це особливо корисно у великих цифрових екосистемах.

Тим не менш, API мають багато переваг, але також і певні недоліки, зокрема це: безпека, оскільки API можуть бути вразливими до атак, якщо не належним чином захищені; залежність від додаткових сторонніх сервісів, оскільки API може припинити роботу у разі зміни роботи сервісів які він поєднує; обмеженість функціоналу, розробка якого напряму залежить від розробників; складність інтеграції, оскільки деякі API можуть бути складними у використанні через погану документацію або складні вимоги; продуктивність, у випадках коли виклики API можуть уповільнювати роботу системи, особливо якщо вони потребують великої кількості запитів.

Для реалізації необхідно було б визначити систему-посередник, яка буде дізнатися про оновлення в СУЗ та в разі ідентифікації події оновлення, відправляє запит через API до системи комунікацій. Весь цей процес відбувається майже миттєво та автоматично. Такий підхід дозволяє розробнику зберігати високу гнучкість і розробці функціоналу, оскільки реалізація повністю залежить від його рішень та дозволяє власноруч розширити існуючий функціонал з часом.

У контексті інтеграції СУЗ з платформами комунікації, API на основі передачі репрезентативного стану (Representational State Transfer, REST) є найпопулярнішим архітектурним стилем через свою простоту та універсальність. Jira Cloud API надає понад 200 різних ендпоінтів для роботи з проєктами, задачами та користувачами. Microsoft Graph API, у свою чергу, забезпечує доступ до всіх сервісів Microsoft 365, включаючи Teams, через єдину точку входу. Такий підхід дозволяє створювати складні інтеграційні сценарії, де зміна статусу задачі в Jira може автоматично запускати створення

повідомлення в Teams каналі або навіть планувати відеоконференцію для обговорення критичних завдань.

Ефективність API-інтеграції значною мірою залежить від правильної стратегії керування запитами та оптимізації мережевого трафіку. Провідні практики включають використання пакетних запитів для зменшення навантаження на сервери, впровадження кешування для часто запитуваної інформації та асинхронну обробку операцій, які не потребують миттєвої відповіді.

Приклад впровадження інтеграції між системою управління задачами та системою комунікацій з використанням API наведений в рисунку 2.1.

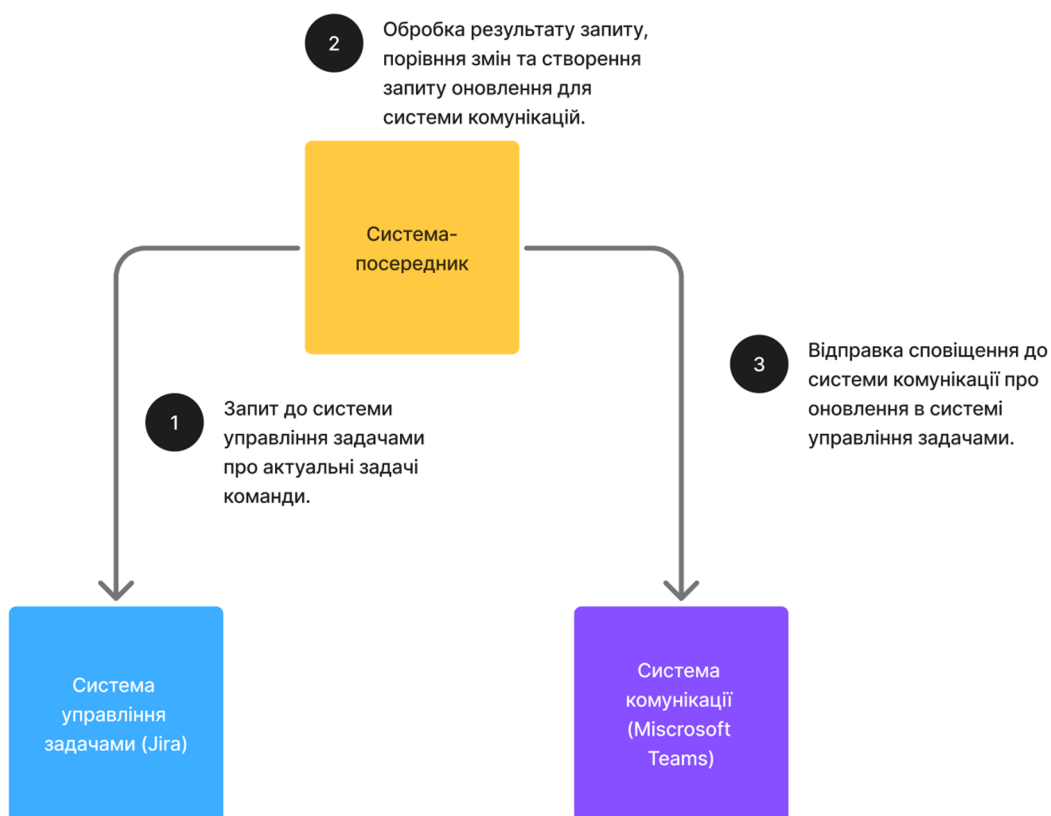


Рисунок 2.1 – Приклад інтеграції за допомогою API

Таким чином ми можемо бачити, що впровадження інтеграції за допомогою API буде вимагати щонайменше виконання трьох основних кроків роботи такої імплементації:

- запит до СУЗ про актуальні задачі команди;
- обробка результату запиту, порівняння змін та створення запиту оновлення для системи комунікацій;
- відправка сповіщення до системи комунікації про оновлення в СУЗ задачами.

Тож важливо зауважити, що така реалізація інтеграції може бути складнішою за інші способи і кінцевий вибір залежить виключно від потреб інтеграції і незважаючи на переваги API, потрібно зважати й на недоліки.

Таблиця 2.1 – Перелік основних переваг та недоліків використання API для інтеграції.

Переваги API	Недоліки
Масштабованість, можливість розширювати функціонал та динамічно розгортати необхідні ресурси.	Час розробки, складні API можуть потребувати значних ресурсів чи часу на імплементацію.
Стандартизований підхід до розробки, велика кількість доступних прикладів створення та підтримки API функціоналу.	Необхідність самостійно забезпечувати безпеку API, якісне логування та інший суміжний функціонал.
Гнучкість та повний контроль над імплементацією інтеграції.	Забезпечення надійності та доступність сервісу частково є обов'язком розробника.
Розширюваність, можливість підключення сторонніх сервісів.	Складність підтримання роботи API може значно зростати з часом через необхідність підтримки старих версій, різних запитів і т.п.

2.3 Використання Webhook для реагування на події між системами

Webhook (вебхук) є механізмом для автоматизованого обміну даними між різними сервісами та додатками у реальному часі. Вони дозволяють одній системі надсилати дані до іншої на основі певних подій та складаються з наступних ключових елементів: подія – визначена дія у системі (наприклад, створення нової задачі в СУЗ); запит – надсилання системою запиту на заздалегідь налаштовану адресу іншого сервісу; обробка – отриманий запит містить дані про подію, які можуть бути використані для подальших дій (наприклад, автоматичне створення відповідного повідомлення в Microsoft Teams).

Вебхуки зазвичай використовуються для автоматизація процесів (синхронізація даних між додатками), інтеграції різних сервісів (наприклад, передача інформації з платформи комунікації в СУЗ).

На відміну від API, які потрібно опитувати для отримання оновлень, Webhooks працюють «за запитом», тобто система, що генерує подію, сама надсилає необхідні дані. Це робить їх ефективним рішенням для інтеграції в цифрових середовищах.

В разі впровадження інтеграції за допомогою Webhook-запиту, необхідно виконати такі кроки як: налаштування вхідного Webhook посилання в Microsoft Teams (на цей вебхук система задач може відсилати оновлення); створення триггеру в СУЗ для відправки запитів. Зазвичай запит формується у форматі JSON. Оновлення СУЗ автоматично відправляються в систему-посередник. Тож така інтеграція має включати в себе наступні три важливі кроки:

- оновлення СУЗ автоматично відправляються в систему-посередник;
- обробка оновлення;
- відправка сповіщення до системи комунікації про оновлення в СУЗ.

