

*О. О. ЛИСЕНКО, І. В. КОНОНЕНКО***МОДЕЛЬ ВИБОРУ СТОРОННІХ ІНСТРУМЕНТІВ В ІТ ПРОЄКТАХ**

Предметом дослідження є модель вибору сторонніх інструментів розробки в ІТ-проектах, що враховує трудомісткість, витрати, якість, ризики, технологічний ефект та обмеження проєктного середовища. Метою дослідження є інтеграція цих п'яти показників у єдину модель вибору сторонніх інструментів розробки шляхом формалізації задачі як багатокритеріальної оптимізаційної з чітко визначеними цільовими функціями та обмеженнями, що дає змогу усунути наявну прогалину в цій науковій сфері. До завдань дослідження належать: аналіз і узагальнення наукових підходів до оцінювання сторонніх інструментів розробки; встановлення відповідності критеріїв оцінювання інструменту цілям оцінювання ефективності його впровадження у проєкт; формалізація п'яти інтегральних показників у вигляді кількісної моделі, придатної до математичної обробки; побудова багатокритеріальної оптимізаційної моделі вибору інструментів з урахуванням ресурсних, бюджетних та якісних обмежень; демонстрація застосування моделі на числовому прикладі з кількома альтернативними інструментами. У дослідженні використано метод TOPSIS для ранжування альтернатив за інтегральними показниками, а також механізм штрафних функцій для врахування порушення критичних бізнес-обмежень. Отримані результати показують, що запропонований підхід дає змогу поєднати в єдиній формалізованій схемі оцінювання трудомісткості, витрат, якості, ризиків та технологічного ефекту сторонніх інструментів, уніфікувати різномірні оцінки в нормалізованій матриці рішень, обчислювати узагальнені показники привабливості альтернатив та враховувати штрафи за порушення обмежень. Числовий приклад демонструє ситуацію, коли найкращою виявляється альтернатива, що не є лідером за окремими критеріями, але забезпечує прийнятний баланс інтегральних показників без перевищення критичних обмежень. Висновки. Наукова новизна дослідження полягає у побудові першої формалізованої багатокритеріальної оптимізаційної моделі вибору сторонніх інструментів розробки, у якій виділено п'ять інтегральних показників, включно з технологічним ефектом, а також запропоновано їх кількісне подання. Запропонована модель, реалізована на основі методу TOPSIS зі штрафними функціями, розширює можливості систем підтримки прийняття рішень в ІТ-проектах, дозволяючи ранжувати альтернативи з урахуванням обмежень щодо ресурсів, бюджету, ризиків і якості у складних та динамічно змінюваних технологічних середовищах.

Ключові слова: сторонні інструменти розробки; міграція; багатокритеріальне прийняття рішень; TOPSIS; модель; оптимізація; вибір сторонніх бібліотек.

*A. LYSENKO, I. KONONENKO***SELECTION MODEL FOR THIRD-PARTY LIBRARIES IN IT PROJECTS**

This study focuses on a model for selecting third-party libraries in IT projects, taking into account efforts, costs, quality, risks, technological effects, and project environment constraints. The aim of the study is to integrate these five indicators into a cohesive model that formalizes the selection process as a multi-criteria optimization problem, with clearly defined objective functions and constraints. This approach addresses a significant gap in the existing area. The study tasks include analyzing and synthesizing scientific methods for evaluating third-party libraries, establishing the relationship between evaluation criteria and project effectiveness objectives, formalizing the five key indicators into a quantitative model suitable for mathematical processing, constructing a multi-criteria optimization model for library selection that considers resource, budgetary, and quality constraints, and demonstrating the model's application through a numerical example featuring several alternative libraries. The study employs the TOPSIS method to rank alternatives based on five indicators, alongside a penalty-function mechanism to address violations of critical business constraints. The results indicate that the proposed approach allows for a unified evaluation of effort, costs, quality, risks, and technological effects of third-party libraries. It successfully normalizes diverse assessments into a decision matrix, computes aggregate measures for the overall attractiveness of alternatives, and incorporates penalties for any constraint violations. The numerical example highlights a case where the optimal choice is not the leader in individual criteria but provides an acceptable balance of integral indicators without exceeding critical constraints. Conclusions: This study provides a novel contribution by developing the first formalized multi-criteria optimization model for selecting third-party libraries. This model identifies five indicators, including technological effects, and proposes their quantitative representation. Implemented using the TOPSIS method with penalty functions, the proposed model enhances the capabilities of decision-support systems in IT projects, enabling the ranking of alternatives while accommodating resource, budget, risk, and quality constraints in complex and dynamically changing technological environments.

Keywords: third-party libraries; migration; multi-criteria decision-making; TOPSIS; model; optimization; third-party library selection.

