



УДК 004.8:005.8

**ВПРОВАДЖЕННЯ ШІ-АГЕНТІВ ДЛЯ АВТОМАТИЗАЦІЇ  
ПРІОРИТИЗАЦІЇ ЗАВДАНЬ В СИСТЕМАХ УПРАВЛІННЯ  
ПРОЄКТАМИ**

*Наталія Полуктова,  
Кафедра інформаційних технологій  
Запорізький інститут економіки та інформаційних технологій  
Запоріжжя, Україна  
Станіслав Левицький  
Кафедра інформаційних технологій  
Запорізький інститут економіки та інформаційних технологій  
Запоріжжя, Україна*

**Анотація** – У статті розглянуто проблему автоматизації пріоритизації завдань у системах управління проектами розробки програмного забезпечення. Традиційні значною мірою залежать від суб'єктивних оцінок менеджерів і команди та не завжди враховують динамічність середовища розробки. У зв'язку з цим актуальним є використання інтелектуальних агентів та методів штучного інтелекту для автоматизації процесу пріоритизації завдань.

Метою дослідження є розробка та тестування інтелектуального агента для системи управління проектами Redmine, який автоматизує визначення пріоритетів завдань на основі методу Weighted Shortest Job First (WSJF). У роботі запропоновано архітектуру агента, що включає модуль збору даних через API системи управління проектами, модуль алгоритмічної обробки для розрахунку коефіцієнтів WSJF, модуль штучного інтелекту для аналізу ризиків та прогнозування затримок, а також інтерфейс взаємодії з користувачами.

У межах дослідження було реалізовано прототип плагіну для Redmine, який використовує користувацькі поля Business Value, Time Criticality, Risk Reduction та Job Size для автоматичного розрахунку коефіцієнтів пріоритетності завдань. Проведене тестування на експериментальному наборі даних показало коректність роботи алгоритму, зручність інтеграції з інтерфейсом Redmine та можливість масштабування рішення для проектів із великою кількістю завдань. Отримані результати підтверджують, що використання інтелектуальних агентів дозволяє підвищити об'єктивність та ефективність процесу управління backlog'ом, зменшити трудомісткість ручної пріоритизації та забезпечити прозорість прийняття рішень.

Перспективи подальших досліджень пов'язані з інтеграцією методів машинного навчання для адаптивного коригування пріоритетів на основі історичних даних виконання завдань, а також із розширенням функціональності агента для використання в інших системах управління проектами.

**Ключові слова** - управління проектами, інтелектуальні агенти, штучний інтелект, пріоритизація завдань, WSJF, Redmine

**І.ВСТУП**

Управління завданнями є одним із ключових аспектів сучасних систем управління

проектами розробки програмного забезпечення. Зростання складності проектів, багатокомандна взаємодія та необхідність швидкої адаптації до змін бізнес-середовища створюють потребу у нових підходах до пріоритизації завдань.

Пріоритизація завдань у системах управління проектами розробки програмного забезпечення базується на класичних методах, які дозволяють структурувати процес прийняття рішень. Серед найбільш поширених підходів можна виділити метод MoSCoW, що класифікує завдання на категорії Must have, Should have, Could have та Won't have, забезпечуючи чітке розмежування між критично важливими та другорядними роботами [1] Інший метод — WSJF (Weighted Shortest Job First) дозволяє визначати пріоритети на основі співвідношення бізнес-цінності, терміновості та ризиків до розміру завдання, що забезпечує більш обґрунтоване планування [2]. Метод Value vs. Effort Matrix пропонує просту двовимірну модель, де завдання оцінюються за цінністю для бізнесу та необхідними зусиллями, що дозволяє швидко виділити «швидкі перемоги» та стратегічні ініціативи [3].