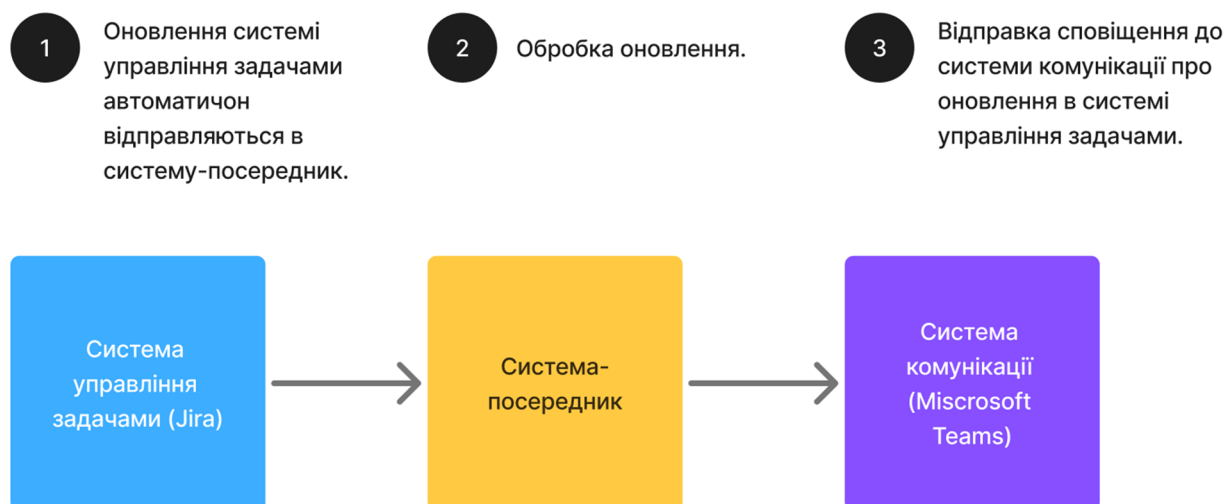


Рисунок 2.2 – Приклад інтеграції за допомогою Webhook

Аналогічно з іншими способами, Webhook має як певні переваги, так і вагомі недоліки, тож кінцеве рішення має залежати напряду від потреб інтеграції та запланованого функціоналу. В таблиці 2.2 наведено більш детальне порівняння.

Таблиця 2.2 – Перелік основних переваг та недоліків використання технології Webhook для інтеграції.

Переваги Webhook	Недоліки
1	2
Миттєва передача даних, Webhook працює за принципом миттєвих сповіщень, що дозволяє отримувати оновлення в реальному часі.	Обмежена гнучкість, необхідна складна логіка для підтримки додаткових механізмів.
Легкість та простота налаштування, такий підхід є дуже зрозумілим і швидким в імплементації.	Відсутність підтвердження отримання, Webhook не надає вбудованного механізму перевірки отримання даних.

Кінець таблиці 2.2

1	2
Асинхронізація оновлення, Webhook містить останні дані та не додає затримок на повну обробку запиту.	Відсутність повторних запитів, такий спосіб не дозволяє відновлення попередніх відповідей.

2.4 Інтеграція через додатки розширення

Microsoft Teams має вбудований каталог додатків-розширень, який дозволяє легко знаходити та інтегрувати сторонні додатки, зокрема Jira Cloud, для покращення робочих процесів. Цей магазин також забезпечує централізовану платформу, де користувачі можуть додати різноманітні інструменти для автоматизації задач, комунікації та управління проектами. Зазвичай такий стандартизований підхід дозволяє значно спростити реалізацію аутентифікації, координацію програмних оновлень та керування дозволів доступу.

Тож важливо підкреслити, що головні переваги інтеграції через каталог додатків є простота встановлення та налаштування інтеграції без необхідності ручної реалізації API чи Webhook; централізоване оновлення забезпечення; офіційна та повноцінна підтримка зі сторони Microsoft та Atlassian. Повноцінна інтеграція Microsoft Teams та Jira Cloud дозволяє командам ефективно керувати завданнями, отримувати сповіщення та взаємодіяти безпосередньо у Teams, не перемикаючись між платформами.

Повноцінна інтеграція Microsoft Teams та Jira Cloud дозволяє командам ефективно керувати завданнями, отримувати сповіщення та взаємодіяти безпосередньо у Teams, не перемикаючись між платформами. Така інтеграція забезпечує високий рівень безпеки та надійності, оскільки проходить через офіційні канали та пройшла сертифікацію від обох компаній. Крім того,

використання готового додатку з каталогу гарантує регулярні оновлення функціоналу та виправлення помилок без додаткових зусиль з боку ІТ-команди.

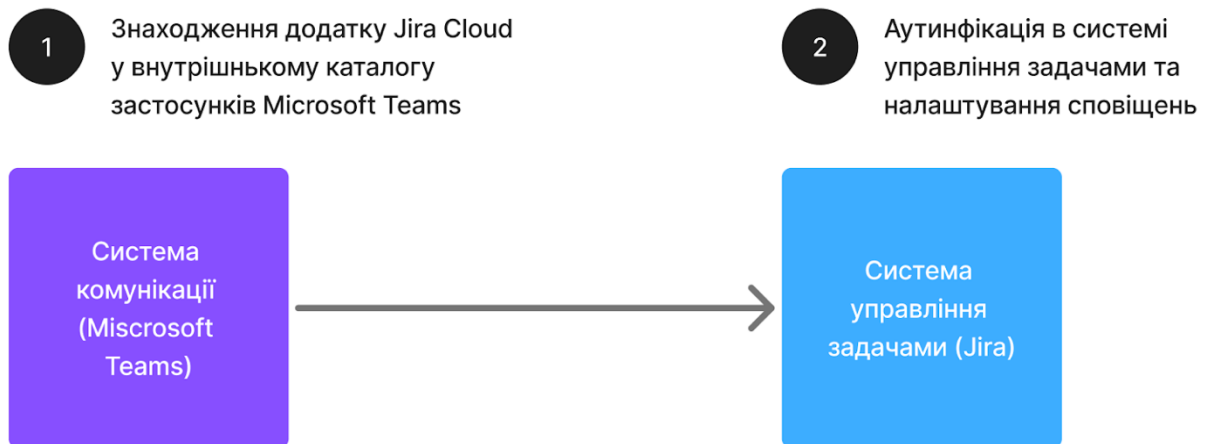


Рисунок 2.3 – Приклад інтеграції через каталог додатків Microsoft Teams

2.5 Порівняльний аналіз підходів: ефективність, масштабованість, застосовність

API, Webhook і інтеграція через застосунок із каталогу додатків представляє собою три різні способи взаємодії програмного забезпечення, які базуються на протоколі передачі гіпертексту (Hypertext Transfer Protocol, HTTP). Ось їхні основні відмінності:

API має гнучкість, що дозволяє програмам взаємодіяти через чітко визначені запити та відповіді підтримуючи синхронну чи асинхронну роботу в якій зазвичай розробники використовують API для доступу до функціональності або даних інших застосунків чи сервісів.

Webhook з іншої сторони є подієво-орієнтований підхід та працює на основі оновлень, які надсилаються системою на визначений уніфікований локатор ресурсу (Uniform Resource Locator, URL) для сповіщення залежних сервісів. На відміну від API, який потребує запитів, Webhook автоматично

надсилає дані без необхідності перевірки стану і такий підхід чудово підходить для поширення оперативних повідомлень.

Інтеграція через застосунок із каталогу зазвичай не вимагає глибоких технічних знань, оскільки інтеграція проходить через графічний інтерфейс. Єдиний дуже важливий недолік, що залишається, це обмежена кастомізація, оскільки можливості гнучкого налаштування можуть бути обмеженими і чітко передбачаються розробленим функціоналом.

Тож важливо підкреслити, що кожен зі способів має свої вагомні переваги та недоліки та має бути застосований там, де він краще всього підходить з точки зору потреб системи та вимог проєкту. Більш детальне порівняння цих методів інтеграції СУЗ та комунікацій наведено в таблиця 2.3.

Таблиця 2.3 – Порівняння різних підходів інтеграції за критеріями.

Характеристика	API	Webhook	Каталог застосунків Microsoft Teams
1	2	3	4
Механізм інтеграції	Запити між серверами через HTTP	Автоматичне надсилання даних	Вбудовані додатки у Microsoft Teams
Тип виклику	Запити GET, POST, PUT, DELETE, PATCH	Подія активує HTTP-запит POST	Встановлення та авторизація в Teams

Кінець таблиці 2.3

1	2	3	4
Формат даних	JSON, eXtensible Markup Language (XML)	JSON	Графічний інтерфейс
Автоматизація	Вимагає написання коду	Може бути створений без коду	Автоматизація через вбудовані функції
Безпека	Авторизація через відкритий протокол авторизації (Open Authorization, OAuth)	Обмежена безпека	Контроль доступу через адміністрування Teams
Гнучкість	Висока – може працювати з будь-яким сервісом	Легке налаштування для певних подій	Обмежено функціональністю Teams Apps
Складність імплементації	Висока – вимагає розробки та тестування	Середня – лише налаштування URL	Низька – встановлення з магазину
Призначення	Глибока інтеграція між системами	Легка передача подій між сервісами	Спрощене використання в Teams

3 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ ІНТЕГРАЦІЇ СИСТЕМ УПРАВЛІННЯ ЗАДАЧАМИ ТА КОМУНІКАЦІЇ

3.1 Проектування інтеграційного рішення для ІТ-команди

Попередні розділи дослідження продемонстрували важливість інтеграції СУЗ та комунікації у підвищенні ефективності роботи ІТ-команд. На основі проведеного аналізу існуючих методів інтеграції та результатів опитування, представлених у розділі 1.4.1, було розроблено загальний план проектування інтеграційного рішення, що відповідає потребам сучасних ІТ-команд.

