

DOI: <https://doi.org/10.15276/ict.03.2026.26>

УДК 005.8:004.89

Модель колективного прийняття рішень та метод контекстно-залежної координації агентів штучного інтелекту в управлінні ІТ проєктами

Лінник Олексій Олександрович¹⁾

ORCID: <https://orcid.org/0009-0007-3060-1412>; alexeylinnik@stud.op.edu.ua

Шутов Сергій Ігорович¹⁾

ORCID: <https://orcid.org/0009-0006-4190-3988>; 3234816@as.op.edu.ua

¹⁾ Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

В статті досліджено процес колективного прийняття адаптивних управлінських рішень спеціалізованими агентами штучного інтелекту в управлінні ІТ-проєктом. Мета дослідження – підвищення обґрунтованості, узгодженості та адаптивності рішень шляхом удосконалення моделі колективного прийняття рішень і методу координації спеціалізованих ІТ-агентів. Модель об'єднує в єдиному адаптивному циклі нормований стан проєкту, спеціалізовані ролі, допустимі альтернативи, локальні оцінки, конфлікти та фактичні результати виконання. Результат рішення зіставляється з прогнозами агентів і використовується для оновлення стану, результативності, довіри та ваг впливу. Конфлікт визначається як контекстно надлишкова оціночна, рангова, прогнозна, інформаційна або дієва несумісність. Коректність циклу забезпечують інваріанти контекстної цілісності, допустимості, простежуваності та стабільності. Удосконалений метод координації виявляє суперечності та динамічно змінює ролі й повноваження з урахуванням стану проєкту, компетентності агентів, щільності конфліктного графа, невизначеності й витрат реконфігурації. Алгоритм охоплює формування версійного знімка, активацію ролей, побудову альтернатив, локальне оцінювання, вибір конфігурації, глобальний розподіл робіт, визначення режиму автономності та ретроспективне оновлення. Гістерезис і заморожений горизонт обмежують необґрунтовані перепризначення, а конфігурація та план вводяться атомарно. Експеримент проведено на сценарії з вісімнадцятьма розробниками, трьома підкомандами, ста двадцятьма користувацькими історіями та шістьма двотижневими спринтами. Упевненість у рішеннях зросла з нуля цілих сімдесяти однієї соті до нуля цілих вісімдесяти дев'яти сотих, частка автономно диспетчеризованих рішень – з нуля цілих п'ятдесяти восьми сотих до нуля цілих вісімдесяти однієї соті, а результативність пом'якшення ризиків – з нуля цілих сорока восьми сотих до нуля цілих сімдесяти двох сотих. Частка розв'язаних конфліктів становила нуль цілих дев'яносто одну соту, індекс стабільності – нуль цілих вісімдесяти сім сотих, частка хибних ескаляцій – нуль цілих тринадцяти сотих. Результати підтвердили ефективність запропонованого підходу.

Ключові слова: управління ІТ-проєктами, багатоагентні системи, колективне прийняття рішень, адаптивний зворотний зв'язок, динамічний розподіл ролей, координація агентів, конфлікти рішень, керована автономність.

Для цитування: Лінник О. О., Шутов С.І. “Модель колективного прийняття рішень та метод контекстно-залежної координації агентів штучного інтелекту в управлінні ІТ проєктами”. *Інформатика. Культура. Техніка.* 2026; Том 3 № 1 (3): 308–320. DOI: <https://doi.org/10.15276/ict.03.2026.26>

Актуальність. Управління сучасним ІТ-проєктом відбувається за постійної зміни вимог, залежностей, доступності фахівців і зовнішніх обмежень. Agile-підходи скоротили цикл зворотного зв'язку та підвищили результативність проєктів [1], [2], [3], [4], однак зі збільшенням масштабу команди та кількості взаємозалежних робіт зростає координаційна складність [5], [6]. Проєктний менеджер повинен одночасно відстежувати строки, бюджет, ресурси, якість, вимоги й ризики, а нові події можуть зробити попереднє рішення неприйнятним ще до завершення спринту. Тому періодичної ретроспективної адаптації недостатньо для ситуацій, у яких рішення має бути сформовано та введено протягом короткого часового вікна.

Багатоагентні системи дають змогу розподілити спостереження та аналітичні функції між спеціалізованими програмними агентами [7], [8], [9], [10]. Агент строків може контролювати календарні відхилення, агент ресурсів – доступність і навантаження, агент ризиків – сценарії загроз, а координатор – формувати спільне рішення. Проте множинність агентів сама по собі не гарантує узгодженості. Якщо вони використовують різні версії даних, мають непорівнювані критерії або формують локально оптимальні, але ресурсно несумісні рекомендації, колектив збільшує інформаційний шум і затримку замість підвищення якості рішення.

© Лінник О., Шутов С., 2026

Стаття опублікована у відкритому доступі відповідно до умов ліцензії CC BY (<https://creativecommons.org/licenses/by/4.0/deed.uk>)

Наявні засоби штучного інтелекту в управлінні проєктами переважно вирішують ізольовані задачі прогнозування тривалості, планування або підтримки Agile-процесів [11], [12], [13]. У багатоагентному плануванні опрацьовано розподіл робіт і ресурсів [13], однак менш дослідженим залишається цілісний перехід від спільного стану проєкту до формування альтернатив, діагностики конфлікту, вибору структури взаємодії, розподілу виконавчих пакетів та визначення допустимого рівня автономності. Особливо важливим є розмежування двох задач: модель повинна пояснювати, що саме становить ситуацію колективного рішення, а метод координації – як змінити колектив і організувати реалізацію рішення в цій ситуації.

Отже, актуальним науково-прикладним завданням є розроблення формального ядра колективного прийняття адаптивних управлінських рішень спеціалізованими ШІ-агентами та методу їх контекстно-залежної координації. Розв'язання цього завдання має забезпечити порівнюваність локальних оцінок, явне опрацювання розбіжностей, структурну допустимість колективу, глобальну узгодженість розподілу робіт і безпечне поєднання автономного виконання з людським контролем.