Попри очевидні переваги традиційних методів — простоту, зрозумілість та можливість швидкого застосування — вони мають і суттєві обмеження. По-перше, вони значною мірою залежать від суб'єктивних оцінок менеджерів та команди, що може призводити до перекосів у пріоритизації. По-друге, ці методи часто не враховують динамічність середовища розробки, де бізнес-цінність та ризики змінюються у реальному часі. По-третє, у великих проєктах із тисячами завдань ручна пріоритизація стає надто трудомісткою та неефективною.

Саме тому сучасні дослідження підкреслюють необхідність автоматизації процесів пріоритизації, особливо у великих командах та масштабних проєктах. Використання інтелектуальних агентів та алгоритмів штучного інтелекту дозволяє зменшити суб'єктивність, забезпечити адаптивність до змінних умов та оптимізувати розподіл ресурсів.

У цьому контексті дедалі більшої актуальності набувають інтелектуальні агенти, які здатні автономно аналізувати дані, приймати рішення та виконувати дії без постійного втручання людини. Вони не лише автоматизують рутинні операції, але й забезпечують адаптивність та прозорість управління, що особливо важливо у великих ІТ-проєктах.

Дослідження показують, що використання штучного інтелекту в управлінні проєктами дозволяє підвищити точність прогнозування ризиків, оптимізувати розподіл ресурсів та забезпечити більш обґрунтоване прийняття рішень. Інтелектуальні агенти стають не лише інструментом автоматизації, а й стратегічним елементом цифрової трансформації бізнес-процесів, здатним змінити підхід до управління завданнями у розробці програмного забезпечення.

Проблема ефективної пріоритизації завдань у великих проєктах розробки програмного забезпечення давно привертає увагу дослідників.

У роботі [4] автори підкреслюють, що традиційні методи управління backlog'ом часто не враховують динамічні зміни бізнес-цінності та ризиків, що призводить до неефективного використання ресурсів у великих організаціях. Інший підхід представлено у статті К.Лі та ін.[5], де пропонується використання багатоагентних систем для автоматичного розподілу завдань у середовищах із високою динамікою. Автори показують, що інтелектуальні агенти здатні оптимізувати процеси за рахунок автономного прийняття рішень та адаптації до зміни умов. У оглядовій статті З.Рахіда та ін [6] розглядається потенціал використання великих мовних моделей (LLM) як інтелектуальних агентів у розробці ПЗ. Автори стверджують, що такі агенти можуть не лише автоматизувати рутинні процеси, але й забезпечувати пояснюваність рішень, що особливо важливо для командної роботи та прозорості управління.

Безпосередньо дослідженням у сфері застосування ШІ для пріоритизації завдань у

проектному менеджменті присв'ячена робота С.А. Закарії та ін [7], де автори показують, як багатоагентні системи можуть оптимізувати розподіл ресурсів та завдань у складних середовищах, використовуючи глибоке підкріплювальне навчання. Автори роботи [8] демонструють, що LLM здатні перетворювати високорівневі інструкції у конкретні завдання та автоматично визначати їх пріоритети, що відкриває нові можливості для інтеграції ШІ-агентів у системи управління проектами.

У статті [9] розглядається концептуальна основа для створення агентів, які можуть поєднувати машинне навчання з символічними методами. Це дозволяє агентам не лише виконувати завдання, але й пояснювати свої рішення, що є критично важливим для прозорості процесів пріоритизації у командній роботі

Таким чином, сучасні дослідження демонструють перехід від статичних методів пріоритизації до динамічних, агентно-орієнтованих систем, які інтегрують алгоритмічні моделі (WSJF, Value vs. Effort) із можливостями штучного інтелекту. У систематичному огляді [10] підкреслюється, що інтеграція AI у процеси управління проектами сприяє більш точному прогнозуванню ризиків та оптимізації ресурсів. Це відкриває перспективи для створення інтелектуальних агентів, здатних не лише автоматизувати процеси, але й підвищувати стратегічну ефективність управління проектами.

## II. МОДЕЛІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

Інтелектуальний агент у контексті управління завданнями можна визначити як програмний модуль, здатний автономно збирати дані, аналізувати їх, формувати рекомендації та виконувати автоматичні дії без постійного втручання людини. Його головна мета полягає у зменшенні суб'єктивності процесу пріоритизації та забезпеченні адаптивності до змінних умов проекту.

