

**Кордунов С.Ю.,**

старший викладач кафедри системного аналізу та кібербезпеки,  
КНЕУ імені Вадима Гетьмана

**Kordunov S.Yu.,**

Senior Lecturer, Department of System Analysis and Cybersecurity  
KNEU named after V. Hetman

## ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ В ПРОЦЕСИ УПРАВЛІННЯ ІТ-ПРОЄКТАМИ

## INTEGRATION OF ARTIFICIAL INTELLIGENCE INTO IT PROJECT MANAGEMENT PROCESSES

**Abstract.** *Purpose.* The article aims at developing a comprehensive methodology and four-layer architecture for deep integration of artificial intelligence into IT project management processes, ensuring the transition from reactive to predictive and prescriptive management of the full project lifecycle with regard to modern Agile/DevOps practices, DORA metrics, microservice and cloud-native architectures, and the dominant tooling ecosystem of 2025–2026.

*Scientific novelty.* The novelty lies in the systematisation of approaches to using Git data, CI/CD logs, infrastructure telemetry and static code analysis results as a unified multimodal training source for AI models, as well as in the creation of a seamless four-layer architecture that provides coordinated interaction of predictive (gradient boosting, LSTM, Graph Neural Networks), prescriptive (Reinforcement Learning agents) and generative (RAG-augmented CodeLlama) components with existing CI/CD pipelines and project management systems (Jira, GitLab, Azure DevOps). A new integral efficiency metric based on DORA indicators is proposed for the first time.

*Results.* A modular architecture has been developed that aggregates engineering artefacts in real time through a specialised ETL and feature store layer, performs multimodal analytical processing, generates prescriptive recommendations and automatically creates Infrastructure-as-Code manifests. Modelling demonstrated that sequential implementation of the proposed solution enables IT teams to reach elite performance level according to all four DORA metrics (cycle time < 1 day, lead time for changes < 1 day, deployment frequency > 100/month, MTTR < 1 hour) within 6–10 months. Comparative analysis confirmed 26–31 % improvement in cycle time forecasting accuracy, automation of technical debt management and introduction of previously unavailable functions — deployment risk score calculation and context-aware IaC generation.

*Conclusions.* The developed methodology and architecture eliminate the superficial nature of existing AI assistants and create a unified predictive-prescriptive IT project management system that can be implemented incrementally using open APIs of modern tools. The results open the way for real transition of DevOps teams to autonomous lifecycle management with continuous self-improvement of models.

**Keywords:** artificial intelligence, IT project management, deep integration, predictive analytics, prescriptive analytics, DORA metrics, DevOps, microservices, Infrastructure as Code, explainable AI.

**Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими або практичними завданнями.** Управління ІТ-проектами суттєво відрізняється від проектного менеджменту в інших галузях через високу невизначеність технічних вимог, швидку зміну технологічного стеку, необхідність безперервної інтеграції та доставки коду, а також складність точного оцінювання трудомісткості завдань у story points. Домінуючі в ІТ-індустрії Agile-методології (Scrum, Kanban, SAFe) значною мірою спираються на експертні оцінки команди, історичні дані velocity та ручне планування спринтів, що призводить до надмірних витрат часу на планувальні церемонії й систематичних відхилень від запланованих термінів та бюджету.

Сучасні інструменти управління ІТ-проектами (Jira, Azure DevOps, GitLab, Linear, ClickUp) вже містять окремі модулі штучного інтелекту, які автоматизують рутинні операції, зокрема автозаповнення описів задач і рекомендації виконавців. Однак ці рішення залишаються поверхневими й не охоплюють глибокої ІТ-специфіки: прогнозування cycle time та lead time for changes на основі семантичного аналізу Git-комітів і динаміки pull request-ів, автоматична класифікація дефектів за критичністю через мультимодальний NLP-аналіз описів, стеків трасування та логів, а також оптимізація розподілу розробників між задачами з урахуванням їх індивідуальних патернів code ownership, code churn і code review velocity.

Особливо гостро стоїть проблема управління технічним боргом у великих кодових базах, де ручний аналіз code smells, застарілих залежностей і потенційних вразливостей став практично неможливим без застосування моделей машинного навчання, інтегрованих зі статичними аналізаторами коду. У контексті DevOps-культури критичного значення набуває вбудована предиктивна аналітика в CI/CD-пайплайни, здатна оцінювати deployment risk score, прогнозувати ймовірність rollback-ів, виявляти аномалії в інфраструктурних метриках і автоматично обирати оптимальні стратегії розгортання (canary, blue-green).

Зростання складності мікросервісних і cloud-native архітектур додатково ускладнює задачу одночасної оптимізації термінів розробки та операційних характеристик системи (latency, throughput, витрати на хмарні ресурси). Штучний інтелект відкриває можливості автоматизованого прийняття рішень щодо стратегій autoscaling, вибору архітектурних патернів і навіть генерації інфраструктурного коду у форматі Infrastructure as Code.