Проектування інтеграційного рішення потребує системного підходу та врахування багатьох факторів, включаючи організаційні, технічні та функціональні аспекти, а також особливості робочих процесів команди. Основні етапи проектування інтеграційного рішення включають:

- аналіз поточних робочих процесів, ідентифікація проблемних місць;
- визначення вимог до інтеграційного рішення;
- вибір оптимального методу інтеграції;
- проектування архітектури рішення;
- розробка плану впровадження та тестування.

У рамках першого етапу було проведено аналіз робочих процесів типової ІТ-команди, яка використовує Jira для управління задачами та Microsoft Teams для комунікації. Аналіз дозволив виявити такі проблемні місця:

- втрата часу на перемикання між системами (в середньому 15-30 хвилин на день згідно з даними опитування);
- втрата контексту при перемиканні між системами (54% респондентів);
- відсутність автоматичного оновлення статусів задач (67% респондентів забувають оновлювати статуси);
- неефективні щоденні зустрічі через необхідність синхронізації інформації з різних джерел.

На основі виявлених проблем та результатів опитування було сформульовано наступні вимоги до інтеграційного рішення:

- забезпечення можливості оновлення статусів задач безпосередньо з Microsoft Teams;
- автоматичне сповіщення про зміни статусів задач у Microsoft Teams;
- можливість перегляду деталей задач без переходу в Jira;
- автоматизація щоденних звітів та стендапів;
- простота налаштування та використання.

Враховуючи сформульовані вимоги та результати аналізу методів інтеграції, представлені в розділі 2, для практичної реалізації було обрано три підходи:

- інтеграція через додаток з каталогу Microsoft Teams;
- інтеграція через API;
- інтеграція через Webhook.

Кожен з цих підходів має свої переваги та недоліки, і вибір конкретного методу залежить від потреб команди, технічних можливостей та наявних ресурсів. Для комплексного вирішення виявлених проблем запропоновано архітектуру інтеграційного рішення, що враховує модулі необхідного функціоналу, але залишає вибір інтеграції за розробником.

Архітектура запропонованого рішення представлена на рис. 3.1 та включає такі основні модулі функціоналу:

- модуль сповіщень у Microsoft Teams про зміни в Jira;
- модуль управління задачами Jira з інтерфейсу Microsoft Teams;
- модуль автоматизації щоденних звітів та стендапів;
- модуль синхронізації даних між системами.

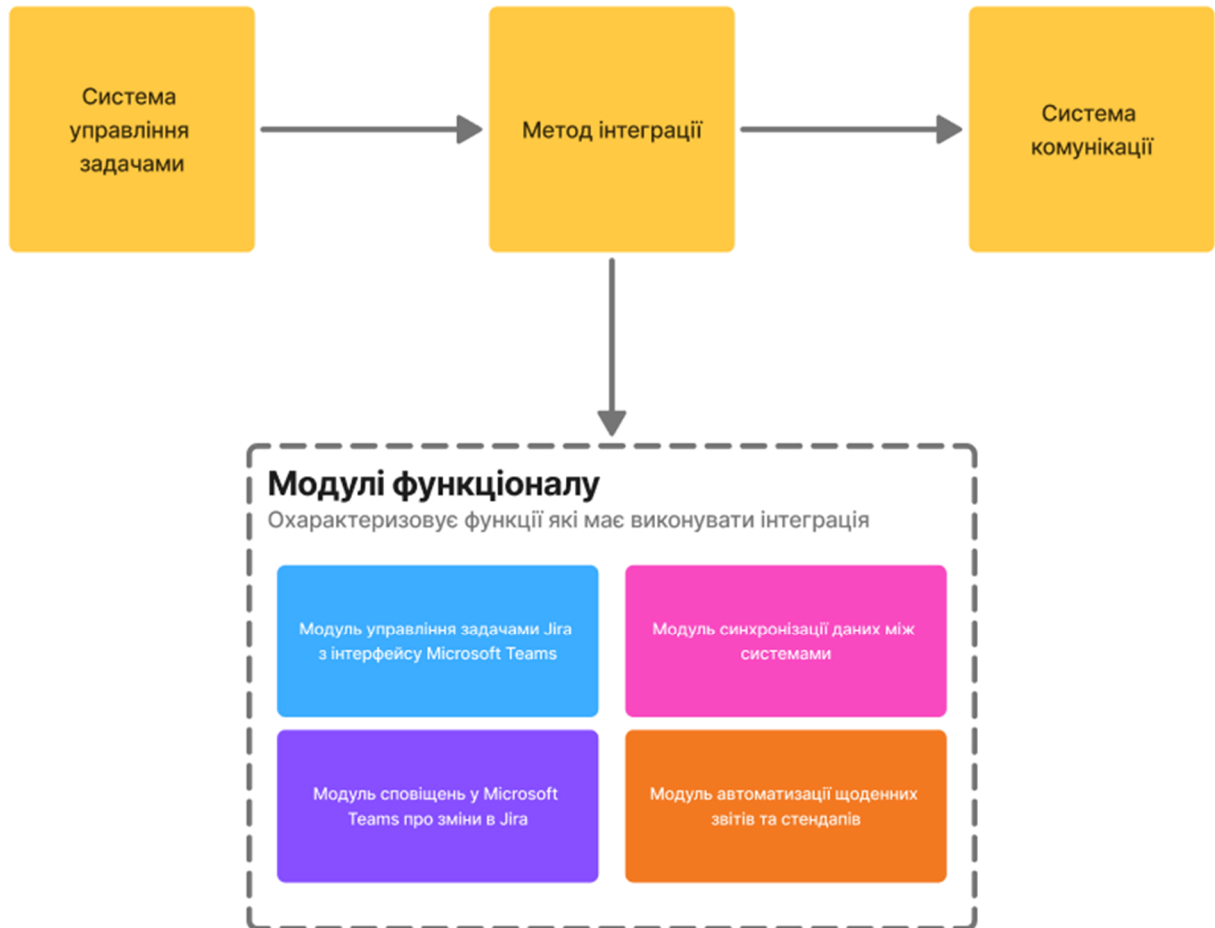


Рисунок 3.1 – Архітектура інтеграційного рішення

Такий підхід до проектування інтеграційного рішення дозволяє забезпечити комплексне вирішення виявлених проблем та підвищити ефективність роботи ІТ-команди. Детальна реалізація кожного з компонентів представлена у наступних підрозділах.

3.2 Покрокова реалізація інтеграції Jira з Microsoft Teams

3.2.1 Реалізація інтеграції через додаток з каталогу

Одним з найбільш доступних та швидких способів інтеграції Jira та Microsoft Teams є використання готового додатку з каталогу Microsoft Teams. Цей підхід не потребує глибоких технічних знань та може бути реалізований

адміністратором Teams або Jira без залучення розробників.

Для практичної реалізації інтеграції через додаток з каталогу було проведено наступні кроки:

Крок 1. Пошук та вибір відповідного додатку

Першим етапом реалізації є пошук офіційного додатку Jira для Microsoft Teams у каталозі. Для цього необхідно:

Відкрити Microsoft Teams та перейти до розділу «Програми» (Apps)

У пошуковому полі ввести «Jira»

Серед запропонованих варіантів обрати офіційний додаток «Jira Cloud» від компанії Atlassian. Результат пошуку наведено на рисунку 3.2.