Архітектура такого агента складається з кількох взаємопов'язаних модулів. Першим є модуль збору даних, який інтегрується з API систем управління проектами, таких як Jira, Trello, Asana або Redmine та отримує інформацію про backlog, статуси завдань, дедлайни й оцінки команди. Далі працює модуль обробки, що застосовує формальні алгоритми пріоритизації, зокрема WSJF (Weighted Shortest Job First) або Value vs. Effort Matrix, для розрахунку пріоритетів на основі бізнес-цінності, терміновості, ризиків та ресурсних витрат.

Ключовим елементом є модуль штучного інтелекту, який використовує методи машинного навчання для прогнозування ризиків, виявлення потенційних затримок та оптимізації розподілу завдань між членами команди. Він може враховувати історичні дані про виконання завдань, продуктивність окремих розробників та залежності між задачами. Нарешті, інтерфейс взаємодії забезпечує комунікацію агента з користувачами — це може бути чат-бот у Slack чи Microsoft Teams, або інтегрована панель, де агент надає рекомендації, пояснює свої рішення та автоматично оновлює статуси завдань.

Таким чином, архітектура інтелектуального агента поєднує збір даних, алгоритмічну обробку, інтелектуальний аналіз та інтерактивну взаємодію, створюючи комплексний інструмент для автоматизації пріоритизації завдань у сучасних системах управління проектами.

## III. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Завданням для практичної частини цього дослідження було створення інтелектуального агента для системи управління проектами Redmine, який автоматизує процес пріоритизації завдань на основі методу WSJF. Було розроблено прототип

програмного модуля, здатного інтегруватися у Redmine, збирати дані про завдання, виконувати алгоритмічний аналіз та формувати обґрунтовані рекомендації щодо порядку їх виконання.

Архітектура агента будується як плагін до Redmine, що складається з кількох взаємопов'язаних компонентів.

1. Першим є модуль збору даних, який взаємодіє з API Redmine та отримує інформацію про завдання, включно з користувацькими полями Business Value, Time Criticality, Risk Reduction та Job Size.

2. Другим є модуль обробки, що реалізує алгоритм WSJF і виконує сортування backlog'у відповідно до розрахованих коефіцієнтів.

3. Третім є модуль штучного інтелекту, який може бути реалізований як додатковий сервіс для прогнозування ризиків та адаптації пріоритетів на основі історичних даних.

4. Нарешті, інтерфейс взаємодії забезпечує відображення результатів у Redmine, оновлення пріоритетів у завданнях та формування звітів для менеджерів.

Опис агента передбачає його функціонування як автономного інструмента, що інтегрується у робочі процеси команди. Він отримує дані про завдання, розраховує WSJF, формує список пріоритетів та надає користувачам прозорі пояснення щодо прийнятих рішень.

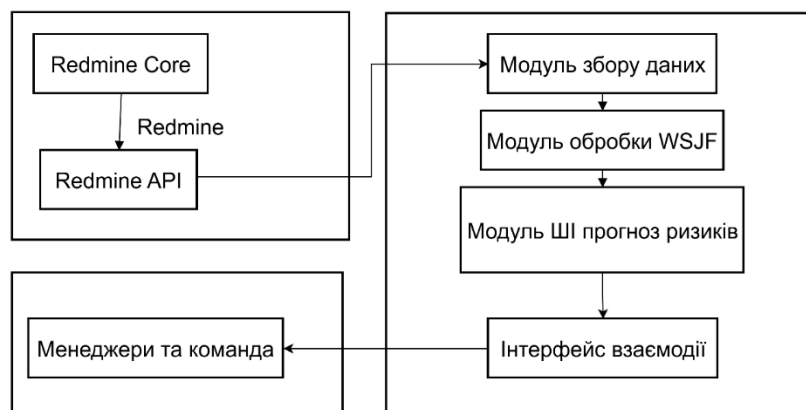


Рис. 1. Архітектура інтелектуального агента в Redmine для пріоритизації завдань

Особливості реалізації полягають у використанні стандартних механізмів розширення Redmine.