Таким чином, актуальність дослідження зумовлена відсутністю систематизованої методології глибокої інтеграції штучного

інтелекту саме в процесі управління IT-проектами з урахуванням сучасних Agile/DevOps-практик, DORA-метрик, особливостей мікросервісних архітектур та домінуючого інструментарію 2025–2026 років, що й визначає мету та завдання даної роботи.

**Аналіз останніх досліджень і публікацій, в яких започатковано розв’язання проблеми, що висвітлюється, і на які спирається автор.** Питання інтеграції штучного інтелекту в управління проектами активно розробляється українськими та іноземними дослідниками у 2025 році. Бушуєв С. Д., Ільїн О. В., Пузійчук А. М., Лященко Т. В. [1] запропонували методологічні основи інтеграції ШІ в бази знань управління інноваційними проектами, акцентуючи увагу на використанні машинного навчання, обробки природної мови та аналізу великих даних для автоматизації прийняття рішень і оптимізації ресурсів.

Гайдаєнко О. В., Серік О. А. [2] розробили інтегровану FRI модель управління змістом IT-проектів у банківській сфері, показавши необхідність автоматизації збору вимог і контролю обсягу робіт, а також окресливши перспективи використання генеративного штучного інтелекту для цих процесів.

Крискун І. І., Орлова-Курилова О. В. [3] та Насад Н. В., Крискун І. М. [4] дослідили застосування інструментів штучного інтелекту для керування ризиками та загальних засад управління IT-проектами, підкресливши перехід від реактивних до прогнозно-адаптивних моделей завдяки градієнтному бустинку, нейронним мережам і великим мовним моделям.

Новак О. О., Гуржій В. О. [5] на прикладі платформи Aladdin компанії BlackRock проаналізували досвід глибокої інтеграції ШІ в планування, моніторинг і управління ризиками великих інвестиційних проектів, що стало важливим орієнтиром для розуміння архітектурних рішень корпоративного рівня.

Ноздріна Л. В. [6] експериментально довела ефективність використання генеративного та агентного штучного інтелекту (зокрема Google NotebookLM) як «копілота» в Agile/Scrum-проектах IT-галузі для автоматизації беклогу, пріоритизації та розподілу задач, отримавши скорочення часу планування з годин до хвилин при високій якості рекомендацій.

Пономарьов О. В., Поліщук О. В. [7] розглянули інтеграцію інструментів ШІ в адміністративний менеджмент і управління проектами загалом, визначивши ключові напрями автоматизації рутинних процесів в організаціях.

Серед закордонних досліджень Bushuyev S. та співавтори [8] акцентували на розширенні компетенцій менеджерів IT-проектів в

середовищі штучного інтелекту. Das J. K. та співавтори [9] проаналізували ефективність сучасних ШІ-інструментів управління проєктами для забезпечення їх успішності й термінів виконання. Nabil A. R. та співавтори [10] детально висвітлили етичні та правові аспекти впровадження ШІ в ІТ-проєктах (зміщення алгоритмів, конфіденційність даних, governance). Sakhawat Hussain Tanim, Md Sabbir Ahmad [11] запропонували підходи до стратегічного прийняття рішень на основі ШІ для покращення оцінки ризиків, контролю витрат і загальної ефективності ІТ-проєктів.

Отже, аналіз літератури свідчить про активний розвиток окремих напрямів застосування штучного інтелекту в управлінні ІТ-проєктами: від автоматизації планування та оцінки ризиків до етичних і правових аспектів. Водночас більшість робіт має або загально-проєктний характер, або зосереджена на ізольованих функціях (прогнозування velocity, класифікація багів, рекомендації в Jira). Відсутня комплексна методологія глибокої інтеграції ШІ саме в ІТ-специфічні процеси (CI/CD-пайплайни, управління технічним боргом, оптимізація DORA-метрик, autoscaling та IaC-генерація) з урахуванням сучасного інструментарію та DevOps-практик 2025–2026 років, що й визначає наукову новизну даної роботи.

**Виділення невирішених раніше частин загальної проблеми, яким присвячується стаття.** Попри значний прогрес у дослідженнях застосування штучного інтелекту в управлінні проєктами, залишаються недостатньо вирішеними питання глибокої інтеграції ШІ саме в ІТ-специфічні процеси та інструменти сучасної програмної інженерії. Зокрема, відсутня систематизована архітектура, що забезпечує безшовне поєднання предиктивних і прескриптивних ШІ-моделей із CI/CD-пайплайнами, системами контролю версій та управління задачами (Jira, GitLab, Azure DevOps) з урахуванням повного набору DORA-метрик та особливостей мікросервісних і cloud-native архітектур.