Аналіз останніх досліджень і публікацій. У дослідженнях з управління ІТ-проєктами значну увагу приділено скороченню циклів планування, оперативному одержанню зворотного зв'язку та адаптації проєктних рішень до змін середовища [1], [2], [3], [4]. Водночас масштабування гнучких підходів супроводжується збільшенням кількості залежностей між роботами, учасниками та ресурсами, що істотно підвищує координаційну складність [5], [6]. Засоби штучного інтелекту застосовують для прогнозування тривалості робіт, оцінювання ризиків, планування ресурсів і підтримки Agile-процесів [11], [12]. Інтегрований підхід до адаптивного управління ІТ-проєктами за допомогою спеціалізованих ШІ-агентів запропоновано у праці [14]. Однак перехід від сукупності окремих прогнозів і рекомендацій до формування узгодженого колективного рішення потребує формального представлення стану проєкту, доступних альтернатив, позицій агентів і результатів реалізації попередніх рішень.

Теоретичну основу розподіленого опрацювання управлінської інформації становлять дослідження інтелектуальних і багатоагентних систем. Інтелектуальний агент визначається як автономна сутність, здатна сприймати середовище, формувати висновки та виконувати дії відповідно до поставлених цілей [7]. У працях [8], [9] систематизовано механізми кооперації, переговорів, розподіленого навчання та спільного розв'язання задач. Координація багатоагентної системи розглядається як організація взаємозалежної діяльності агентів шляхом розподілу функцій, обміну інформацією й узгодження поведінки [10]. Водночас загальні багатоагентні архітектури не визначають структуру ситуації прийняття рішення саме для ІТ-проєкту, у якій значення кожного сигналу залежить від фази життєвого циклу, базового плану, критичності робіт, доступності ресурсів і чинних обмежень.

Важливою складовою колективного прийняття рішень є агрегування локальних оцінок. Для їх об'єднання використовують зважені та впорядковані оператори, які дають змогу враховувати різну значущість окремих оцінок [15]. Моделі обчислювальної довіри та репутації забезпечують коригування впливу агентів на основі результатів їхньої попередньої діяльності [16]. Проте історична результативність не може компенсувати відсутність необхідної компетентності, повноважень або актуальних даних для оцінювання конкретної альтернативи. Тому визначенню ваг агентів повинна передувати перевірка відповідності їхніх компетентностей активним ролям, достатності інформаційних підстав і допустимості сформованого складу колективу.

Підхід до координованої підтримки прийняття рішень в ІТ-проєктах із використанням багатоагентної системи розглянуто у праці [17]. Його застосування дає змогу узгоджувати результати спеціалізованого аналізу та зменшувати залежність рішення від позиції окремого агента. Подальший розвиток моделей довіри та репутації для відкритих багатоагентних систем представлено у праці [18]. Разом із тим відмінність між рекомендаціями агентів не завжди свідчить про конфлікт, оскільки вона може бути природним наслідком різних професійних ролей і локальних цілей. Для коректного виявлення суперечностей необхідно

зіставляти не лише числові оцінки, а й ранги альтернатив, прогнози наслідків, запропоновані дії та використані інформаційні підстави.

Окремий напрям досліджень охоплює багатоагентне планування та розподіл завдань і ресурсів [13]. Основи децентралізованого оголошення завдань, подання пропозицій і вибору виконавців сформовано у праці [19]. Сучасні методи проєктного планування та контролю узагальнено у дослідженні [20]. Децентралізована взаємодія дає змогу враховувати локальні можливості виконавців, однак сукупність найкращих локальних призначень не обов'язково утворює допустимий план ІТ-проєкту. Такий план може порушувати календарні залежності, перевищувати доступну місткість ресурсів, перевантажувати критичного фахівця або вимагати невинновданого перегляду вже прийнятих призначень. Отже, локальне оцінювання пропозицій необхідно доповнювати глобальною перевіркою плану, контролем ресурсних і часових обмежень, а також урахуванням витрат на реконфігурацію рольової структури.

Аналіз розглянутих праць також виявив недостатнє розмежування ризику управлінської дії та впевненості в інформаційних підставах її вибору. Низьке прогнозоване значення втрат, отримане за неповними або застарілими даними, не може бути достатньою підставою для автономного виконання рішення. Так само фактичну розбіжність між позиціями агентів доцільно порівнювати з очікуваним для відповідної пари ролей рівнем, щоб не ототожнювати продуктивну міжрольову різницю з надлишковою несумісністю.

Таким чином, у наявних дослідженнях сформовано теоретичні та прикладні засади інтелектуальної підтримки управління проєктами, багатоагентної координації, агрегування оцінок, обчислення довіри та розподілу робіт. Водночас недостатньо формалізованими залишаються два взаємопов'язані аспекти. Перший полягає в об'єднанні стану ІТ-проєкту, спеціалізованих ролей, альтернативних дій, локальних оцінок, конфліктів і фактичних результатів у межах єдиного адаптивного циклу прийняття рішення. Другий стосується координації колективу через динамічний розподіл ролей і повноважень з урахуванням компетентності агентів, щільності конфліктів та витрат на зміну чинної рольової структури. Формалізація зазначених аспектів визначає мету і зміст цього дослідження.

Метою дослідження є удосконалення моделі колективного прийняття рішень і методу координації спеціалізованих ІІІ-агентів для адаптивного управління ІТ-проєктом. Для досягнення мети необхідно: пов'язати стан, ролі, альтернативи, локальні оцінки, конфлікти й результати виконання в замкненому циклі; формалізувати динамічний розподіл ролей і повноважень; урахувати щільність конфліктів і витрати рольової реконфігурації; забезпечити узгоджений розподіл завдань і ресурсів; експериментально оцінити результативність і стійкість запропонованих рішень.

Наукова новизна першого результату полягає в удосконаленні моделі колективного прийняття рішень агентами штучного інтелекту в управлінні ІТ-проєктом, яка, на відміну від наявних, пов'язує поточний стан проєкту, спеціалізовані ролі агентів, множину альтернативних дій, локальні оцінки агентів, конфлікти між ними та результати виконання прийнятих рішень у єдиному адаптивному циклі.

Наукова новизна другого результату полягає в удосконаленні методу координації спеціалізованих агентів штучного інтелекту, який передбачає виявлення суперечностей між запропонованими діями та динамічний розподіл ролей і повноважень агентів з урахуванням стану ІТ-проєкту, компетентності агентів, щільності конфліктів і витрат на зміну ролей.