Плагін був створений у директорії plugins, де визначається файл init.rb із базовою конфігурацією. Для зберігання параметрів WSJF у Redmine створюються custom fields, які прив'язуються до завдань. Алгоритм WSJF реалізується у вигляді окремого класу, що виконує розрахунки та сортування backlog'у.

Технологія розрахунку WSJF у плагіні для Redmine базується на інтеграції алгоритмічного модуля у структуру системи управління завданнями. Основна ідея полягає у тому, що кожне завдання отримує набір параметрів, які зберігаються у спеціальних користувацьких полях: Business Value (бізнес-цінність), Time Criticality (терміновість), Risk Reduction (зменшення ризиків) та Job Size (розмір завдання). Ці поля створюються адміністратором Redmine і стають доступними для кожного елемента типу Issue.

Плагін реалізує клас, який отримує значення цих полів через внутрішні моделі Redmine.

Далі виконується формула WSJF:

$$JSWSJF = \frac{BV+TC+RR}{JS}, \quad [1]$$

де

Business Value (BV) — це показник бізнес-цінності завдання. Він відображає, наскільки виконання конкретної задачі принесе користь організації, клієнтам чи користувачам. Наприклад, завдання, що безпосередньо впливає на прибуток, задоволеність клієнтів або стратегічні цілі компанії, отримує високий BV. У Redmine цей параметр може задаватися як числове поле, яке оцінюється менеджером або продуктовим власником.

Time Criticality (TC) — це показник терміновості. Він показує, наскільки важливо виконати завдання у визначений час. Якщо затримка у виконанні призведе до втрати можливостей, штрафів чи зниження конкурентоспроможності, TC буде високим. У Redmine цей параметр також задається у вигляді числового поля, що дозволяє встановлювати часові обмеження та дедлайни.

Risk Reduction (RR) — це показник зменшення ризиків або створення додаткових можливостей. Він відображає, наскільки виконання завдання допоможе уникнути потенційних проблем у майбутньому або створить нові перспективи для розвитку. Наприклад, завдання, що усуває технічний борг, знижує ймовірність критичних помилок або відкриває шлях до нових функцій, має високий RR. У Redmine цей параметр також реалізується як числове поле.

У сумі ці три складові формують сукупну вигоду від виконання завдання, яка ділиться на Job Size (JS) — показник трудомісткості чи складності. Таким чином, WSJF дозволяє об'єктивно визначити, які завдання приносять найбільшу користь при найменших витратах, і забезпечує прозору та обґрунтовану пріоритизацію у плагіні для Redmine.

Алгоритм реалізується як метод у класі плагіну, який перебирає всі завдання у backlog'у, розраховує WSJF для кожного з них та формує відсортований список. Цей список може бути відображений у спеціальній вкладці Redmine або використаний для автоматичного оновлення поля Priority у завданнях. Для інтеграції з інтерфейсом користувача застосовуються стандартні механізми Rails: контролери для обробки запитів, моделі для доступу до даних та представлення для візуалізації результатів.

Таким чином, технологія розрахунку WSJF у плагіні для Redmine поєднує створення custom fields, використання внутрішніх моделей системи, реалізацію алгоритмічного класу для обчислень та інтеграцію з інтерфейсом користувача. Це забезпечує прозору, обґрунтовану та автоматизовану пріоритизацію завдань, яка зменшує суб'єктивність рішень і підвищує ефективність управління проектами.

Інтеграція з Redmine API забезпечує доступ до даних та можливість оновлення пріоритетів. Інтерфейс користувача реалізується як додаткова вкладка у меню проекту, де відображається таблиця із завданнями та їхніми WSJF-значеннями.