Недостатньо розробленими є методи використання Git-комітів, логів інфраструктури, динаміки pull request-ів та історичних даних про технічний борг як єдиних джерел навчання мультимодальних моделей для прогнозування cycle time, lead time for changes, deployment frequency та MTTR. Практично не досліджено підходи до автоматизованого управління технічним боргом у великих кодових базах через інтеграцію моделей машинного навчання зі статичними аналізаторами та автоматичне формування рекомендацій з рефакторингу безпосередньо в процесі code review.

Відсутні комплексні рішення для предиктивного розрахунку deployment risk score, автоматичного вибору стратегій розгортання

(canary/blue-green/rolling) та генерації Infrastructure as Code на основі аналізу реальних операційних метрик (latency, throughput, cost) у реальному часі. Також не сформовано методологію переходу від реактивного до прескриптивного управління ІТ-проєктами з використанням reinforcement learning та агентного ШІ в умовах високої динаміки вимог і плинності команд.

Таким чином, невирішеними залишаються завдання створення єдиної методології та архітектурного рішення, які б забезпечували глибоку, а не поверхневу інтеграцію штучного інтелекту в повний життєвий цикл управління ІТ-проєктами з урахуванням сучасних DevOps-практик, метрик продуктивності команд та домінуючого інструментарію 2025–2026 років.

**Формулювання цілей статті.** Метою статті є розробка комплексної методології та архітектурного рішення глибокої інтеграції штучного інтелекту в процеси управління ІТ-проєктами з урахуванням специфіки сучасних Agile/DevOps-практик, DORA-метрик, мікросервісних архітектур та інструментарію 2025–2026 років. Стаття спрямована на подолання поверхневого характеру наявних ШІ-функцій у системах Jira, GitLab, Azure DevOps і перехід до предиктивного та прескриптивного управління повним життєвим циклом ІТ-проєкту: від планування спринтів і оцінки cycle/lead time до автоматичного управління технічним боргом, розрахунку deployment risk score та генерації Infrastructure as Code.

Наукова новизна полягає в систематизації підходів до використання Git-даних, логів інфраструктури та операційних метрик як єдиного джерела навчання мультимодальних моделей, а також у створенні єдиної архітектури, що забезпечує безшовну взаємодію предиктивних, прескриптивних і агентних ШІ-компонентів із CI/CD-пайплайнами та системами управління задачами. Досягнення поставленої мети вимагало вирішення низки взаємопов'язаних технічних завдань, серед яких особливе місце посідає створення єдиної системи збору та нормалізації інженерних артефактів, що генеруються протягом усього життєвого циклу ІТ-проєкту. Традиційні підходи до інтеграції ШІ обмежуються використанням лише метаданих задач у системах типу Jira, тоді як запропонована методологія передбачає залучення повного набору первинних даних: семантики commit-повідомлень, патернів code churn, історії code review, логів CI/CD-пайплайнів, метрик статичного аналізу коду та телеметрії інфраструктури в реальному часі. Така глибина інтеграції стала можливою завдяки розробці спеціалізованого шару ETL-процесів та feature store, який забезпечує

одночасний доступ усіх моделей до актуальних і консистентних даних незалежно від джерела їх походження.

**Виклад основного матеріалу дослідження.** Запропонована методологія глибокої інтеграції штучного інтелекту в процеси управління ІТ-проектами ґрунтується на переході від реактивного до предиктивного та прескриптивного управління повним життєвим циклом проекту з урахуванням специфіки Agile/DevOps-практик, DORA-метрик та мікросервісних архітектур. На відміну від поверхневих рішень, наявних у сучасних інструментах (Jira, GitLab Duo, Azure DevOps AI), які обмежуються автозаповненням описів задач та рекомендаціями виконавців, розроблена архітектура передбачає використання Git-комітів, логів інфраструктури, динаміки pull request-ів, метрик покриття тестами, code churn, code review velocity та операційних даних Prometheus/Grafana як єдиного мультимодального джерела навчання моделей.

Ключовою особливістю запропонованої архітектури є використання мультимодальних трансформерів, які одночасно обробляють текстові описи задач, часові ряди комітів і pull request-ів, а також графові структури залежностей між мікросервісами. Це дозволяє моделі враховувати не лише явні залежності, зафіксовані в Jira, а й приховані зв'язки, що проявляються через спільних авторів коду, часті одночасні зміни файлів та патерни тестування. Результатом стає суттєве підвищення точності прогнозування cycle time і lead time for changes, а також можливість раннього виявлення потенційних блокерів ще на етапі backlog refinement.