Модель колективного прийняття адаптивного управлінського рішення. Одиницею моделювання є ситуація прийняття рішення в момент часу t . Усі активні агенти отримують один версійно зафіксований знімок, а не самостійно вибирають актуальні джерела. Щоб модель описувала не окремих акт вибору, а замкнений адаптивний цикл, до ситуації включено локальні оцінки альтернатив та історію фактичних результатів виконання:

$$z(t) = \langle x(t), e(t), B(t), I(t), R(t), A(t), Q(t), Y^{hist}(t) \rangle, \quad (1)$$

де $z(t)$ – ситуація прийняття рішення в момент t ; $x(t)$ – нормований стан ІТ-проєкту; $e(t)$ – контекстні події; $B(t)$ – жорсткі й м'які обмеження; $I(t)$ – активний склад ІІІ-агентів; $R(t)$ –

активні ролі; $A(t)$ – допустимі альтернативи; $Q(t)$ – множина локальних оцінок $q_i(a, t)$ агентів i для альтернатив a ; $Y^{hist}(t)$ – історія фактичних результатів раніше виконаних рішень. Вектор $x(t)$ охоплює строки, вартість, виконання робіт, ресурси, якості, вимоги та ризики. Відсутнє значення не прирівнюється до нуля: для нього фіксуються доступність, давність і достовірність.

Ролі не закріплюються за агентами назавжди. Для кожної ситуації активуються функції планування строків, бюджету, робіт, ресурсів, якості, вимог, ризиків і координації. Придатність агента до ролі повинна одночасно враховувати компетентність, контекстну релевантність, історичну результативність, довіру, доступність та навантаження:

$$suit_{ir}(t) = cap_i(t) [fit_{ir}(t) rel_{ir}(t) perf_{ir}(t) trust_i(t)]^{1/4}, \quad (2)$$

де: $suit_{ir}(t)$ – придатність агента i до ролі r у момент t ; $cap_i(t)$ – доступна потужність агента після врахування навантаження; $fit_{ir}(t)$ – відповідність його компетентностей вимогам ролі; $rel_{ir}(t)$ – контекстна релевантність; $perf_{ir}(t)$ – результативність у подібних ситуаціях; $trust_i(t)$ – рівень довіри. Середній бал доповнюється контролем критичних компетентностей. Якщо обов'язкова роль не покрита або не має резервного агента, система не переходить до автономного рішення, а залучає експерта чи безпечну конфігурацію.

Множина альтернатив формується контекстно кількома джерелами: локальними генераторами спеціалізованих агентів, нормативно обов'язковими діями, історичними рішеннями та базовою дією без втручання. Пропозиції нормуються до спільної схеми, змістово дедуплікуються та перевіряються за жорсткими обмеженнями до колективного оцінювання:

$$A(t) = Feas_{B^H(t)} \left(\text{Uniq} \left(\text{Norm} \left(\{a_0\} \cup A^{reg} \cup A^{hist} \cup \bigcup_{i \in I(t)} Gen_i(z^0(t)) \right) \right) \right), \quad (3)$$

де $A(t)$ – підсумкова множина допустимих альтернатив; a_0 – базова дія без втручання; A^{reg} – нормативно обов'язкові дії; A^{hist} – релевантні історичні рішення; Gen_i – генератор пропозицій агента i ; $z^0(t)$ – зафіксований знімок ситуації; $Norm$, $Uniq$ і $Feas$ – оператори нормалізації, дедуплікації та перевірки допустимості; $B^H(t)$ – жорсткі обмеження. Такий порядок усуває витрати на оцінювання недопустимих дій і зберігає походження еквівалентних пропозицій. Відсутня локальна оцінка не дорівнює нулю: агент виключається лише з оцінювання відповідної альтернативи.

Після формування $A(t)$ кожний агент оцінює лише ті альтернативи, щодо яких має компетентність і достатню доказову базу. Локальна оцінка $q_i(a, t)$ містить очікуваний вплив на цілі проєкту, прогноз наслідків, рекомендовану дію, упевненість і посилення на використанні дані.

Конфлікт визначається як суттєва контекстно неочікувана несумісність позицій; для пари агентів аналізуються лише спільно оцінені альтернативи:

$$d_{ij}(t) = \frac{\alpha_q d_{ij}^q + \alpha_r d_{ij}^r + \alpha_y d_{ij}^y + \alpha_a d_{ij}^a + \alpha_d d_{ij}^{data}}{\alpha_q + \alpha_r + \alpha_y + \alpha_a + \alpha_d}, \quad (4)$$

де: $d_{ij}(t)$ – узагальнена розбіжність між агентами i та j ; d^q , d^r , d^y , d^a і d^{data} – відмінності відповідно локальних оцінок, рангів, прогнозів наслідків, рекомендованих дій та інформаційних підстав; α_q , α_r , α_y , α_a , α_d – ваги компонентів. Від фактичної розбіжності віднімається очікуваний базовий рівень для відповідної пари ролей. Це зменшує хибну ескалацію продуктивних суперечностей. Матриця розбіжностей формує конфліктний граф, ребра якого з'єднують лише агентів із суттєвою надлишковою несумісністю.

Ризик оцінюється не для абстрактної альтернативи, а для пари «дія – план реалізації» у фактично введеній конфігурації. Профіль включає проєктний, виконавчий, координаційний, інформаційний і модельний компоненти.

Інтегральний ризик визначається консервативно як максимум зваженого середнього та критичного виміру:

$$(a, t) = \max \left\{ \sum_{v \in V_p} \beta_v \rho_v(a, t), \max_{v \in V_{crit}} \rho_v(a, t) \right\}, \quad (5)$$

де: $\rho(a, t)$ – інтегральний ризик альтернативи a у момент t ; $\rho_v(a, t)$ – ризик за компонентом v ; β_v – вага компонента; V_p – множина всіх компонентів ризику; V_{crit} – множина критичних компонентів. Перший доданок у операторі максимуму відображає середній зважений ризик, другий – найгірший критичний вимір.

Упевненість характеризує надійність підстав для оцінки та вибору, тому не є доповненням ризику до одиниці:

$$h(a, t) = gate(a, t) \sum_{k \in K_h} \gamma_k h_k(a, t), \quad \sum_{k \in K_h} \gamma_k = 1, \quad (6)$$

де $h(a, t)$ – інтегральна впевненість щодо альтернативи a ; $gate(a, t)$ – критичний множник допустимості; $h_k(a, t)$ – часткова впевненість за складовою k ; γ_k – вага складової; K_h – множина складових упевненості. Множник $gate$ дорівнює нулю за критичного дефіциту: непокритої ролі, слабких даних, недопустимого плану, невведеної конфігурації або відсутності необхідного людського погодження. Для режиму виконання використовуються нижня межа впевненості та верхня межа ризику.