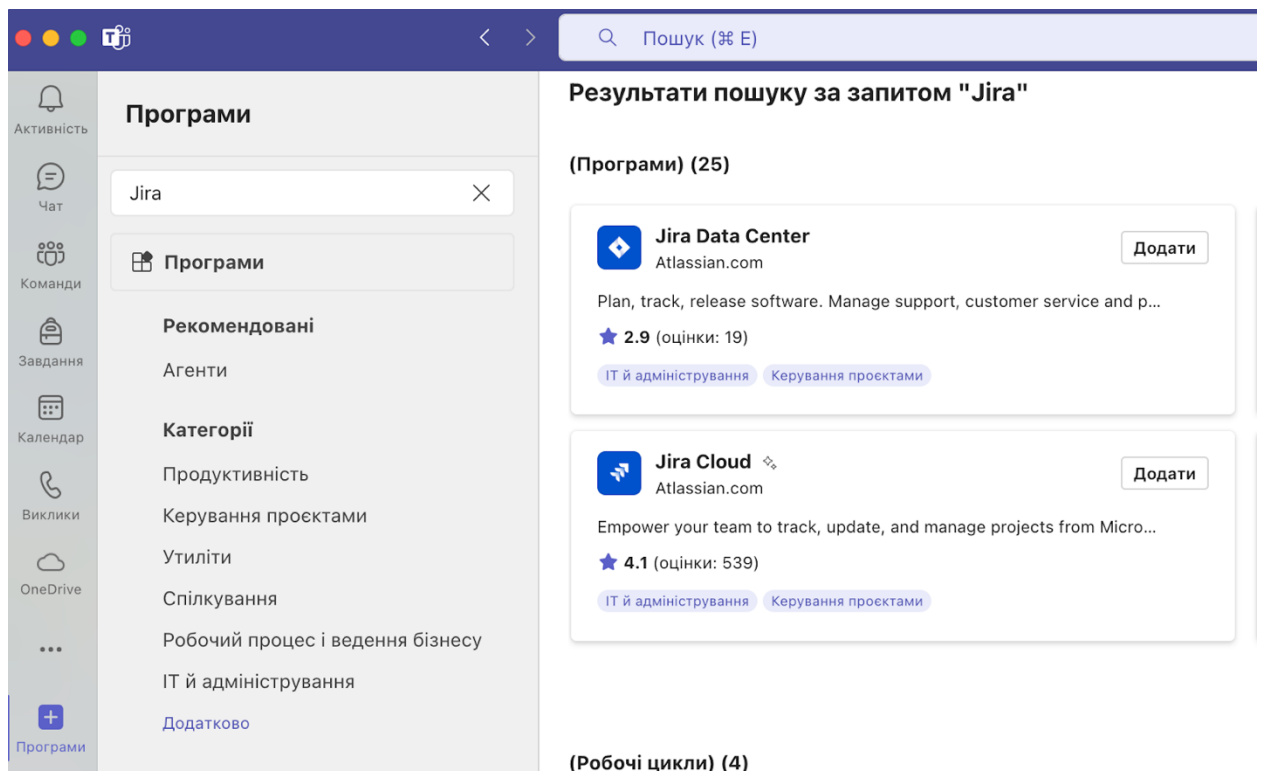


Рисунок 3.2 – Результати пошуку офіційного додатку Jira

Крок 2. Авторизація та налаштування з'єднання

Після встановлення додатку необхідно провести авторизацію та налаштувати з'єднання з обліковим записом Jira. Для цього необхідно перейти в чат програми, який буде створено після її встановлення, натиснути на кнопку «Почати», та авторизуватися в системі Atlassian. Таким чином додаток Jira

отримає доступ до необхідного облікового запису.

Важливо зазначити, що для успішної авторизації користувач повинен мати відповідні права в обох системах. Зазвичай для повноцінної інтеграції потрібні права адміністратора або користувача з розширеними можливостями як у Jira, так і в Microsoft Teams. Це пов'язано з тим, що додаток потребує доступу до різних API та ресурсів обох платформ для забезпечення безперебійної роботи інтеграційного рішення.

Після завершення авторизації система автоматично створить захищений канал зв'язку між Jira та Microsoft Teams, що дозволить безпечно передавати дані про задачі, коментарі, статуси та інші важливі елементи робочого процесу а користувач отримує доступ до чат-боту Jira, за допомогою якого ми можемо налаштовувати сповіщення та взаємодіяти з Jira за допомогою Microsoft Teams.

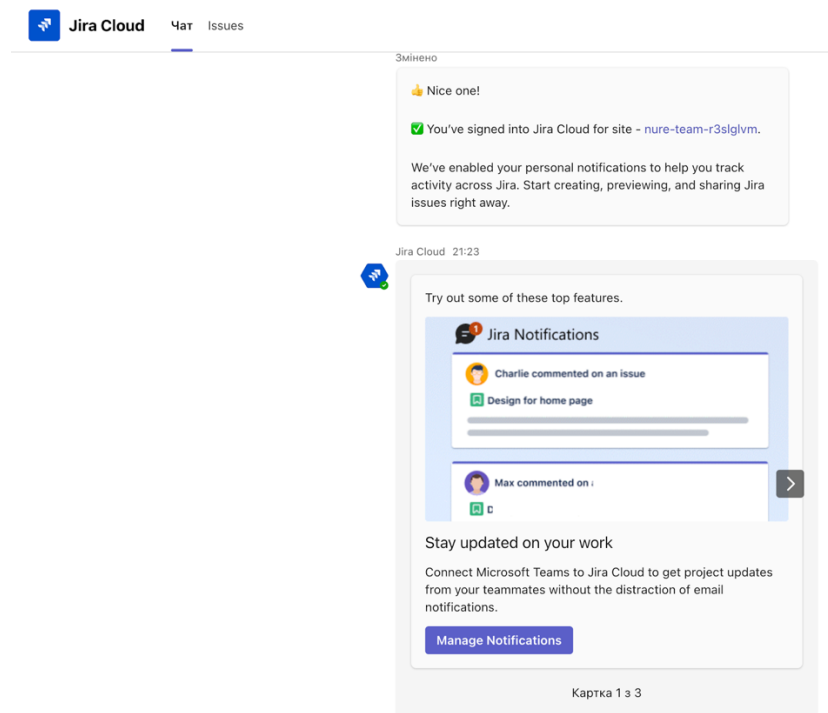


Рисунок 3.3 – Результат авторизації

Крок 3. Налаштування проєктів та сповіщень
Для налаштування сповіщень достатньо перейти у чат-бот Jira у

програмі Microsoft Teams, відправити повідомлення help, і в меню що відкрилось обрати опцію Сповіщення (Notifications), та увімкнути сповіщення для свого акаунту. Процес налаштування сповіщень наведено на рисунку 3.4.

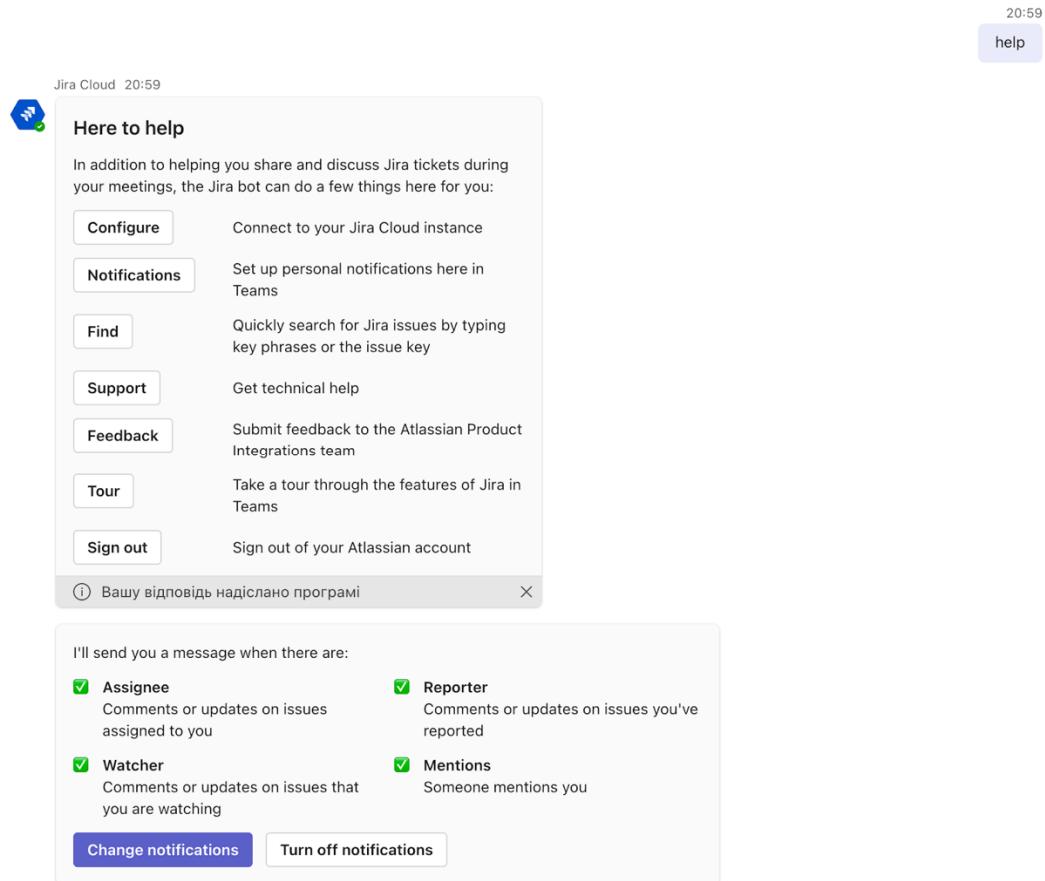


Рисунок 3.4 – Процес налаштування сповіщень

Після завершення налаштувань інтеграції, система починає автоматично надсилати повідомлення у Microsoft Teams при виникненні визначених подій у Jira. Наприклад, коли користувача призначають відповідальним за нову задачу, він миттєво отримує відповідне сповіщення у Teams.