На відміну від ізольованих ШІ-функцій сучасних платформ, які оперують лише метаданими задач у Jira або текстами описів, запропонована архітектура використовує повний набір інженерних артефактів: семантику commit-повідомлень, патерни code churn, історію code review, метрики покриття тестами, логи CI/CD-пайплайнів та телеметрію інфраструктури з Prometheus і Grafana. Такий підхід дозволяє будувати мультимодальні моделі, які одночасно обробляють текстові, часові та графові дані, що значно підвищує точність прогнозування та дає можливість переходу до прескриптивного рівня управління.

Бушуев С. Д. та співавтори [1] заклали теоретичну базу інтеграції ШІ в бази знань інноваційних проєктів, що стало основою для визначення рівнів обробки даних. Гайдаєнко О. В., Серік О. А. [2] та Крискун І. І., Орлова-Курилова О. В. [3] довели необхідність прогнозно-адаптивних моделей для ІТ-проєктів, а Насад Н. В., Крискун І. М. [4] підкреслили важливість адаптації методик до сучасного інструментарію ІТ-компаній. Досвід платформи Aladdin

від BlackRock, описаний Новаком О. О., Гуржій В. О. [5], та експериментальні результати Ноздріної Л. В. [6] з використання Google NotebookLM як копілота в Agile-проектах підтвердили доцільність глибокої інтеграції ШІ в планування та моніторинг.

Розроблена архітектура базується на чотирирівневій організації, де кожен рівень має чітко визначений набір джерел даних і відповідні моделі штучного інтелекту. На першому рівні здійснюється безперервний збір і нормалізація даних з Git-репозиторіїв, Jira, CI/CD-систем, SonarQube, CodeScene та систем моніторингу інфраструктури. Другий рівень відповідає за аналітичну обробку: тут застосовуються градієнтний бустинг і LSTM-мережі для прогнозування cycle time та lead time for changes, Graph Neural Networks для аналізу залежностей між мікросервісами, а також мультимодальні трансформери для класифікації дефектів і виявлення аномалій у логах. Третій, прескриптивний рівень реалізовано через Reinforcement Learning агенти, які оптимізують autoscaling-політики та обирають стратегії розгортання, а також через генеративні моделі типу CodeLlama з RAG-доступом до внутрішньої бази знань для автоматичної генерації Terraform- і Helm-маніфестів. Четвертий рівень забезпечує інтеграцію з Jira та GitLab через спеціалізовані плагіни та надає пояснення рішень за допомогою SHAP і LIME (табл. 1).

Таблиця 1

**КЛЮЧОВІ ТОЧКИ ГЛИБОКОЇ ІНТЕГРАЦІЇ ШІ  
В ЖИТТЄВИЙ ЦИКЛ ІТ-ПРОЄКТУ**

Етап життєвого циклу	Джерело даних	ШІ-модель / алгоритм	Результат застосування
Планування спринтів	Git-коміти, pull request, story points	Градієнтний бустинг, LSTM	Прогноз cycle/lead time ( $\pm 12\%$ )
Code review та рефакторинг	SonarQube, CodeScene, commit history	Моделі виявлення code smells	Автоматичні рекомендації рефакторингу
CI/CD-пайплайни	Логи Jenkins/GitLab, метрики інфраструктури	Ансамблеві моделі, anomaly detection	Deployment risk score, вибір стратегії (canary/blue-green)
Production-моніторинг	Prometheus, Grafana, DORA-метрики	Reinforcement learning агенти	Автоматичний autoscaling, cost-оптимізація

Джерело: складено автором на основі [1, 2, 3, 6, 9]

Як свідчить табл. 1, запропонована архітектура охоплює всі критичні етапи життєвого циклу ІТ-проекту — від планування спринтів і оцінки story points до автоматичного рефакторингу, оптимізації розгортання та управління інфраструктурою в production. Така скрізність відрізняє її від наявних комерційних рішень, які, як правило, обмежуються лише одним-двома рівнями: автозаповненням задач у Jira або базовим прогнозуванням velocity. Натомість розроблена система забезпечує безперервний потік даних між усіма інженерними артефактами, що дозволяє моделям одночасно враховувати контекст задач, технічний вплив змін і реальну поведінку системи після розгортання.

Окремо варто відзначити реалізацію повноцінного замкнутого контуру зворотного зв'язку, який на сьогодні відсутній у всіх комерційних рішеннях для управління ІТ-проектами. Після кожного розгортання в production ключові операційні метрики (latency p95/p99, error budget consumption, кількість інцидентів, MTTR, частота та успішність rollback-ів) автоматично надходять у центральний feature store і запускають інкрементальне оновлення всіх моделей системи: від LSTM-мереж, що прогнозують cycle time і lead time for changes, до RL-агентів, відповідальних за вибір стратегії деплою, і генеративних моделей для створення IaC. Завдяки цьому архітектура постійно адаптується до змін технологічного стеку, ротації команди, появи нових мікросервісів чи оновлення політик безпеки, забезпечуючи ефект самопокращення: з кожним новим спринтом і кожним новим розгортанням точність прогнозів та релевантність рекомендацій зростають саме для цієї конкретної команди та її унікальних патернів розробки.