Метод контекстно-залежної координації спеціалізованих ШІ-агентів. Метод перетворює результати моделей стану, ролей, альтернатив і конфліктів на цілісну конфігурацію взаємодії. Кандидатна конфігурація включає активний склад, матрицю рольових призначень, граф каналів, механізм узгодження, ваги впливу та максимально допустиму автономність:

$$cf g_h(t) = \langle I_h(t), Z_h(t), E_h(t), mech_h(t), W_h(t), u_h(t) \rangle, \quad (7)$$

де $cf g_h(t)$ – кандидатна конфігурація h ; $I_h(t)$ – її активний склад агентів; $Z_h(t)$ – матриця рольових призначень і повноважень; $E_h(t)$ – граф комунікацій; $mech_h(t)$ – механізм узгодження; $W_h(t)$ – ваги впливу агентів; $u_h(t)$ – максимально допустимий рівень автономності. Потреба у реконфігурації виникає за значної зміни ситуації, суттєвого конфлікту, втрати рольового покриття або погіршення готовності даних. Перед ранжуванням кандидат проходить жорстку перевірку допустимості.

Для кожного кандидата матриця $Z_h(t)$ задає не лише назву ролі, а й межі читання даних, формування пропозицій, голосування, диспетчеризації та ескалації. Після виявлення суперечності метод може передати координаційну роль незалежному агенту, додати резервного експерта, обмежити повноваження конфліктного учасника або розділити функції оцінювання і виконання.

Допустимі конфігурації порівнюються за критерієм, що прямо враховує щільність конфліктів і витрати рольової реконфігурації:

$$J_h^c(t) = \sum_v \lambda_v B_{hv}(t) - \lambda_D \text{dens}(G_h^{conf}) - \lambda_T T_h^c - \lambda_C C_h^c - \lambda_R C_h^{role} - \lambda_U U_h, \quad (8)$$

де $J_h^c(t)$ – робастна оцінка конфігурації h ; $B_{hv}(t)$ – її корисність за перевагою v ; $\text{dens}(G_h^{conf})$ – щільність конфліктного графа як частка фактичних суттєвих ребер серед усіх можливих пар активних агентів; T_h^c і C_h^c – очікувані час і комунікаційна вартість координації; C_h^{role} –

зважені витрати на зміну ролевих призначень і повноважень відносно чинної матриці $Z(t-1)$; U_h – невизначеність; $\lambda_v, \lambda_D, \lambda_T, \lambda_C, \lambda_R$ і λ_U – вагові коефіцієнти.

Математична перевага кандидата ще не означає негайного перемикавання: перевіряються сигнал потреби, відрив від другого кандидата та чистий виграш після витрат переходу:

$$\widetilde{c}fg(t) = \begin{cases} cf g_{safe}(t), & CFG_F(t) = \emptyset; \\ cf g_{best}(t), & q_{sw}(t) = 1; \\ cf g(t-1), & q_{sw}(t) = 0. \end{cases} \quad (9)$$

де $\widetilde{c}fg(t)$ – цільова конфігурація; $cf g_{safe}(t)$ – безпечна конфігурація; $CFG_F(t)$ – множина допустимих кандидатів; $cf g_{best}(t)$ – кандидат із найкращою робастною оцінкою; $cf g(t-1)$ – чинна конфігурація; $q_{sw}(t)$ – бінарний сигнал перемикавання. Цільова конфігурація вводиться атомарно після підготовки агентів, каналів, повноважень, версій даних і людського погодження. За невиконання хоча б однієї перевірки продовжує діяти попередня допустима конфігурація або виконується відкат.

Після введення конфігурації метод розподіляє атомарні робочі пакети. Кожний пакет має вимоги до ролі й компетентностей, тривалість, строк, пріоритет, ресурсну потребу, залежності та критерій завершення. Розширена заявка агента містить прогноз якості, часу, вартості, ризику невиконання, навантаження, доказовість і припущення.

Локальний робастний бал заявки визначається так:

$$B_{ui}^{rob}(t) = B_{ui}(t) - \kappa_U U_{ui}^{bid}(t), \quad (10)$$

де $B_{ui}^{rob}(t)$ – робастний бал заявки агента i на виконання пакета u ; $B_{ui}(t)$ – базовий бал заявки; $U_{ui}^{bid}(t)$ – невизначеність її прогнозів і припущень; κ_U – коефіцієнт штрафу за невизначеність. Вибір виконавців виконується глобально для всіх задач і спільних ресурсів.

Критерій винагороджує корисність заявок, баланс навантаження і ресурсний резерв та штрафуює конфлікти, запізнення, невизначеність і зміну вже прийнятих призначень:

$$alloc^*(t) = \arg \max_{h \in ALLOC_F(t)} \left[\sum_{u,i} \pi_{ui}^h B_{ui}^{rob} + \lambda_b bal_h + \lambda_r res_h - P_h^{coord} - P_h^{late} - P_h^{change} \right], \quad (11)$$

де: $alloc^*(t)$ – оптимальний план розподілу; $ALLOC_F(t)$ – множина допустимих планів; π_{ui}^h – ознака призначення пакета u агенту i у плані h ; bal_h і res_h – баланс навантаження та ресурсний резерв; P_h^{coord} , P_h^{late} і P_h^{change} – штрафи за координаційні конфлікти, запізнення та перепризначення; λ_b і λ_r – ваги балансу й резерву. До допустимих належать лише плани, в яких кожне завдання має виконавця, призначення відповідає ролям і повноваженням, а ресурсні та часові обмеження виконано.