Таке повідомлення містить ключову інформацію: назву задачі, проєкт, її поточний статус, пріоритет, а також пряме посилання для переходу в Jira. Це дозволяє оперативно відреагувати на зміну, не залишаючи середовища Teams, що, у свою чергу, знижує кількість перемикачів між системами та підвищує ефективність комунікації в команді.

На рисунку 3.5 наведено приклад такого сповіщення про призначення задачі.

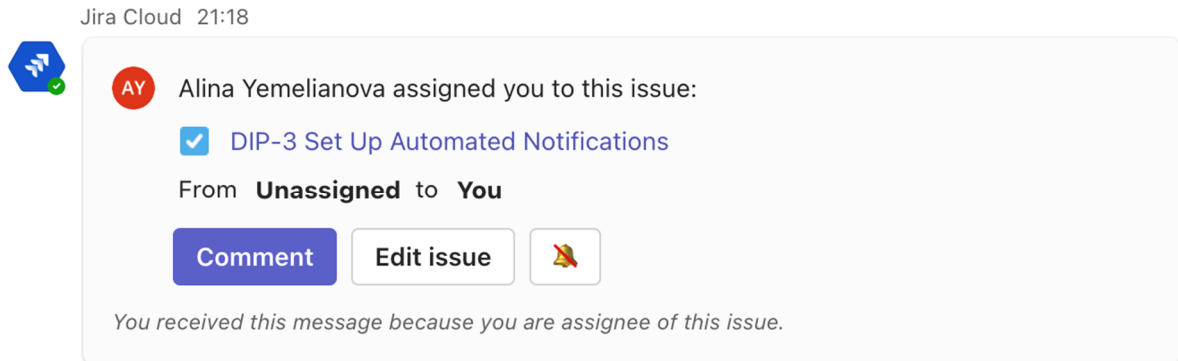


Рисунок 3.5 – Приклад сповіщення про призначення задачі

Після завершення налаштувань додаток Jira Cloud у Microsoft Teams одразу готовий до роботи. Основні функції доступні прямо з Teams, що дуже зручно для щоденної командної взаємодії.

Тепер можна легко переглядати свої задачі Jira без потреби відкривати окрему вкладку. Коли в Jira щось змінюється – наприклад, вас призначили відповідальним або оновився статус задачі – ви миттєво отримуєте повідомлення у Teams. Це допомагає бути в курсі подій без зайвих перемикань.

Загалом, така інтеграція є достатньо простою у впровадженні: налаштування займає буквально 15–30 хвилин. Інтерфейс зручний і зрозумілий навіть для тих, хто раніше не працював із подібними інтеграціями. При цьому, функціональність досить гнучка – можна адаптувати під конкретні потреби команди. А оскільки додаток офіційний і підтримується Atlassian, усе працює стабільно і безпечно.

Для більшості команд це чудовий варіант, щоб швидко почати працювати з інтеграцією Jira та Microsoft Teams і одразу відчути покращення в комунікації та управлінні задачами.

3.2.2 Дослідження інтеграції через REST API

REST API є одним із найпотужніших методів інтеграції СУЗ та комунікації, забезпечуючи гнучкість, масштабованість та високу адаптивність рішень. Цей метод базується на архітектурному стилі REST (Representational State Transfer), який використовує стандартні методи протоколу передачі гіпертексту (HTTP) для взаємодії між системами.

Інтеграція Jira з Microsoft Teams через REST API будується на кількох ключових принципах. Насамперед, це чітке розділення функцій клієнта (Microsoft Teams) та сервера (Jira), що забезпечує незалежність систем. Важливим аспектом є безстановість комунікації, тобто кожен запит містить всю необхідну інформацію, що спрощує масштабування та підвищує надійність. Не менш важливою є можливість кешування результатів запитів для підвищення продуктивності, а також уніфікований інтерфейс зі стандартними методами взаємодії (GET, POST, PUT, DELETE), що робить інтеграцію інтуїтивно зрозумілою.

Jira надає потужний REST API з широким набором ендпоінтів. Основні з них включають:

- `/rest/api/2/issue` – для роботи з задачами;
- `/rest/api/2/project` – для роботи з проектами;
- `/rest/api/2/user` – для роботи з користувачами;
- `/rest/api/2/comment` – для роботи з коментарями;

Jira використовує нотацію об'єктів JavaScript (JavaScript Object Notation, JSON) як основний формат передачі даних і підтримує кілька методів автентифікації, включаючи Basic Authentication з API-токеном та OAuth 2.0.

Microsoft Graph API, у свою чергу, є єдиною точкою доступу до ресурсів екосистеми Microsoft 365, включаючи Teams. Цей API також використовує JSON як основний формат даних і OAuth 2.0 для автентифікації. Важливо, що Graph API надає доступ не лише до Teams, але й до інших сервісів Microsoft,

що розширює можливості інтеграції.

Порівняльний аналіз методів автентифікації виявив суттєві відмінності між ними, які представлені в таблиці 3.1.

Таблиця 3.1 – Порівняння методів автентифікації для інтеграції Jira з Microsoft Teams

Метод автентифікації	Переваги	Недоліки	Рекомендоване застосування
Basic Authentication з API-токеном	Простота реалізації, висока швидкість	Нижчий рівень безпеки	Внутрішні або прототипні рішення
OAuth 2.0	Висока безпека, можливість обмеження доступу	Складність налаштування	Продуктивні рішення, особливо з чутливими даними
JWT (JSON Web Tokens (JWT))	Ефективність для мікросервісної архітектури	Потребує додаткової інфраструктури	Складні інтеграційні рішення з багатьма компонентами

Досліджуючи архітектурні моделі інтеграції через REST API, було виявлено три основні підходи. Перший – пряма інтеграція, коли системи безпосередньо взаємодіють через API без проміжних компонентів. Цей підхід відзначається простотою реалізації, але має обмежену функціональність і меншу гнучкість.

Другий підхід передбачає використання проміжного сервісу, який взаємодіє з API обох систем. Така архітектура забезпечує більшу гнучкість і розширені можливості налаштування. Проміжний сервіс може виконувати додаткову логіку, трансформувати дані та забезпечувати додатковий рівень

безпеки.

Третій підхід – мікросервісна інтеграція, коли система розбивається на окремі компоненти, кожен з яких відповідає за певний аспект інтеграції. Цей підхід забезпечує найвищу гнучкість і масштабованість, але є найскладнішим у реалізації та потребує більше ресурсів для розробки та підтримки.

Ефективність інтеграції через REST API значною мірою залежить від того як саме вона використовується. Якщо мова йде про синхронізацію даних у режимі реального часу REST API не завжди є найкращим варіантом – адже потребує постійного звернення до систем. Це створює додаткове навантаження. У таких ситуаціях доцільно поєднувати REST API з Webhook – це дозволяє оперативно отримувати сповіщення про важливі події без надмірної кількості запитів.

З іншого боку для періодичної синхронізації REST API показує себе дуже добре. Це просте в реалізації рішення яке не створює надмірного навантаження. Наші спостереження свідчать, що оптимальний інтервал синхронізації – від 5 до 15 хвилин, залежно від того наскільки часто змінюються дані та наскільки актуальними вони мають бути.

REST API також добре підходить для реалізації складних сценаріїв де потрібно обробляти кілька етапів чи виконувати складну логіку. В таких випадках він дає максимальну гнучкість але при цьому вимагає уважного підходу до обробки помилок і передбачення ситуацій на випадок збоїв.

У процесі роботи ми також зіткнулися з кількома викликами. Наприклад обидві системи (Jira і Teams) мають обмеження на кількість запитів за певний час що може впливати на швидкість обміну даними. Це потребує оптимізації – важливо уникати зайвих запитів і грамотно планувати навантаження.

Окрему увагу варто приділити безпеці. Потрібно надійно захищати облікові дані й токени дотримуватись вимог щодо безпеки корпоративної мережі, та відповідати вимогам щодо захисту персональних даних. Також варто враховувати технічні нюанси – наприклад різницю у версіях API, сумісність між ними та потенційні зміни при оновленнях.

У процесі роботи було визначено ключові рекомендації для того, щоб зробити інтеграцію через REST API максимально ефективною:

Пакетна обробка запитів – об'єднання кількох операцій в один запит для зменшення мережевого трафіку та пришвидшення обробки даних.

Кешування даних – зберігання результатів частих запитів що значно зменшує навантаження на API.

Асинхронна обробка – використання черг повідомлень для операцій які не потребують миттєвої відповіді, що дозволяє більш рівномірно розподілити навантаження.