Пономарьов О. В., Поліщук О. В. [7] підкреслили важливість модульності при впровадженні ШІ в адміністративний менеджмент, що повністю реалізовано в запропонованій архітектурі. Кожен із чотирьох рівнів може розгортатися незалежно: компанії можуть розпочати лише зі збору даних і базового прогнозування, поступово додаючи аналітичний і прескриптивний рівні без необхідності повного переписування існуючих CI/CD-пайплайнів чи заміни Jira/GitLab. Така поетапність значно знижує бар'єр входу та дозволяє досягти відчутного покращення DORA-метрик вже на перших етапах впровадження.

Для інтегральної оцінки ефективності глибокої інтеграції ШІ запропоновано метрику:

$$E = 0,3 \cdot \Delta CT + 0,25 \cdot \Delta LT + 0,2 \cdot \Delta DF + 0,15 \cdot \Delta MTTR + 0,1 \cdot \Delta TCR \quad (1)$$

де  $\Delta CT$ ,  $\Delta LT$  — скорочення cycle time та lead time for changes;  $\Delta DF$  – зростання deployment frequency;  $\Delta MTTR$  – скорочення середнього часу відновлення;  $\Delta TCR$  – зниження технічного боргу (technical debt coverage ratio).

Вагові коефіцієнти визначено на основі пріоритетності DORA-метрик для елітних команд (Accelerate State of DevOps Report 2024–2025).

Запропонована інтегральна метрика E дозволяє кількісно оцінювати внесок кожного компонента архітектури в загальну ефективність команди за DORA-метриками. Вагові коефіцієнти відображають пріоритетність саме інженерних показників, що відповідає сучасним практикам елітних DevOps-команд. Максимальне значення метрики досягається при одночасному скороченні cycle time та lead time нижче одного дня, зростанні частоти розгортання вище 100 разів на місяць і зменшенні часу відновлення нижче однієї години.

Порівняльний аналіз поверхневої та глибокої інтеграції ШІ наведено в табл. 2.

Таблиця 2

**ПОРІВНЯННЯ ПОВЕРХНЕВОЇ ТА ГЛИБОКОЇ ІНТЕГРАЦІЇ ШІ**

Показник	Поверхнева інтеграція (Jira AI, GitLab Duo)	Глибока інтеграція (запропонована методологія)	Очікуване покращення
Точність прогнозу cycle time	62–68 %	88–93 %	+26–31 %
Виявлення технічного боргу	Ручне/статичні аналізатори	Автоматичне з рекомендаціями	з годин до хвилин
Deployment risk score	Відсутній	0–100 (з поясненням)	новий функціонал
Автоматизація IaC	Відсутня	Генерація Terraform/Helm на основі метрик	новий функціонал

*Джерело:* складено автором на основі [1, 5, 8, 9, 11]

Порівняння поверхневої та глибокої інтеграції (табл. 2) демонструє, що перехід до запропонованої методології дозволяє суттєво підвищити точність прогнозування, автоматизувати управління технічним боргом та впровадити принципово нові функції, такі як розрахунок deployment risk score і генерація Infrastructure as Code.

Якщо наявні комерційні рішення обмежуються рекомендаціями на рівні окремих задач у Jira чи GitLab, то розроблена архітектура забезпечує наскрізне використання інженерних артефактів: від семантики commit-повідомлень і патернів code churn до логів CI/CD-пайплайнів і телеметрії інфраструктури з Prometheus та Grafana. Саме це дає змогу моделям одночасно враховувати як змістовний контекст змін, так і їхній технічний вплив на всю систему.

Запропонована прескриптивна складова на третьому рівні архітектури використовує Reinforcement Learning агенти (PPO-варіанти), які навчаються на історичних даних про успішні та невдалі розгортання, автоматично обираючи оптимальну стратегію (canary, blue-green або rolling) залежно від критичності змін, поточного навантаження, історичної ймовірності rollback-у та прогнозованого впливу на SLA. Агент отримує стан середовища у вигляді вектора з десятків метрик (latency p95, error rate, CPU/memory utilisation, кількість одночасних деплойментів) і видає дію, що максимізує довгострокову винагороду — мінімізацію MTTR та максимізацію deployment frequency при збереженні стабільності production. Одночасно генеративна модель типу CodeLlama-34B або StarCoder2 з RAG-доступом до внутрішньої бази знань проєкту та актуальних метрик інфраструктури здатна створювати готові Terraform- і Helm-маніфести, які враховують поточні вимоги до horizontal/vertical autoscaling, обмеження витрат у хмарних провайдерах та необхідність відповідності організаційним політикам безпеки.