Остаточний режим виконання поєднує результат координації з оцінками ризику й упевненості. Автономна диспетчеризація дозволена лише для структурно допустимої конфігурації та плану за достатньої нижньої межі впевненості й прийнятної верхньої межі ризику:

$$m(t) = \begin{cases} \text{auto}, & h^- \geq \tau_H, \rho^+ \leq \tau_{\rho L}; \\ \text{control}, & h^- \geq \tau_L, \rho^+ < \tau_{\rho H}; \\ \text{manager}, & \text{otherwise}, \end{cases} \quad (12)$$

де $m(t)$ – режим виконання; *auto* – автономне виконання оборотних дій; *control* – виконання з повідомленням або контрольним вікном; *manager* – режим явного погодження менеджером; h^- – консервативна нижня межа впевненості; ρ^+ – консервативна верхня межа ризику; τ_H і τ_L – високий та мінімальний пороги впевненості; $\tau_{\rho L}$ і $\tau_{\rho H}$ – низький та граничний пороги ризику. Мовчання людини не трактується як згода. Підвищення ризику негайно знижує автономність, а повернення до вищого рівня потребує кількох стабільних спостережень.

Алгоритм функціонує подієво. Після фіксації зміни проєкту формується незмінний знімок ситуації, активуються ролі, генеруються та локально оцінюються альтернативи, діагностуються конфлікти, реконфігурується колектив і вводиться погоджений план. Після виконання фіксується результат $y(t)$: фактичні строки, вартість, якість, використані ресурси, реалізовані ризики та відхилення від прогнозів.

Цей результат не залишається лише в журналі, а замикає адаптивний цикл:

$$m(z(t+1), \theta(t+1)) = \mathcal{U}(z(t), \theta(t), a^*(t), alloc^*(t), y(t), \{\hat{y}_i(t)\}_{i \in I(t)}), \quad (13)$$

де $z(t+1)$ – ситуація наступного циклу; $\theta(t)$ – адаптивні параметри, що охоплюють історичну результативність $perf_i(t)$, довіру $trust_i(t)$, ваги впливу $W(t)$ і накопичену історію $Y^{hist}(t)$; U – оператор ретроспективного оновлення; $a^*(t)$ і $alloc^*(t)$ – виконані управлінська дія та план розподілу; $y(t)$ – фактичний результат; $\hat{y}_i(t)$ – прогноз агента i .

Оновлення підвищує вплив агента лише тоді, коли його прогноз був каліброваним і рішення виконано в межах його ролі; систематична помилка або порушення припущень знижує результативність і довіру. Отримані параметри безпосередньо входять у формули (1), (2) та наступний вибір конфігурації, тому цикл є адаптивним, а не просто повторюваним.

Перевірювані властивості та інваріанти моделі й методу. Формальне поєднання стану IT-проєкту, локальних оцінок агентів, конфігурації колективу та плану виконання потребує контролю узгодженості між усіма складовими рішення. З цією метою визначено сукупність інваріантів, які перевіряються перед вибором альтернативи, під час реконфігурації колективу та після одержання фактичного результату. На відміну від значень цільових критеріїв, інваріанти не використовуються для ранжування: їх порушення виключає відповідний варіант із подальшого розгляду.

Першим є інваріант контекстної цілісності. Стан проєкту, активні події, ролі, альтернативи, локальні оцінки, конфігурація та план повинні належати одній версії ситуації. Фактичний результат пов'язується саме з тією альтернативою і тим планом, які були введені в дію. Завдяки цьому оцінки, сформовані для попередньої версії беклогу, не використовуються після його зміни, а результат перепланованих робіт не приписується початковому рішення. Збереження відомостей про походження альтернатив, використані дані, прогнози агентів і причини відсіву забезпечує семантичну простежуваність усього циклу. Основні умови перевірки запропонованої моделі та методу наведено в Табл. 1.

Інваріант допустимості встановлює пріоритет жорстких обмежень над прогнозованою корисністю. Висока оцінка альтернативи не може компенсувати відсутність необхідних повноважень, критичної компетентності, ресурсної місткості або людського погодження. Якщо допустиму конфігурацію чи план сформувавши неможливо, алгоритм переходить до контрольованого менеджерського режиму, а не обирає найкращий серед недопустимих варіантів.

Таблиця 1. Перевірювані інваріанти моделі й методу

Інваріант	Умова виконання	Результат порушення
Контекстна цілісність	Усі складові рішення мають спільну версію ситуації	Рішення блокується, знімок стану формується повторно
Допустимість	Виконано рольові, ресурсні, часові, регуляторні та ризикові обмеження	Варіант вилучається або передається менеджеру
Коректність конфлікту	Агенти оцінювали спільні альтернативи, а розбіжність перевищує очікуваний міжрольовий рівень	Конфліктне ребро не створюється або призначається уточнення
Стабільність	Виграш від реконфігурації перевищує витрати переходу та поріг гістерезису	Зберігається чинна допустима конфігурація

Джерело: складено авторами

Окремо контролюється незалежність ризику та впевненості. Низьке прогнозоване значення втрат не є достатньою підставою для автономного виконання за неповних даних.

Водночас висока впевненість у прогнозі не компенсує перевищення критичної межі ризику. Стабільність забезпечується поєднанням гістерезису та замороженого горизонту: перший обмежує часті зміни рольової структури, а другий захищає найближчі пакети робіт від необґрунтованого перепризначення. У разі порушення жорсткого обмеження або виникнення критичного ризику обидва механізми допускають негайну реконфігурацію.

Алгоритмічна реалізація та обчислювальна складність. Запропоновані модель і метод реалізуються у вигляді подієвого алгоритму, який запускається після фіксації суттєвої зміни стану ІТ-проєкту. Вхідними даними є актуальний знімок проєкту, реєстр агентів і їхніх компетентностей, чинна конфігурація, обмеження та історія виконаних рішень. Вихідними результатами є обрана управлінська дія, конфігурація агентів, план розподілу завдань і ресурсів та допустимий режим автономності.

Алгоритм колективного прийняття рішення та координація ШІ-агентів можна представити у вигляді наступної послідовності кроків.

1. Сформувані версійно зафіксований знімок стану проєкту та перевірити готовність даних.

2. Визначити активні ролі, оцінити придатність агентів за формулою (2) та призначити основних і резервних учасників.

3. Сформувані множину альтернатив відповідно до формули (3), виконавши їх нормалізацію, дедуплікацію та перевірку жорстких обмежень.

4. Отримати локальні оцінки агентів, обчислити попарні розбіжності за формулою (4) та сформувані конфліктний граф.

5. Побудувати допустимі конфігурації за формулою (7), оцінити їх за критерієм (8) і застосувати правило перемикавання (9).