Опціонально додається REST-ендпоінт для зовнішньої взаємодії, що дозволяє інтегрувати агент із іншими системами. Завдяки такій архітектурі плагін є гнучким, масштабованим і здатним адаптуватися до потреб різних команд, забезпечуючи прозору та обґрунтовану пріоритизацію завдань у Redmine.

У тестовому середовищі було створено проєкт із двадцятьма завданнями, кожне з яких мало заповнені користувацькі поля Business Value, Time Criticality, Risk Reduction та Job Size. Після запуску плагіну агент автоматично розрахував коефіцієнти WSJF для кожного завдання та сформував відсортований список пріоритетів.

Результати показали, що завдання з високою бізнес-цінністю та низькою трудомісткістю отримали найвищі пріоритети. Наприклад, завдання «Інтеграція платіжного модуля» мало BV=8, TC=7, RR=6, JS=2, що дало WSJF=10.5 і поставило його на перше місце у списку. Завдання «Оптимізація внутрішнього звіту» мало BV=3, TC=2,

RR=1, JS=5, що дало WSJF=1.2 і віднесло його до нижнього сегмента backlog'у.

Тестування також показало коректність роботи інтерфейсу: у вкладці «WSJF Prioritization» відображався список завдань із розрахованими коефіцієнтами, а при натисканні кнопки «Оновити пріоритети» поле Priority у Redmine змінювалося відповідно до нового порядку.

Додатково було проведено навантажувальне тестування на проєкті із понад тисячею завдань. Плагін виконав розрахунки менш ніж за 10 секунд, що підтвердило його масштабованість. У тестах із використанням історичних даних модуль ШІ коригував пріоритети для завдань, які мали високий ризик затримки, що підвищило точність прогнозів.

#### IV. ВИСНОВКИ

Результати проведеного дослідження та тестування плагіну для Redmine показують, що інтеграція алгоритму WSJF у систему управління проєктами забезпечує прозору та обґрунтовану пріоритизацію завдань. Використання користувацьких полів для Business Value, Time Criticality, Risk Reduction та Job Size дозволяє формалізувати процес оцінки задач, зменшити суб'єктивність рішень та підвищити ефективність роботи команди. Тестові результати підтвердили коректність розрахунків, масштабованість рішення та зручність інтерфейсу, що робить плагін придатним для використання як у невеликих командах, так і у великих організаціях.

#### V. ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Перспективи подальших досліджень полягають у розширенні функціоналу агента за рахунок інтеграції методів машинного навчання для прогнозування ризиків та адаптації пріоритетів у реальному часі. Доцільним є також дослідження можливостей інтеграції WSJF-плагіну з іншими системами управління проєктами, такими як Jira, Trello чи Asana, для забезпечення універсальності рішення. Окремим напрямом розвитку може стати створення інтелектуальних рекомендаційних систем, що враховують історичні дані виконання завдань, продуктивність команди та зовнішні фактори. Таким чином, подальші дослідження мають на меті перетворити плагін із інструмента автоматизації на повноцінного інтелектуального помічника, здатного підтримувати стратегічне управління проєктами та підвищувати їхню результативність.

#### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Miranda E. Moscow Rules: A Quantitative Exposé // *Agile Processes in Software Engineering and Extreme Programming* / V. Stray, K. J. Stol, M. Paasivaara, P. Kruchten (eds.). – Cham: Springer, 2022. – Vol. 445. – P. 21–35. [https://doi.org/10.1007/978-3-031-08169-9\\_2](https://doi.org/10.1007/978-3-031-08169-9_2)

[2] Knaster R., Leffingwell D. *SAFe 4.5 Distilled: Applying the Scaled Agile Framework for Lean Software and Systems Engineering*. – Boston: Addison-Wesley Professional, 2018.

[3] McGrath R. G. *The End of Competitive Advantage: How to Keep Your Strategy Moving as Fast as Your Business*. – Boston: Harvard Business Review Press, 2013.