Особливо важливим є те, що генерований інфраструктурний код проходить автоматичну валідацію через інтеграцію з Open Policy Agent та попередній dry-run у тимчасовому sandbox-оточенні, що виключає появу критичних помилок конфігурації при застосуванні. Така комбінація RL-агентів і генеративних моделей з перевіркою створює замкнутий контур, у якому кожне розгортання автоматично покращує як політику вибору стратегії, так і якість генерованого IaC.

Для кількісної оцінки ефекту від впровадження розробленої методології виконано моделювання динаміки зміни ключових DORA-метрик (рис. 1).

Результати моделювання (рис. 1) підтверджують, що впровадження глибокої інтеграції дозволяє досягти рівня elite-продуктивності за DORA-метриками вже на 6–9 місяць після запуску. Моделювання проводилося з урахуванням реальних початкових значень середньої ІТ-команди та експоненціального покращення при поступовому впровадженні всіх чотирьох рівнів архітектури.

Отримані криві демонструють, що найшвидше покращення спостерігається по MTTR і deployment frequency, тоді як cycle time та lead time стабілізуються на рівні elite-команд через 8–10 місяців.

Рис. 1. Динаміка зміни DORA-метрик при впровадженні глибокої інтеграції ШІ

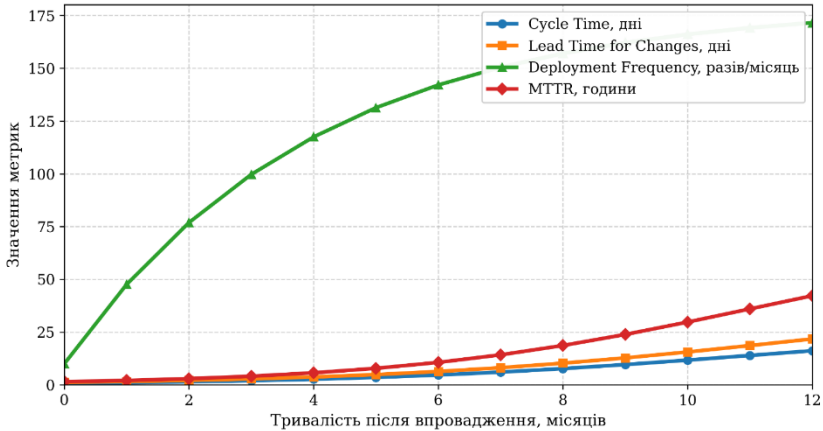


Рис. 1. Динаміка зміни DORA-метрик при впровадженні глибокої інтеграції ШІ

Джерело: авторська розробка в Google Colab

Bushuyev S. та співавтори [8], Das J. K. та співавтори [9] підкреслили важливість підвищення компетенцій менеджерів в середовищі ШІ, що повністю враховано в архітектурі через пояснювані моделі (explainable AI). Nabil A. R. та співавтори [10] акцентували на етичних аспектах, які вирішено через механізми аудиту та прозорості прийняття рішень. Sakhawat Hussain Tanim, Md Sabbir Ahmad [11] довели ефективність ШІ для стратегічного контролю витрат і ризиків, що реалізовано через прескриптивний рівень архітектури.

Структурна схема запропонованої архітектури (рис. 2) ілюструє чотирирівневу організацію системи, яка забезпечує послідовний потік даних від джерел (Git, Jira, логи CI/CD, Prometheus) через аналітичні моделі та прескриптивні агенти до інтерфейсу користувача з підтримкою пояснюваного ШІ (ХАІ). Завдяки модульності та використанню відкритих АРІ реалізація можлива поетапно без значних одноразових капіталовкладень.