6. Сформувані заявки агентів та визначити глобальний план розподілу завдань і ресурсів за формулами (10) і (11).

7. Оцінити ризик і впевненість за формулами (5) і (6), визначити режим виконання за формулою (12) та атомарно ввести конфігурацію і план.

8. Після завершення виконання зіставити фактичний результат із прогнозами агентів і за формулою (13) оновити результативність, довіру та ваги наступного циклу.

Основне обчислювальне навантаження зосереджене на двох операціях. Попарне виявлення конфліктів у гіршому випадку має квадратичну складність за кількістю активних агентів, однак рольова та інформаційна фільтрація дає змогу порівнювати лише агентів, які оцінювали спільні альтернативи. Глобальний розподіл взаємозалежних робіт за обмежених ресурсів є комбінаторною задачею [13], [20]. Для скорочення простору пошуку застосовуються попередній відсів недопустимих заявок, декомпозиція графа робіт, заморожений горизонт і локалізація перепланування областю впливу події.

Якщо точний план не вдається одержати в межах установленого часового бюджету, алгоритм використовує найкращий знайдений допустимий варіант. Введення конфігурації та плану виконується атомарно: агенти спочатку підтверджують ролі, повноваження, ресурси й версію даних, після чого координатор публікує спільне рішення. Невдале введення завершується відкотом, що запобігає одночасному використанню несумісних версій конфігурації та плану.

Експериментальна перевірка. Апробацію виконано на контрольованому сценарії динамічного ІТ-проєкту, використаному для перевірки багатоагентної системи підтримки рішень [19], [20]. Сценарій охоплював 18 розробників у трьох підкомандах, беклог зі 120 користувацьких історій та шість двотижневих спринтів. У спринтах 2, 4 і 5 вводилися розширення обсягу, затримка постачальника та втрата ключового виконавця. Пороги впевненості й ескалації фіксувалися до експерименту. Після кожного спринту фактичні строки, виконаний обсяг і результат пом'якшення ризиків передавалися до Y^{hist} ; прогнозні помилки використовувалися для оновлення $perf$ і $trust$ за наперед визначеним правилом, але порогові значення після одержання результатів не підлаштовувалися (Табл. 2).

Таблиця 2. Параметри експериментального сценарію

Параметр	Значення
Розробники / підкоманди	18 / 3
Беклог / тривалість	120 історій / 6 спринтів
Контрольовані збурення	Розширення обсягу; затримка постачальника; втрата виконавця
Базові підходи	Одноагентний ІІІ; Scrum лише з людським управлінням
Пороги	впевненість 0,75; ескалація 0,40

Джерело: складено авторами

Порівнювалися запропонована система, одноагентний підхід без внутрішнього механізму узгодження та Scrum з рішеннями керівника на межах спринтів (Табл. 3). Для оцінювання першої новизни використовували середню впевненість, частку розв'язаних конфліктів і частку хибних ескалацій. Другу новизну оцінювали за часткою автономно диспетчеризованих рішень, часом координації, стабільністю реакції, результативністю пом'якшення ризиків і здатністю зберегти допустимий план після збурення.

Таблиця 3. Динаміка основних показників протягом шести спринтів

Показник	1	2	3	4	5	6
Середня впевненість	0,71	0,77	0,83	0,86	0,88	0,89
ADR	0,58	0,64	0,70	0,61	0,75	0,81
ME	0,48	0,55	0,62	0,67	0,70	0,72

Джерело: складено авторами

Середня впевненість зросла з 0,71 до 0,89, ADR – з 0,58 до 0,81, а ME – з 0,48 до 0,72. Тимчасове зниження ADR у четвертому спринті з 0,70 до 0,61 було наслідком переходу до аварійного режиму після затримки постачальника. Це не є погіршенням методу: модель свідомо зменшила автономність, коли одночасно зросли ризик і невизначеність. Після стабілізації конфігурації та плану показник повернувся до висхідної траєкторії (Табл. 4).

Таблиця 4. Показники координації за режимами роботи

Показник	Номінальний	Підвищений	Критичний / аварійний	Загалом	Ціль
CRR	0,96	0,89	0,81	0,91	$\geq 0,85$
Час координації, хв	2,1	3,8	6,4	3,4	≤ 5
RSI	0,94	0,83	0,71	0,87	$\geq 0,80$
FER	–	–	–	0,13	$\leq 0,15$

Джерело: складено авторами

Загальне значення CRR 0,91 свідчить, що багатокomпонентна діагностика та адресний конфліктний граф забезпечили розв'язання більшості суттєвих суперечностей. Для одноагентної системи CRR дорівнює нулю за визначенням, оскільки вона не має внутрішніх незалежних позицій і механізму арбітражу. Частка хибних ескалацій 0,13 не перевищила встановленої межі 0,15, що підтверджує доцільність віднімання рольово очікуваної розбіжності та використання окремої моделі впевненості.

Середній час координації становив 3,4 хв і залишався нижчим за цільові 5 хв, хоча в критичному та аварійному режимах зростав до 6,4 хв через додаткові перевірки й людський маршрут. Індекс RSI 0,87 перевищив цільове значення 0,80. Отже, гістерезис і заморожений горизонт зменшили структурні коливання без втрати здатності негайно реагувати на критичну подію (Таблиця 5).

Усі три збурення були локалізовані. Під час втрати виконавця метод не переніс усі завдання до локально найкращого агента, а перебудував допустимий план з урахуванням

місткості, ролей, залежностей і резерву. Порівняно з управлінням лише за Scrum середня затримка прийняття рішення скоротилася на 68 %, а порівняно з одноагентним підходом випередження виявлення ризиків покращилося у 2,4 рази [19]. Частка автономного виконання була більшою на 29 відсоткових пунктів, але критичні рішення залишалися в людському маршруті.