Вступ. У сучасних ІТ-проектах сторонні інструменти розробки стали невід'ємним елементом технологічного стека: вони прискорюють розроблення, знижують початкові витрати та дають змогу зосередитися на предметній області, а не на типових інфраструктурних задачах. Водночас використання таких інструментів супроводжується низькою ризиків – від залежності від конкретного постачальника й можливого припинення підтримки до прихованих витрат на інтеграцію, навчання команди та подальші міграції. На практиці вибір інструменту часто здійснюється на основі попереднього досвіду, рекомендацій колег або неформальних критеріїв, що ускладнює обґрунтування рішення та його відтворюваність у різних проєктах.

Стан досліджень. У науковій площині питання вибору сторонніх рішень традиційно розв'язувалося переважно для компонентів програмного забезпечення (COTS) та модульних програмних систем. Для таких задач запропоновано низку формальних моделей із чітко визначеними цільовими функціями, обмеженнями та методами знаходження оптимального набору компонентів. Натомість саме для сторонніх інструментів розробки, які працюють на рівні коду застосунку й безпосередньо впливають на архітектуру, супровід і еволюцію системи, формалізовані підходи лише починають формуватися. Більшість наявних досліджень зосереджені або на переліку факторів [1, 2, 3], що враховуються розробниками, або на побудові рекомендаційних систем [4, 5], які оптимізують точність передбачення вибору за історичними даними, але не інтегрують повною мірою трудомісткість,

витрати, якість, ризики та технологічний ефект у єдину модель. Наразі відсутні систематичні дослідження, що розглядають вибір сторонніх інструментів розробки як оптимізаційну задачу з формалізованими цільовими функціями та обмеженнями.

На цьому тлі особливу цінність набувають роботи, де вибір сторонніх інструментів розробки розглядається як формалізована задача прийняття рішень із заданою системою показників та визначеною процедурою оптимізації альтернатив. Саме до цього напрямку належать подані матеріали [6]: вони демонструють перші спроби поєднати структуровану оцінку бібліотек, візуальні засоби аналізу та математичні методи підтримки вибору, а також окреслюють прогалини, які надалі можуть бути заповнені шляхом побудови оптимізаційних моделей, що враховують комплексні проєктні цілі.

У науковій літературі як окремий, але суміжний із вибором сторонніх інструментів розробки напрям розглядаються також задачі вибору COTS-рішень, які становлять інтерес з огляду на запропоновані в них моделі та методи розв'язання задачі. Для коректного відмежування цих двох класів стороннього програмного забезпечення доцільно коротко уточнити їх відмінності.

COTS-рішення розглядаються як готові програмні продукти, що використовуються «як є» та впроваджуються й конфігуруються переважно на рівні всієї інформаційної системи або організації. Прикладами таких рішень можуть бути Jira, SAP, Salesforce, корпоративні системи класу CRM/ERP, платформи для керування документами тощо.

Натомість сторонні інструменти розробки виступають у ролі бібліотек, SDK та фреймворків, які інтегруються безпосередньо у програмний продукт на рівні вихідного коду та використовуються через API. На відміну від COTS-рішень, такі інструменти зазвичай підключаються до коду застосунку, збираються разом із застосунком, а команда проєкту має більший контроль над вибором версій, оновленнями та заміною цих компонентів.

У роботі [7] розглядається задача вибору COTS-компонентів для модульної системи програмного забезпечення. Для кожного модуля є кілька альтернативних COTS-продуктів, і потрібно обрати по одному варіанту так, щоб: максимізувати якість системи; не перевищити заданий бюджет на закупівлю компонентів. Це по суті перше формальне формулювання задачі «якість ↔ вартість» для COTS компонентів у вигляді оптимізаційної моделі. Обидві моделі — це формулювання задачі цілочисельного програмування з однією цільовою функцією (якість) і ресурсним обмеженням (вартість).

У роботі [8] розглядається задача побудови модульної програмної системи на основі COTS-компонентів. Для кожного модуля задається кілька альтернативних COTS-продуктів, які характеризуються числовими оцінками показників якості та вартості. Постановка формулюється як багатокритеріальна оптимізаційна модель: необхідно обрати рівно один варіант для кожного модуля так, щоб

отримана система одночасно задовольняла цільові орієнтири щодо якості та не перевищувала задані витратні обмеження. Модель включає три цільові функції — мінімізацію загальної вартості системи, мінімізацію часу виконання/постачання та максимізацію інтегрального показника якості — а також систему ресурсних обмежень і обмежень сумісності між компонентами. Розв'язання задачі здійснюється з використанням лексикографічного цільового програмування.