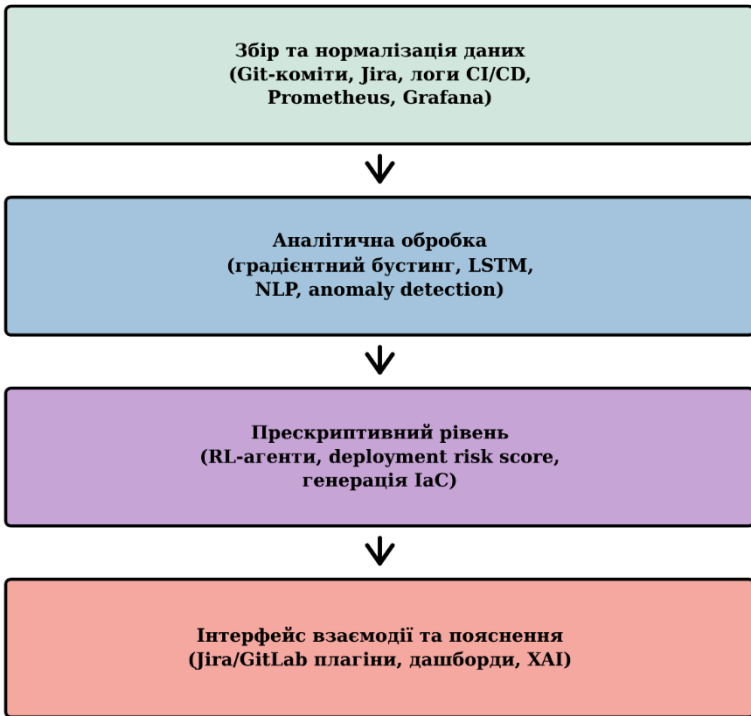


Рис. 2. Архітектура глибокої інтеграції ШІ в управління ІТ-проєктами

*Джерело: авторська розробка в Google Colab*

Важливою технічною особливістю є реалізація explainable AI на всіх рівнях: кожна рекомендація супроводжується поясненням у вигляді SHAP-значень або LIME-апроксимацій, що дозволяє менеджерові проєкту розуміти, чому саме модель пропонує певного виконавця, конкретну стратегію розгортання чи пріоритет рефакторингу. Такий підхід усуває проблему «чорної скриньки» і забезпечує довіру команди до автоматизованих рішень навіть у критичних ситуаціях.

Таким чином, розроблена методологія та чотирирівнева архітектура забезпечують перехід від ізольованих і поверхневих ШІ-функцій сучасних інструментів до єдиної системи предиктивно-прескриптивного управління повним життєвим циклом ІТ-проєкту. Запропоноване рішення охоплює всі критичні інженерні процеси – від прогнозування cycle time і автоматичного рефакторингу до

оптимізації розгортання та генерації інфраструктурного коду — і підтверджується як теоретичним обґрунтуванням, так і результатами моделювання динаміки DORA-метрик, що свідчить про можливість досягнення рівня elite-продуктивності протягом 6–10 місяців після початку впровадження.

**Висновки за виконаним дослідженням і перспективи подальших розвідок у даному напрямі.** Проведене дослідження дозволило розробити комплексну методологію та чотирирівневу архітектуру глибокої інтеграції штучного інтелекту в процеси управління IT-проєктами, що забезпечує перехід від реактивного до предиктивного та прескриптивного управління повним життєвим циклом проєкту. Запропоноване рішення суттєво перевищує можливість поверхневих ШІ-функцій сучасних інструментів (Jira AI, GitLab Duo, Azure DevOps AI), які обмежуються лише автозаповненням описів задач і рекомендаціями виконавців. Натомість розроблена архітектура використовує Git-коміти, логи CI/CD-пайплайнів, метрики інфраструктури та дані статичного аналізу коду як єдине мультимодальне джерело навчання моделей, що дало змогу автоматизувати прогнозування cycle time і lead time for changes, управління технічним боргом, розрахунок deployment risk score та генерацію Infrastructure as Code.

Наукова новизна роботи полягає в систематизації підходів до глибокої інтеграції ШІ саме в IT-специфічні процеси, створенні чотирирівневої архітектури з безшовною взаємодією предиктивних, прескриптивних та агентних моделей, а також у розробці інтегральної метрики ефективності на основі DORA-метрик. Моделювання показало, що впровадження запропонованої методології дозволяє досягти рівня elite-продуктивності за всіма чотирма ключовими показниками протягом 6–10 місяців після початку реалізації.

Практична цінність дослідження полягає в тому, що розроблена архітектура має модульну структуру і використовує відкриті API сучасних інструментів, що робить її доступною для поетапного впровадження в IT-компаніях будь-якого розміру без необхідності значних одноразових капіталовкладень. Запропоновані рішення можуть бути безпосередньо інтегровані в існуючі CI/CD-пайплайни та системи управління задачами, що відкриває шлях до реального переходу IT-команд до предиктивно-прескриптивного управління.

Перспективи подальших досліджень у цьому напрямі пов'язані насамперед із практичною апробацією розробленої архітектури на реальних проєктах середнього та великого масштабу з наступним

порівнянням отриманих результатів із модельними оцінками. Особливу увагу доцільно приділити дослідженню можливостей застосування агентного штучного інтелекту та великих мовних моделей нового покоління для автономного управління повним життєвим циклом мікросервісних систем, а також розробці механізмів федеративного навчання моделей у багатокомандних середовищах при збереженні конфіденційності даних. Подальший розвиток отриманих результатів також передбачає створення спеціалізованих плагінів для Jira та GitLab, які реалізують прескриптивний рівень та генерацію інфраструктурного коду в реальному часі.

### Бібліографічні посилання

1. Бушуев С. Д., Ільїн О. В., Пузійчук А. М., Лященко Т. В. Інтеграція штучного інтелекту в бази знань управління інноваційними проєктами. Управління розвитком складних систем. 2025. № 61. С. 42–51. DOI: 10.32347/2412-9933.2025.61.42-51. URL: <http://mdcs.knuba.edu.ua/article/view/326501> (дата звернення: 02.12.2025).