Таблиця 5. Результати опрацювання контрольованих збурень

Подія	RDLT, хв	ME	Стратегія	Результат
Спринт 2 – розширення обсягу	7,2	0,68	Репріоритизація беклогу	Вплив на графік < 1 дня
Спринт 4 – затримка постачальника	5,8	0,71	Активізація резерву	Критичний шлях збережено
Спринт 5 – втрата виконавця	8,4	0,64	Перерозподіл робіт	Зниження швидкості < 8 %
Загалом	7,1	0,68	—	Усі ризики локалізовано

Джерело: складено авторами

Обговорення результатів. Експериментальні результати підтверджують взаємодоповнюваність двох складових. Перша новизна забезпечує замкнений простір рішення: спільний стан, рольову допустимість, альтернативи, локальні оцінки, конфлікти, виконаний план і фактичний результат. Зростання середньої впевненості протягом спринтів не трактується як довільне збільшення оптимізму: воно допускається лише після зіставлення прогнозів із фактичними наслідками та оновлення історичної результативності й довіри. Друга новизна перетворює модельний результат на структурно допустиму поведінку: залежно від стану, компетентності, щільності конфліктів і вартості переходу змінює ролі, повноваження, канали, механізм узгодження, автономність і план виконання.

Перевага над одноагентним підходом має архітектурний характер. Один агент може бути точним у межах власної моделі, але не здатний незалежно виявити суперечність між різними функціональними позиціями. Колектив створює таку можливість, проте потребує обмежень проти надмірної комунікації. Конфліктний граф і контекстний вибір механізму дають змогу залучати лише релевантних учасників, а робастний бал запобігає вибору конфігурації, яка виглядає найкращою лише через малу або зміщену історію.

Перевага над управлінням лише за Scrum пов'язана з подієвим, а не періодичним циклом. Агентний колектив фіксує відхилення під час спринту та готує альтернативи до того, як накопичений ефект потрапить на ретроспективу. Водночас запропонований підхід не усуває керівника проєкту. Режим виконання концентрує людську увагу на ситуаціях із високим ризиком, слабкими даними, суттєвим конфліктом або незворотними наслідками, а рутинні оборотні рішення диспетчеризуються автоматично.

Висновки. Удосконалено модель колективного прийняття рішень агентами штучного інтелекту в управлінні IT-проєктом, яка, на відміну від наявних, пов'язує поточний стан проєкту, спеціалізовані ролі агентів, множину альтернативних дій, локальні оцінки агентів, конфлікти між ними та результати виконання прийнятих рішень у єдиному адаптивному циклі. Формула (13) замикає цей цикл: фактичний результат і похибки локальних прогнозів оновлюють стан, історичну результативність, довіру та ваги наступного рішення.

Удосконалено метод координації спеціалізованих агентів штучного інтелекту, який передбачає виявлення суперечностей між запропонованими діями та динамічний розподіл ролей і повноважень агентів з урахуванням стану IT-проєкту, компетентності агентів, щільності конфліктів і витрат на зміну ролей. Метод відсіює недопустимі конфігурації до ранжування, застосовує гістерезис і атомарне введення, а за відсутності допустимого кандидата активує безпечний стан і людський маршрут.

Апробація на шестиспринтовому сценарії показала зростання впевненості до 0,89, ADR до 0,81 і ME до 0,72; CRR становив 0,91, RSI – 0,87, FER – 0,13. Тимчасове зниження автономності під час критичної події підтвердило, що метод підпорядковує автоматизацію ризику та людському контролю.

Конфлікт інтересів: Автори декларують, що не мають конфлікту інтересів, зокрема фінансового, особистого, авторського чи будь-якого іншого характеру, який міг би вплинути на дослідження, а також на результати, опубліковані в цій статті

Фінансування: Дослідження проводилося без фінансової підтримки

Доступність даних: Усі дані доступні в цифровій або графічній формі в основному тексті рукопису

Використання засобів штучного інтелекту (ШІ): Автори підтверджують, що не застосовували інструментальні засоби ШІ для написання цієї роботи

СПИСОК ЛІТЕРАТУРИ

1. Fernandez, D. J. & Fernandez, J. D. “Agile project management – agilism versus traditional approaches”. *Journal of Computer Information Systems*. 2008; 49 (2): 10–17. DOI: <https://doi.org/10.1080/08874417.2009.11646044>.
2. Conforto, E. C. & Amaral, D. C. “Agile project management and stage-gate model – a hybrid framework for technology-based companies”. *Journal of Engineering and Technology Management*. 2016; 40: 1–14. DOI: <https://doi.org/10.1016/j.jengtecman.2016.02.003>.
3. Dingsøyr, T., Nerur, S., Baliyepally, V. & Moe, N. B. “A decade of agile methodologies: towards explaining agile software development”. *Journal of Systems and Software*. 2012; 85 (6): 1213–1221. DOI: <https://doi.org/10.1016/j.jss.2012.02.033>.
4. Serrador, P. & Pinto, J. K. “Does Agile work? A quantitative analysis of agile project success”. *International Journal of Project Management*. 2015; 33 (5): 1040–1051. DOI: <https://doi.org/10.1016/j.ijproman.2015.01.006>.
5. Dikert, K., Paasivaara, M. & Lassenius, C. “Challenges and success factors for large-scale agile transformations: a systematic literature review”. *Journal of Systems and Software*. 2016; 119: 87–108. DOI: <https://doi.org/10.1016/j.jss.2016.06.013>.
6. Maylor, H., Vidgen, R. & Carver, S. “Managerial complexity in project-based operations: A grounded model and its implications for practice”. *Project Management Journal*. 2008; 39 (1): S15–S26. DOI: <https://doi.org/10.1002/pmj.20057>.
7. Wooldridge, M. & Jennings, N. R. “Intelligent agents: theory and practice”. *The Knowledge Engineering Review*. 1995; 10 (2): 115–152. DOI: <https://doi.org/10.1017/S0269888900008122>.
8. Dorri, A., Kanhere, S. S. & Jurdak, R. “Multi-agent systems: a survey”. *IEEE Access*. 2018; 6: 28573–28593. DOI: <https://doi.org/10.1109/ACCESS.2018.2831228>.
9. Stone, P. & Veloso, M. “Multiagent systems: a survey from a machine learning perspective”. *Autonomous Robots*. 2000; 8 (3): 345–383. DOI: <https://doi.org/10.1023/A:1008942012299>.
10. Lesser, V. R. “Reflections on the nature of multi-agent coordination and its implications for an agent architecture”. *Autonomous Agents and Multi-Agent Systems*. 1998; 1 (1): 89–111. DOI: <https://doi.org/10.1023/A:1010046623013>.
11. Taboada, I., Daneshpajouh, A., Toledo, N. & de Vass, T. “Artificial intelligence enabled project management: a systematic literature review”. *Applied Sciences*. 2023; 13 (8): 5014. DOI: <https://doi.org/10.3390/app13085014>.
12. Dam, H. K., Tran, T., Grundy, J., Ghose, A. K. & Kamei, Y. “Towards effective AI-powered agile project management”. *Proceedings of the Forty-First International Conference on Software Engineering: New Ideas and Emerging Results*. 2019. p. 41–44. DOI: <https://doi.org/10.1109/ICSE-NIER.2019.00019>.
13. He, N., Zhang, D. Z. & Yuce, B. “Integrated multi-project planning and scheduling – a multiagent approach”. *European Journal of Operational Research*. 2022; 302 (2): 688–699. DOI: <https://doi.org/10.1016/j.ejor.2022.01.018>.