Стратегії повторних запитів – впровадження механізмів з експоненційною затримкою для автоматичного відновлення після тимчасових, помилок.

Інтеграція через REST API – це потужний і гнучкий спосіб поєднання СУЗ та комунікації. Вона дозволяє реалізовувати широкий спектр сценаріїв: від простої синхронізації даних до складних бізнес процесів. Проте для того щоб така інтеграція працювала ефективно, важливо заздалегідь продумати архітектуру, оптимізувати запити і впровадити механізми що відповідають за надійність і безпеку.

3.2.3 Дослідження інтеграції за допомогою Webhook

Webhook – це один із найзручніших і справді ефективних способів інтеграції між різними системами, який особливо активно почали використовувати останні роки. Його підхід до обміну даними суттєво відрізняється від класичного REST API, адже він працює за реактивним принципом.

Під час дослідження було виявлено, що Webhook відкриває нові можливості для інтеграції СУЗ та комунікацій, особливо в тих випадках, коли

важлива швидка реакція на зміни. Цей метод особливо ефективний для команд, які працюють у динамічному середовищі, де статуси задач змінюються часто протягом дня.

Фактично Webhook – це механізм, який дозволяє «підписатись» на певні події. На відміну від звичайної схеми, де клієнт постійно звертається до сервера в пошуках змін, Webhook працює за моделлю «публікація–підписка», тобто сама система-джерело надсилає повідомлення, щойно щось змінюється. Такий спосіб дає помітні переваги у швидкодії масштабованості та зменшує затримки в обміні даними.

На практиці це працює наступним чином: система-джерело (Jira) надсилає HTTP POST-запит на попередньо налаштований URL системи-приймача кожного разу, коли відбувається певна подія, така як створення нової задачі або зміна статусу існуючої. Система-приймач оброблює цей запит і здійснює відповідні дії, які можуть включати відображення сповіщення у Microsoft Teams. Завдяки асинхронній природі Webhook, такі сповіщення надходять практично миттєво після зміни статусу задачі. Відповідна комунікація зображена на рисунку 3.6.



Рисунок 3.6 – Алгоритм роботи Webhook інтеграції

Порівнюючи Webhook з іншими методами інтеграції, було виявлено суттєві відмінності, представлені в таблиці 3.2:

Таблиця 3.2 – Порівняння Webhook з іншими методами інтеграції

Характеристика	Webhook	REST API	Черги повідомлень
Швидкість передачі даних	Дуже висока	Середня	Висока
Гарантія доставки	Низька	Висока	Дуже висока
Складність реалізації	Низька	Середня	Висока
Масштабованість	Середня	Висока	Дуже висока
Режим комунікації	Асинхронний	Синхронний/Асинхронний	Асинхронний

Під час дослідження були визначені основні компоненти Webhook у Jira:

Підсистема реєстрації Webhook дозволяє вказати URL-адресу і налаштувати події, які мають запускати надсилання повідомлень.

Фільтри подій – допомагають зменшити кількість непотрібних сповіщень, використовуючи JQL-запити, які дозволяють реагувати лише на конкретні події

Формувач даних – відповідає за підготовку JSON-повідомлення, що містить інформацію про подію, змінені поля, користувача, тощо.

Механізм доставки – надсилає HTTP POST-запит, і намагається обробити помилки, якщо такі виникають.

Однак Webhook сам по собі не гарантує 100% доставку подій – у ньому немає вбудованого механізму підтвердження чи повторної відправки у разі помилки.

Щоб зробити інтеграцію більш надійною, було запропоновано кілька концептуальних рішень:

- концепція ідемпотентної обробки, яка допомагає уникнути дублювання дій при повторному надходженні однієї й тієї ж події;

- архітектурне рішення з проміжним сервісом, який зберігає події у буфері, перш ніж передати їх на обробку;

- методологія впровадження системи логування та моніторингу, щоб відслідковувати успішність обробки повідомлень.

Безпеці також приділили багато уваги. Теоретичний підхід включає:

- використання секретних токенів, щоб перевірити, що запит дійсно надійшов із Jira;
- підпис повідомлень цифровим підписом, щоб перевірити, що дані не були змінені;
- обмеження доступу до Webhook за IP-адресами і обов'язкове використання HTTPS для шифрування даних.

У результаті дослідження було з'ясовано, що найкраще працює архітектура з використанням проміжного сервісу – вона дає більшу стабільність, контроль над обробкою і можливість масштабування.

Проведене дослідження виявило оптимальні сценарії використання Webhook:

- сповіщення про критичні події в проєкті;
- автоматизація робочих процесів;
- агрегація та візуалізація статистики по проєкту.

Важливим практичним аспектом стало визначення оптимальних параметрів конфігурації. Зокрема, було виявлено що для більшості випадків оптимальним є використання кількох спеціалізованих Webhook з чітко визначеними JQL-фільтрами, замість одного загального. Це дозволяє значно знизити навантаження на системи та спростити обробку повідомлень.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ ЗАПРОПОНОВАНИХ МЕТОДІВ ІНТЕГРАЦІЇ

4.1 Критерії ефективності з точки зору керування ІТ-командою

Розробка та впровадження інтеграційних рішень між СУЗ та комунікації вимагає чіткого розуміння критеріїв ефективності, за якими можна оцінити результативність таких рішень. У цьому підрозділі визначено та аргументовано основні критерії оцінювання ефективності інтеграційних рішень з точки зору керування ІТ-командою.

На основі проведеного в розділі 1.4.1 опитування ІТ-спеціалістів компанії CodeIT (31 респондент, серед яких 6 проєктних менеджерів, 22 розробники та 3 дизайнери), а також аналізу наукової літератури та галузевих досліджень, були визначені ключові критерії ефективності, які найбільш релевантні для інтеграції СУЗ та комунікації:

- часові показники ефективності – включають скорочення часу на перемикання між системами, швидкість доступу до контекстної інформації та оперативність оновлення статусів задач;
- якісні показники комунікації – оцінюють цілісність інформаційного контексту, прозорість процесів та ефективність асинхронної комунікації;
- управлінські критерії – визначають актуальність даних моніторингу, зниження адміністративного навантаження та швидкість реакції на зміни й ризику;
- технічні критерії – враховують надійність інтеграційного рішення, безпеку даних, масштабованість та гнучкість системи.

Згідно з результатами опитування, 72% учасників витрачають на перемикання між Jira та Microsoft Teams від 15 до 30 хвилин щодня, що становить до 2,5 годин робочого часу на тиждень. За даними дослідження McKinsey & Company, близько 19% робочого часу в технологічних компаніях витрачається на пошук та обмін інформацією між різними системами [13].

Скорочення часу на перемикання між системами є ключовим критерієм, що вимірюється як відсоток зменшення часу, який учасники команди витрачають на переключення між системами. Особливо гостро це питання стоїть для 67% розробників, які, за даними опитування, періодично забувають оновлювати статус задач у Jira, що ускладнює моніторинг і планування проєктів.

Якісні показники комунікації та взаємодії безпосередньо впливають на результативність роботи ІТ-команди. Команди, які використовують інтегровані рішення для управління проєктами та комунікації, демонструють на 23% вищу ефективність командної взаємодії [14]. За результатами опитування, 42% респондентів вказали на відсутність контексту обговорень у Jira як одну з головних проблем у взаємодії між системами.

Таблиця 4.1 – Порівняльна характеристика методів інтеграції за основними критеріями ефективності

Критерій ефективності	API-інтеграція	Webhook-інтеграція	Додаток з каталогу Microsoft Teams
1	2	3	4
Часова ефективність	Висока за умови оптимізації запитів	Дуже висока завдяки миттєвим оновленням	Середня, обмежена функціоналом додатка
Ефективність комунікації	Висока, гнучке налаштування сповіщень	Середня, односторонні оновлення	Дуже висока завдяки нативній інтеграції
Технічна інтеграція	Середня, потребує розробки	Висока, просте налаштування	Дуже висока, мінімальні технічні вимоги

Кінець таблиці 4.1

1	2	3	4
Управлінська ефективність	Дуже висока завдяки можливості глибокого аналізу даних	Висока для реактивного управління	Середня, обмежена функціоналом додатка
Складність впровадження	Висока, потребує розробки	Середня	Низька, встановлення «в один клік»
Безпека даних	Висока, гнучкі налаштування	Середня, обмежені можливості контролю	Висока, сертифікована Microsoft

Інтеграція через API демонструє найкращі результати за критеріями гнучкості та можливості глибокого аналізу даних, що особливо важливо для управлінської ефективності. Webhook-інтеграція забезпечує найвищу часову ефективність завдяки миттєвому оновленню статусів без необхідності періодичного опитування системи. Інтеграція через додаток з каталогу Microsoft Teams показує найкращі результати за критеріями простоти налаштування та технічної інтеграції, а також забезпечує високий рівень ефективності комунікації через нативну інтеграцію з платформою.