[4] Bugayenko Y., Farina M., Kruglov A., Pedrycz W., Plaksin Y., Succi G. Automatically prioritizing tasks in software development // *IEEE Access*. – 2023. – Vol. 11. – P. 90322–90334. <https://doi.org/10.1109/ACCESS.2023.3305249>

[5] Li K., Hu Y., Wang J., Li L., Zhang S., Luo G. Task prioritization in multiagent environments: A novel approach using Nash Q-learning // IEEE Transactions on Consumer Electronics. – 2025. – Vol. 71, № 2. – P. 4206–4220. <https://doi.org/10.1109/TCE.2025.3542833>

[6] Rasheed Z. та ін. Autonomous agents in software development: A vision paper // Agile Processes in Software Engineering and Extreme Programming – Workshops / L. Marchesi та ін. (eds.). – Cham: Springer, 2025. – Vol. 524. – P. 15–27. [https://doi.org/10.1007/978-3-031-72781-8\\_2](https://doi.org/10.1007/978-3-031-72781-8_2)

[7] Zakaryia S. A., Meaad M., Nabil T. та ін. Task offloading and resource allocation for multi-UAV asset edge computing with multi-agent deep reinforcement learning // Computing. – 2025. – Vol. 107. – P. 126. <https://doi.org/10.1007/s00607-025-01472-5>

[8] Kannan S. S., Venkatesh V. L. N., Min B.-C. SMART-LLM: Smart multi-agent robot task planning using large language models // Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – Abu Dhabi, 2024. – P. 12140–12147. <https://doi.org/10.1109/IROS58592.2024.10802322>

[9] Mao J., Tenenbaum J. B., Wu J. Building intelligent agents with neuro-symbolic concepts // Communications of the ACM. – 2026. – Vol. 69, № 2. – P. 69–79. <https://doi.org/10.1145/3715316>

[10] Adamantiadou D. S., Tsironis L. Leveraging artificial intelligence in project management: A systematic review of applications, challenges, and future directions // Computers. – 2025. – Vol. 14. – P. 66. <https://doi.org/10.3390/computers14020066>

**Отримано 18.12.2025 р.**

## **IMPLEMENTATION OF AI AGENTS FOR AUTOMATING TASK PRIORITIZATION IN PROJECT MANAGEMENT SYSTEMS**

*Nataliya Poluektova*  
*Department of Information Technologies*  
*Zaporizhzhia Institute of Economics and Information Technologies*  
*Zaporizhzhia, Ukraine*

*Stanislav Levytsky*  
*Department of Information Technology*  
*Zaporizhzhia Institute of Economics and Information Technologies*  
*Zaporizhzhia, Ukraine*

**Abstract** - The article addresses the problem of automating task prioritization in software development project management systems. Traditional prioritization approaches largely depend on subjective assessments made by project managers and team members and do not always take into account the dynamic nature of the development environment. In this regard, the use of intelligent agents and artificial intelligence methods for automating the task prioritization process becomes increasingly relevant.

The purpose of the research is to develop and test an intelligent agent for the Redmine project management system that automates task priority determination based on the Weighted Shortest Job First (WSJF) method. The paper proposes an agent architecture that includes a data collection module using the project management system API, an algorithmic processing module for calculating WSJF coefficients, an artificial intelligence module for risk analysis and delay prediction, and a user interaction interface.

Within the scope of the study, a prototype plugin for Redmine was implemented, which uses custom fields such as Business Value, Time Criticality, Risk Reduction, and Job Size to automatically calculate task priority coefficients. Testing conducted on an experimental dataset demonstrated the correctness of the algorithm, the convenience of integration with the Redmine interface, and the possibility of scaling the solution for projects with a large number of tasks. The obtained results confirm that the use of intelligent agents can improve the objectivity and efficiency of backlog management, reduce the effort required for manual prioritization, and ensure transparency in decision-making.

Future research prospects include the integration of machine learning methods for adaptive priority adjustment based on historical task execution data, as well as expanding the agent's functionality for use in other project management systems

***Keywords:** project management, intelligent agents, artificial intelligence, task prioritization, WSJF, Redmine.*

**Recieved 18.12.2025 p.**