14. Hodovychenko, M. A., Lobachev, M. V., Boeiv, O. P., Linnyk, O. O. & Horalik, O. E. “Adaptive and coordinated IT project management in dynamic environments: A multi-agent AI perspective”. *Applied Aspects of Information Technology*. 2026; 9 (2): 208–219. DOI: <https://doi.org/10.15276/aa.2026.15>.
15. Yager, R. R. “On ordered weighted averaging aggregation operators in multicriteria decisionmaking”. *IEEE Transactions on Systems, Man, and Cybernetics*. 1988; 18 (1): 183–190. DOI: <https://doi.org/10.1109/21.87068>.
16. Jøsang, A., Ismail, R. & Boyd, C. “A survey of trust and reputation systems for online service provision”. *Decision Support Systems*. 2007; 43 (2): 618–644. DOI: <https://doi.org/10.1016/j.dss.2005.05.019>.
17. Hodovychenko, M. A., Lobachev, M. V., Linnyk, O. O., Horalik, O. Y. & Revenko, P. H. “Coordinated decision support for IT projects using multi-agent artificial intelligence systems”. *Herald of Advanced Information Technology*. 2026; 9 (3): 429–442. DOI: <https://doi.org/10.15276/hait.2026.28>.
18. Pinyol, I. & Sabater-Mir, J. “Computational trust and reputation models for open multi-agent systems: a review”. *Artificial Intelligence Review*. 2013; 40 (1): 1–25. DOI: <https://doi.org/10.1007/s10462-011-9277-z>.
19. Smith, R. G. “The Contract Net Protocol: high-level communication and control in a distributed problem solver”. *IEEE Transactions on Computers*. 1980; C-29 (12): 1104–1113. DOI: <https://doi.org/10.1109/TC.1980.1675516>.
20. Pellerin, R. & Perrier, N. “A review of methods, techniques and tools for project planning and control”. *International Journal of Production Research*. 2019; 57 (7): 2160–2178. DOI: <https://doi.org/10.1080/00207543.2018.1524168>.

Отримано 22.07.2026

Отримано після перегляду 28.08.2026

Прийнято 04.09.2026

DOI: <https://doi.org/10.15276/ict.03.2026.26>

UDC 005.8:004.89

A model of collective decision-making and a method for context-dependent coordination of artificial intelligence agents in IT project management

Oleksii O. Linnyk¹⁾

ORCID: <https://orcid.org/0009-0007-3060-1412>; alexeylinnik@stud.op.edu.ua

Serhii I. Shutov¹⁾

ORCID: <https://orcid.org/0009-0006-4190-3988>; 3234816@as.op.edu.ua

¹⁾ Odesa Polytechnic National University, 1, Shevchenko Av. Odesa, 65044, Ukraine

ABSTRACT

This article examines collective adaptive management decisions by specialized artificial intelligence agents in IT projects. It aims to improve decision validity, consistency, and adaptability by refining a collective decision model and an agent coordination method. The model combines project state, active roles, feasible alternatives, local assessments, conflicts, and execution results in one adaptive cycle. Results are compared with forecasts and used to update project state, performance, trust, and influence weights. Conflict is defined as a contradiction exceeding the difference normally expected between roles in the current context. Version consistency, feasibility, traceability, and stability are enforced as invariants. The coordination method detects contradictions and dynamically changes roles and authorities according to project state, agent competence, conflict graph density, uncertainty, and reconfiguration costs. The algorithm creates a versioned snapshot, activates roles, forms and evaluates alternatives, selects a feasible configuration, allocates tasks and resources, determines autonomy, and updates parameters after execution. Hysteresis and a frozen horizon limit reassignments, while configuration and plan are introduced atomically. Testing involved eighteen developers, three subteams, one hundred and twenty user stories, and six sprints lasting two weeks. Confidence increased from zero point seven one to

zero point eight nine; autonomous dispatching from zero point five eight to zero point eight one; and risk mitigation effectiveness from zero point four eight to zero point seven two. The resolved conflict rate was zero point nine one, the response stability index was zero point eight seven, and the false escalation rate was zero point one three. Results confirm overall effectiveness.

Keywords: IT project management; multi-agent systems; collective decision-making; adaptive feedback; dynamic role allocation; agent coordination; decision conflicts; controlled autonomy

ІНФОРМАЦІЯ ПРО АВТОРІВ



Лінник Олексій Олександрович – аспірант, кафедра Штучного інтелекту та аналізу даних. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Галузь досліджень: штучний інтелект, мультиагентні системи, розподілені обчислення, імітаційне моделювання, машинне навчання

Oleksii O. Linnyk – postgraduate, Artificial Intelligence and Data Analysis Department. Odesa Polytechnic National University, 1, Shevchenko Av. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0007-3060-1412>; alexeylinnik@stud.op.edu.ua

Research field: artificial intelligence, multi-agent systems, distributed computing, simulation modeling, machine learning



Шутов Сергій Ігорович – аспірант, кафедра Штучного інтелекту та аналізу даних. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Галузь досліджень: інтелектуальні інформаційні системи, багатоагентні системи та колективне прийняття рішень, методи машинного навчання, адаптивне управління IT-проектами, моделювання та управління ризиками

Serhii I. Shutov – postgraduate, Artificial Intelligence and Data Analysis Department. Odesa Polytechnic National University, 1, Shevchenko Av. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0006-4190-3988>; 3234816@as.op.edu.ua

Research field: intelligent information systems, multi-agent systems and collective decision-making, machine learning methods, adaptive IT project management, risk modeling and management.