Робота [9] пропонує формальну оптимізаційну модель вибору компонентів для компонентно-орієнтованої програмної системи (CBSS). Автори виходять із підходу: спочатку формується набір модулів системи, а далі для кожного модуля потрібно обрати один із наявних варіантів компонента. Основний внесок роботи — врахування не лише функціональної придатності компонентів, а й їхнього впливу на структурну якість системи через показники залежності та цілісності між модулями. Формулюється двоцільова постановка: максимізація функціональної продуктивності/корисності всієї CBSS та одночасна максимізація цілісності й мінімізація залежності модулів. Для розв'язання задачі комбінаторної оптимізації використовується генетичний алгоритм, який шукає оптимальний набір компонентів по модулях.

Автори роботи [10] запропонували цілочислову оптимізаційну модель вибору програмних компонентів у ситуації, коли організація паралельно розробляє кілька застосунків. На відміну від більш ранніх моделей, що орієнтувалися на одну систему, автори вводять показник повторного використання компонентів і переглядають формулювання матриці сумісності, щоб одночасно оптимізувати повторне використання компонентів та їхню взаємну сумісність. Задачу розв'язують за допомогою спеціалізованого генетичного алгоритму, а результати демонструють, що спільна оптимізація для множини застосунків дозволяє досягти вищого рівня повторного використання і потенційно знизити сумарні витрати портфеля CBSD-проєктів.

У роботі [11] формалізують задачу вибору компонентів у CBSS як задачу цілочисельного лінійного програмування, у якій для кожного компонента вводяться бінарні змінні «обрано / не обрано». Цільова функція одночасно: максимізує внутрішньомодульні взаємодії, мінімізує витрати на розробку та адаптацію компонентів; а система обмежень забезпечує дотримання ресурсних обмежень і приблизно однакову «гранулярність» модулів (кількість компонентів у модулі). Модель розв'язується стандартними методами цілочисельної оптимізації.

Автори роботи [12] розглядають задачу оцінювання й вибору програмних компонентів для модульної системи, у якій одночасно розробляється кілька програмних застосунків. Для кожної функціональної частини системи передбачено кілька альтернативних компонентів: як готові комерційні програмні продукти. На першому етапі автори оцінюють альтернативи за низкою критеріїв (витрати,

час постачання або розробки, показники якості, надійність, можливість повторного використання, сумісність між компонентами) із використанням методу TOPSIS, у результаті чого кожному компоненту надається узагальнена числова оцінка придатності. На другому етапі будується багатокритеріальна оптимізаційна модель, яка одночасно вирішує, які саме компоненти обрати для кожного застосунку та в яких випадках доцільніше придбати готове рішення, а в яких – розробляти компонент самостійно. Цілі моделі полягають у підвищенні загальної якості системи та зниженні витрат і тривалості впровадження за наявності обмежень на надійність, сумісність і повторне використання компонентів. Для узгодження пріоритетів між цими цілями застосовується нечіткий інтерактивний підхід: особі, що приймає рішення, пропонуються варіанти рішень із різним ступенем задоволення за витратами, часом і якістю, після чого обирається прийнятний компроміс.

Робота [13] пропонує підхід до вибору програмних компонентів (COTS) на основі методики доказового міркування, яка спирається на теорію доказів Демпстера–Шефера. Автор формує ієрархію критеріїв якості (зручність, продуктивність, безпека, сумісність тощо) і задає оцінки компонентів у вигляді лінгвістичних рівнів із невизначеністю. Теорія доказів використовується для того, щоб кожна оцінка (з невизначеністю) стала доказом для рівня якості компонента. Ці оцінки перетворюються на ймовірнісні маси, агрегуються по всій ієрархії, після чого для кожного компоненту обчислюється очікувана корисність та інтервал впевненості. Докази комбінуються за узагальненим правилом Демпстера (ER-rule), яке базується на ортогональній сумі мас і враховує ваги критеріїв.

Робота [14] пропонує цілочисельну лінійну модель для задачі скорочення тривалості робіт шляхом додаткових ресурсів у проекті в СРМ-мережі, яка одночасно враховує час, вартість, якість і ризик. Автори інтегрують у цільову функцію: прямі витрати на скорочення робіт, витрати на забезпечення якості, витрати через дефекти, очікувані витрати від ризиків. Цільова функція мінімізує сукупну вартість проекту, а кінцевий строк виконання задається як жорстке обмеження. На прикладі невеликої СРМ-сітки автори показують, що модель дозволяє знайти такий варіант скорочення робіт і заходів з якості, який забезпечує виконання проекту в заданий термін при мінімальних сумарних витратах з урахуванням ризиків.

Класичні підходи до управління проектами описують проект через три ключові обмеження: час, вартість та якість. У більшості джерел ці обмеження розглядаються як взаємопов'язані змінні, де зміна одного параметра неминуче впливає на інші. З часом ця модель еволюціонує й доповнюється додатковими критеріями [15], зокрема ризиками та стратегічними вигодами, що відображає перехід від вузького розуміння «успіху за графіком і бюджетом» до орієнтації на досягнення результатів і довгострокового ефекту для організації.