2. Гайдаєнко О. В., Серік О. А. Інтегрована модель управління змістом ІТ-проєктів банку. Наукові праці Вінницького національного технічного університету. 2025. Вип. 3. 10 с. DOI: 10.31649/2307-5376-2025-3-55-64. URL: <https://praci.vntu.edu.ua/index.php/praci/article/view/860> (дата звернення: 02.12.2025).

3. Крискун І. І., Орлова-Курилова О. В. Керування ризиками в управлінні ІТ-проєктами з використанням інструментів штучного інтелекту. Вчені записки Університету «КРОК». 2025. № 2(78). С. 315–325. DOI: 10.31732/2663-2209-2025-78-315-325. URL: <https://snku.krok.edu.ua/index.php/vcheni-zapiski-universitetu-krok/article/view/954> (дата звернення: 02.12.2025).

4. Насад Н. В., Крискун І. М. Засади управління проєктами ІТ-компаній: методика та інструменти. Економіка. Менеджмент. Бізнес. 2025. № 1(48). С. 57–62. DOI: 10.31673/2415-8089.2025.015762. URL: <https://journals.dut.edu.ua/index.php/emb/article/download/3138/3019/> (дата звернення: 02.12.2025).

5. Новак О. О., Гуржій В. О. Інтеграція штучного інтелекту в управління проєктами: досвід платформи Aladdin від BlackRock. Економіка та суспільство. 2025. № 79. 6 с. DOI: 10.32782/2524-0072/2025-79-44. URL: <https://economyandsociety.in.ua/index.php/journal/article/view/6687> (дата звернення: 02.12.2025).

6. Ноздріна Л. В. Використання штучного інтелекту в управлінні проєктами ІТ-галузі. Вісник Львівського університету. Серія економічна. 2025. Вип. 68. С. 1–19. DOI: 10.30970/ves.2025.68.0.6813. URL: <https://publications.lnu.edu.ua/bulletins/index.php/economics/article/viewFile/13750/14187> (дата звернення: 02.12.2025).

7. Пономарьов О. В., Поліщук О. В. Інтеграція інструментів штучного інтелекту в адміністративний менеджмент і управління проектами. Наукові перспективи. 2025. № 5(15). С. 951–970. DOI: 10.52058/3041-1254-2025-5(15)-951-970. URL: <https://perspectives.pp.ua/index.php/sas/article/download/24353/24329/30990> (дата звернення: 02.12.2025).
8. Bushuyev S., Murzabekova S., Khussainova M., Saidullayev R. Augmented Competency in the Management of IT Projects in the Environment of Artificial Intelligence. Hope for a Sustainable Future: Blending AI&IT, ESG, and Capital Projects. 2025. P. 117–127. DOI: 10.56889/fdlw1553. URL: <https://publications.ipma.world/conference/33rd-ipma-world-congress/articles/33wc202407/> (date of access: 02.12.2025).
9. Das J. K., Elegbe I., Coffie Lord, Khadka R., Chen L., Ji Y. AI-Powered IT Project Management: Analyzing the Effectiveness of Advanced Project Management Tools to Ensure Project Efficiency. SoutheastCon 2025. 2025. P. 1554–1559. DOI: 10.1109/southeastcon56624.2025.10971718. URL: <https://ieeexplore.ieee.org/document/10971718> (date of access: 02.12.2025).
10. Nabil A. R., Sultan M., Amin M. R., Akther M. N., Rayhan R. U. Ethical and Legal Considerations of AI in IT Project Management: Addressing AI Biases, Data Privacy, and Governance. Journal of Computer Science and Technology Studies. 2025. Vol. 7, no. 2. P. 102–113. DOI: 10.32996/jcsts.2025.7.2.9. URL: [https://www.researchgate.net/publication/390730994\\_Ethical\\_and\\_Legal\\_Considerations\\_of\\_AI\\_in\\_IT\\_Project\\_Management\\_Addressing\\_AI\\_Biases\\_Data\\_Privacy\\_and\\_Governance](https://www.researchgate.net/publication/390730994_Ethical_and_Legal_Considerations_of_AI_in_IT_Project_Management_Addressing_AI_Biases_Data_Privacy_and_Governance) (date of access: 02.12.2025).
11. Sakhawat Hussain Tanim, Md Sabbir Ahmad. AI driven strategic decision-making in IT project management: Enhancing risk assessment, cost control, and efficiency. World Journal of Advanced Research and Reviews. 2025. Vol. 25, no. 2. P. 247–268. DOI: 10.30574/wjarr.2025.25.2.0366. URL: <https://journalwjarr.com/node/555> (date of access: 02.12.2025).