Проведене дослідження підтверджує, що вибір оптимального методу інтеграції залежить від конкретних пріоритетів команди та проєкту. Для команд, де критично важлива швидкість оновлення інформації, Webhook може бути оптимальним вибором. Для проєктів, де потрібна глибока аналітика та гнучкість налаштувань, інтеграція через API має переваги. Якщо ж пріоритетом є швидке впровадження та низькі витрати на розробку, додаток з каталогу Microsoft Teams є найбільш доцільним рішенням.

Важливо зазначити, що при оцінці різних інтеграційних рішень слід враховувати контекст конкретної ІТ-команди та проєкту. Результати опитування показують, що пріоритети можуть суттєво відрізнятися: для великих команд (понад 15 осіб) на перший план виходить прозорість процесів та ефективність асинхронної комунікації, тоді як для менших команд більш важливими є швидкість доступу до інформації та зниження адміністративного навантаження.

Застосування визначених критеріїв ефективності дозволяє об'єктивно порівняти різні методи інтеграції між СУЗ та комунікації, а також обрати оптимальне рішення для конкретної ІТ-команди та проєкту. Детальний аналіз результатів практичного тестування та методика оцінки представлені в наступному підрозділі.

4.2 Методика практичної перевірки та оцінка впливу запропонованих рішень

Для оцінювання ефективності запропонованих методів інтеграції СУЗ та комунікації було розроблено методику їх практичного тестування у реальних умовах. Тестування проводилося в українській ІТ-компанії CodeIT що спеціалізується на розробці програмного забезпечення, та має у своєму складі різнопрофільні проєктні команди.

Дослідження базувалося на результатах опитування 31 учасника проведеного в розділі 1.4.1, серед яких були 6 проєктних менеджерів, 22 розробники та 3 дизайнери. На основі цього опитування було визначено ключові проблеми пов'язані з використанням окремих інструментів без інтеграції, а також розроблено методику практичної перевірки запропонованих рішень.

Для детального тестування методів інтеграції було взято три існуючі

проектні команди різного розміру та складу що представляли типові структури ІТ-проектів у компанії. Команди працювали над реальними проектами з використанням Jira як СУЗ та Microsoft Teams як основного засобу комунікації. Тестування проводилося протягом трьох тижнів, причому кожен тиждень був присвячений окремому методу інтеграції.

На підготовчому етапі було проаналізовано початкові показники ефективності роботи команд без використання інтеграційних рішень. Результати підтвердили проблеми виявлені в опитуванні: значні витрати часу на перемикання між системами, несвоєчасне оновлення статусів завдань та втрату контексту при обговоренні завдань.

Результати тестування показали, що інтеграція СУЗ та комунікації має значний позитивний вплив на ключові аспекти роботи ІТ-команд:

Вплив на контроль задач. У середньому частота оновлення статусів завдань зростає на 38%, а кількість випадків несвоєчасного оновлення зменшилася на 45%. Це значно покращило видимість проектів для менеджерів та дозволило оперативно реагувати на зміни у статусах завдань.

Вплив на процеси взаємодії. Найбільш суттєвим поліпшенням став зменшений час на перемикання між системами, який за суб'єктивними оцінками учасників скоротився приблизно вдвічі. Це особливо помітно при використанні інтеграції через Webhook яка забезпечувала миттєві оновлення.

Вплив на комунікацію. Кількість випадків втрати контексту при обговоренні завдань зменшилася на 36%. Учасники відзначили що можливість бачити деталі завдання безпосередньо в комунікаційній системі значно покращила якість обговорень та зменшила ймовірність непорозумінь.

Порівняльна оцінка трьох методів інтеграції показала, що кожен має свої особливості та оптимальні сфери застосування.

4.3 Рекомендації до застосування і можливості розширення

Вибір методу інтеграції має бути продуманим і враховувати низку факторів: специфіку команди, тип проекту, доступні технічні ресурси та очікувану швидкість впровадження. Наприклад, якщо команда невелика (до 10 осіб), а технічних спеціалістів для розробки чи налаштування складних рішень немає, найбільш логічним буде використання офіційного додатку з каталогу Microsoft Teams. Він дозволяє швидко розпочати роботу без потреби у глибокій технічній підготовці, що зручно для нетехнічних менеджерів і команд, яким важлива простота.

Коли мова йде про середні команди, особливо в динамічних проєктах, де часто змінюються пріоритети або задачі, доцільніше використовувати Webhook-інтеграцію. Такий підхід забезпечує майже миттєву реакцію системи на зміни, що критично для команд, які цінують оперативність. Водночас, для його налаштування потрібні базові технічні навички, особливо з точки зору безпеки.

Для великих команд і складних, довготривалих проєктів найбільш ефективною буде API-інтеграція. Вона дає змогу створити максимально гнучке та налаштоване під конкретні потреби рішення, особливо якщо необхідна поглиблена аналітика чи обробка даних. Проте варто пам'ятати, що впровадження такого рішення вимагає значних технічних ресурсів і супроводу зі сторони кваліфікованих розробників.

Незалежно від обраного підходу, сам процес впровадження інтеграції потребує системності. Доцільно розпочинати з аналізу внутрішніх процесів команди, щоб чітко визначити, що саме має покращити інтеграція. Після цього варто провести пілотне тестування з участю представників різних ролей у команді, що дозволить виявити проблеми до масштабного впровадження. Не менш важливо подбати про навчання користувачів і створення зручної документації, аби кожен учасник команди міг впевнено користуватись новим

інструментом. Обов'язково потрібно також передбачити механізми зворотного зв'язку та моніторингу ефективності інтеграції, адже тільки на основі реальних відгуків можна вдосконалювати рішення. В окремих випадках (наприклад, при роботі з Webhook чи API) особливу увагу слід приділити безпеці передачі даних.

Вибір методу інтеграції має бути продуманим і враховувати низку факторів: специфіку команди, тип проєкту, доступні технічні ресурси та очікувану швидкість впровадження. Наприклад, якщо команда невелика (до 10 осіб), а технічних спеціалістів для розробки чи налаштування складних рішень немає, найбільш логічним буде використання офіційного додатку з каталогу Microsoft Teams. Він дозволяє швидко розпочати роботу без потреби у глибокій технічній підготовці, що зручно для нетехнічних менеджерів і команд, яким важлива простота.

Коли мова йде про середні команди, особливо в динамічних проєктах, де часто змінюються пріоритети або задачі, доцільніше використовувати Webhook-інтеграцію. Такий підхід забезпечує майже миттєву реакцію системи на зміни, що критично для команд, які цінують оперативність. Водночас, для його налаштування потрібні базові технічні навички, особливо з точки зору безпеки.

Для великих команд і складних, довготривалих проєктів найбільш ефективною буде API-інтеграція. Вона дає змогу створити максимально гнучке та налаштоване під конкретні потреби рішення, особливо якщо необхідна поглиблена аналітика чи обробка даних. Проте варто пам'ятати, що впровадження такого рішення вимагає значних технічних ресурсів і супроводу зі сторони кваліфікованих розробників.

Незалежно від обраного підходу, сам процес впровадження інтеграції потребує системності. Доцільно розпочинати з аналізу внутрішніх процесів команди, щоб чітко визначити, що саме має покращити інтеграція. Після цього варто провести пілотне тестування з участю представників різних ролей у команді, що дозволить виявити проблеми до масштабного впровадження. Не

менш важливо подбати про навчання користувачів і створення зручної документації, аби кожен учасник команди міг впевнено користуватись новим інструментом. Обов'язково потрібно також передбачити механізми зворотного зв'язку та моніторингу ефективності інтеграції, адже тільки на основі реальних відгуків можна вдосконалювати рішення. В окремих випадках (наприклад, при роботі з Webhook чи API) особливу увагу слід приділити безпеці передачі даних.

Перспективи вдосконалення інтеграційних рішень є досить широкими. Одним із ключових напрямів є автоматизація дій між системами. Наприклад, інтеграція може не лише передавати дані між Jira та Teams, а й створювати задачі автоматично на основі певних повідомлень, планувати зустрічі або навіть формувати короткі зведення з обговорень. Розвиток API дозволяє розширювати ці можливості досить гнучко.

Крім того, важливим вектором розвитку є інтеграція не лише систем задач і комунікацій, але й таких інструментів, як GitHub, GitLab, сервіси безперервної інтеграції та безперервного розгортання (Continuous Integration/Continuous Deployment, CI/CD) або системи моніторингу. Це дозволить створити повноцінне інформаційне середовище для всього життєвого циклу розробки.

Серед перспективних напрямів розвитку інтеграцій варто виділити такі як розвиток аналітичних можливостей. Розширення функціоналу аналітики та звітності на основі даних, отриманих з інтегрованих систем, дозволить менеджерам проєктів отримувати більш глибоке розуміння процесів та приймати обґрунтовані рішення.

Впровадження елементів штучного інтелекту. Використання AI для аналізу даних з інтегрованих систем допоможе у прогнозуванні ризиків, оптимізації розподілу ресурсів і підвищенні ефективності управління.

Розвиток мобільних рішень. Створення мобільних версій інтеграційних рішень дозволить командам залишатися на зв'язку та керувати процесами навіть у дорозі — що актуально в умовах гнучкої роботи.

Підсумовуючи, можна сказати, що ефективна інтеграція СУЗ та комунікації — це не лише зручність, а й стратегічний крок для підвищення результативності ІТ-команд. Врахування особливостей кожного проекту та гнучкість у впровадженні створюють підґрунтя для сталого зростання ефективності та прозорості робочих процесів.

ВИСНОВКИ

У результаті проведеного дослідження методів інтеграції систем управління задачами та комунікації для підвищення ефективності виконання ІТ-проектів вирішено ряд важливих завдань та отримано суттєві результати.

Аналіз предметної області та опитування 31 фахівця ІТ-галузі підтвердили актуальність проблеми: 72% учасників витрачають на перемикання між Jira та Microsoft Teams від 15 до 30 хвилин щодня, а 67% розробників періодично забувають оновлювати статуси задач.

Детально досліджено три основні методи інтеграції: додаток з каталогу Microsoft Teams, Webhook-інтеграцію та API-інтеграцію. Визначено їх принципи роботи, технічні особливості, переваги та обмеження. Практичне тестування показало, що впровадження інтеграційних рішень дозволяє суттєво підвищити ефективність роботи команди: зменшити час на перемикання між системами, збільшити частоту оновлення статусів задач та зменшити кількість випадків втрати контексту.

Порівняльний аналіз методів за ключовими критеріями ефективності продемонстрував, що кожен має свої оптимальні сфери застосування. Додаток з каталогу Microsoft Teams рекомендовано для невеликих команд та проектів, що потребують швидкого впровадження. Webhook-інтеграція оптимальна для динамічних проектів з високою інтенсивністю змін. API-інтеграція найкраще підходить для великих команд та складних проектів, де потрібні розширені аналітичні можливості.

Визначено п'ять ключових рекомендацій щодо процесу впровадження інтеграції, що включають попередній аналіз робочих процесів, пілотне тестування, навчання команди, моніторинг ефективності та забезпечення безпеки даних. Також окреслено перспективні напрямки подальшого розвитку: розширення функціональності автоматизації, інтеграція з іншими інструментами розробки, розвиток аналітичних можливостей, впровадження

елементів штучного інтелекту та мобільних рішень.

Наукова новизна дослідження полягає у розробці комплексного підходу до інтеграції СУЗ та комунікації, який враховує як технічні аспекти різних методів, так і їх практичне застосування в реальних ІТ-проєктах. Практична цінність роботи визначається можливістю використання розроблених рекомендацій для підвищення ефективності виконання ІТ-проєктів.

Таким чином, проведене дослідження має суттєве теоретичне та практичне значення для сфери управління ІТ-проєктами та відкриває перспективи для подальших досліджень у напрямку інтеграції різноманітних інструментів розробки та управління проєктами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 17th State of Agile Report | Analyst Reports | Digital.ai. URL: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report> (дата звернення: 30.04.2025).
2. Ємельянова А.В. Інтеграція систем управління задачами та комунікації як фактор підвищення ефективності виконання іт-проектів. *Proceedings of VII International Scientific and Practical Conference* (Stockholm, Sweden, 19-21 May 2025) : зб. матеріалів конференції. Stockholm, 2025. С. 152–157.
3. Agile Statistics: 16 Amazing Agile Statistics | Zippia. URL: <https://www.zippia.com/advice/agile-statistics/> (дата звернення: 28.04.2025).
4. 20+ Agile Statistics: All About Agile Adoption | Runn. URL: <https://www.runn.io/blog/agile-statistics> (дата звернення: 29.04.2025).
5. Slack Statistics By Revenue, Usage, Industry Usage And Demographics | Electroiq. URL: <https://electroiq.com/stats/slack-statistics/> (дата звернення: 27.04.2025).
6. Project Management Statistics and Facts | Scoop Market. URL: <https://scoop.market.us/project-management-statistics/> (дата звернення: 28.04.2025).
7. Slack Statistics 2023 - Revenue And Usage Statistics | Enterprise Apps Today. URL: <https://www.enterpriseappstoday.com/stats/slack-statistics.html> (дата звернення: 29.04.2025).
8. Slack Statistics: Revenue & Users Data | Demandsage. URL: <https://www.demandsage.com/slack-statistics/> (дата звернення: 30.04.2025).
9. Is your team using too many productivity tools? | Mattermost. URL: <https://mattermost.com/blog/too-many-productivity-tools/> (дата звернення: 01.05.2025).
10. Microsoft Teams vs. Slack in 2024 | CRS Info Solutions. URL:

<https://www.crsinfosolutions.com/microsoft-teams-vs-slack-2024/> (дата
звернення: 01.05.2025).

11. Slack vs Teams: What's Right for 2024? | Tetra. URL:
<https://tetra.com/article/slack-vs-teams/> (дата звернення: 01.05.2025).

12. Zoom vs. Slack vs. Teams: The best collaboration tools for IT |
LogicMonitor. URL: [https://www.logicmonitor.com/blog/best-collaboration-tools-
for-it-ms-teams-vs-zoom-vs-slack](https://www.logicmonitor.com/blog/best-collaboration-tools-for-it-ms-teams-vs-zoom-vs-slack) (дата звернення: 01.05.2025).

13. How companies can improve workplace productivity | McKinsey. URL:
[https://www.mckinsey.com/capabilities/people-and-organizational-
performance/our-insights/some-employees-are-destroying-value-others-are-
building-it-do-you-know-the-difference](https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/some-employees-are-destroying-value-others-are-building-it-do-you-know-the-difference) (дата звернення: 01.05.2025).

14. State of SaaS - Productiv. URL: <https://productiv.com/state-of-saas/> (дата
звернення: 03.05.2025).

15. Key Slack Statistics in 2024 | Axolo. URL: [https://axolo.co/blog/p/key-
slack-statistics-in-2024](https://axolo.co/blog/p/key-slack-statistics-in-2024) (дата звернення: 01.05.2025).

16. Slack Statistics (2025) – Revenue & Users Data | Demandsage. URL:
<https://www.demandsage.com/slack-statistics/> (дата звернення: 01.05.2025).

17. Integrate Jira with Slack: Complete Guide for 2024 | Bardeen. URL:
<https://www.bardeen.ai/integrations/jira/slack> (дата звернення: 01.05.2025).

18. Confluence: Collaborate on shared documentation | Atlassian. URL:
<https://www.atlassian.com/software/confluence> (дата звернення: 01.05.2025).

19. Tools used by software developers worldwide 2022 | Statista. URL:
[https://www.statista.com/statistics/869106/worldwide-software-developer-survey-
tools-in-use/](https://www.statista.com/statistics/869106/worldwide-software-developer-survey-tools-in-use/) (дата звернення: 02.05.2025).

20. UX Tools - Design Tools Survey - About This Report. URL:
<https://www.uxtools.co/survey/introduction/about-this-report> (дата
звернення: 02.05.2025).

21. 54 Essential Slack Statistics: 2024 Market Share Analysis & Data |
FinancesOnline. URL: <https://financesonline.com/slack-statistics/> (дата звернення:
01.05.2025).

22. Buffer | State Of Remote Work 2023. URL: <https://buffer.com/state-of-remote-work/2023> (дата звернення: 02.05.2025).

23. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня програми «Інформаційні управляючі системи та технології») / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. – Харків: ХНУРЕ, 2021. – 30с.

24. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

25. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.

26. Jira Software: Issue and Project Tracking for Software Teams | Atlassian. URL: <https://www.atlassian.com/software/jira> (дата звернення: 19.05.2025).

27. Top 12 Project Management Methodologies | Smartsheet. URL: https://www.smartsheet.com/project-management%20guide/methodologies?srsltid=AfmBOooGczTYkTIDiY1nV_schDPfghSqdOJJI1bCx6_-YU0940j1skwp (дата звернення: 19.05.2025).

28. Kinsella, P. Effective Communication in Project Management. *Project Smart*. 2021. № 2. С. 54–65.

29. What is Agile? | Atlassian. URL: <https://www.atlassian.com/agile> (дата звернення: 19.05.2025).

30. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition and The Standard for Project Management. Newtown Square, Pennsylvania. *Project Management Institute, Inc.*, 2021. 370 с.