У роботі [16] сформульовано концепцію піраміди управління проектом, у межах якої проект характеризують не лише класичною тріадою «час–вартість–якість», а сукупністю взаємопов'язаних вимірів: змістом, часом реалізації, вартістю, якістю продукту, ризиками та цінністю (ефектами) результатів проекту. Цінність трактується як багатовимірною характеристика, що може оцінюватися економічними, соціально-політичними, екологічними, технологічними ефектами.

У сфері програмної інженерії ці ідеї були конкретизовані в роботах Б. Боєма, у яких закладено засади економіки програмної інженерії. У його монографії «Економіка програмної інженерії» [17] трудомісткість, вартість, якість і ризики розглядаються як ключові змінні, що визначають доцільність і успішність програмних проєктів, а завдання оцінювання зусиль, вибору альтернатив і порівняння сценаріїв прямо формулюються як проблеми багатокритеріальної оптимізації.

Паралельно з економічними моделями важливе місце посідає поняття якості. Міжнародний стандарт ISO 9000 визначає якість як «ступінь, до якого сукупність властивих характеристик об'єкта задовольняє споживача», що підкреслює її багатовимірний і відносний характер: якість завжди розглядається у зв'язку з набором очікувань і потреб зацікавлених сторін [18, 19].

Ще один системний блок критеріїв у сучасних моделях становлять ризики. У проєктному менеджменті та інвестиційному аналізі дедалі ширше застосовуються багатокритеріальні методи оцінювання ризику, в яких імовірності настання подій, можливі збитки, чутливість проєкту до змін середовища та інші фактори поєднуються в інтегральні показники «ризикованості» окремих сценаріїв або альтернатив [20].

Окремий блок сучасних досліджень пов'язаний із технологічною оцінкою та вибором технологій, де критерії групують ще й за вигодами, рівнем інноваційності, технологічною ефективністю та впливом на зацікавлені сторони [21].

Таким чином, проведений аналіз літератури засвідчує, що виокремлення інтегральних показників трудомісткості, витрат, якості, ризиків та технологічного ефекту узгоджується з ключовими напрямками розвитку проєктного менеджменту, економіки програмної інженерії та багатокритеріальної оцінки технологій. Водночас у більшості виявлених досліджень детально опрацьовано лише один або два з перелічених показників, тоді як для комплексної підтримки прийняття рішень доцільним є одночасне врахування всіх п'яти показників та їхніх взаємозв'язків. Особливо показовим є те, що інтегральний показник технологічного ефекту фактично не формується як самостійний об'єкт аналізу: у наявних роботах розглядаються лише окремі його складові, фрагментарно та залежно від контексту дослідження. Виявлений розрив між теоретичними підходами та практичними потребами ІТ-проєктів мотивує подальший виклад: наступні розділи статті

формалізують систему критеріїв для вибору сторонніх інструментів розробки та пропонують модель інтегральних показників, спеціально адаптовану до специфіки сучасних ІТ-проектів.

Мета статті. Метою є інтеграція показників трудомісткості, витрат, якості, ризиків та технологічного ефекту в єдину модель вибору сторонніх інструментів розробки шляхом формалізації цієї задачі як багатокритеріальної оптимізаційної з чітко визначеними цільовими функціями та обмеженнями, що дає змогу усунути наявну прогалину в цій сфері.

Відповідність систематизованих критеріїв інструменту цілям оцінювання ефективності його впровадження у проєкт. З метою спрощення

подальшого моделювання 44 критерії вибору інструменту згруповано у п'ять інтегральних показників, що відображають основні цілі оцінювання ефективності його впровадження в ІТ-проект: трудомісткість, витрати, якість, ризиків та технологічний ефект.

Таблиця 1 відображає відповідність між технічними критеріями інструменту та інтегральними показниками, які характеризують основні напрями оцінювання ефективності його впровадження. По вертикалі наведено технічні критерії інструменту, по горизонталі – інтегральні показники.

Таблиця 2 – відповідність між людськими критеріями та інтегральними показниками. Таблиця 3 – відповідність між економічними критеріями та інтегральними показниками.

Таблиця 1 – Відповідність технічних критеріїв інструменту цілям оцінювання ефективності його впровадження у проєкт

	Трудомісткість	Витрати	Якість	Ризики	Технологічний ефект
Зручність використання	✓		✓		
Рівень досконалості документації	✓		✓	✓	
Відповідність функціоналу цілям та завданням проєкту	✓		✓	✓	
Тестованість	✓		✓	✓	
Зрілість та стабільність	✓		✓	✓	✓
Частота випусків	✓		✓	✓	✓
Сумісність з іншими інструментами	✓		✓	✓	✓
Відповідність архітектурі проєкту	✓		✓	✓	✓
Кросплатформеність	✓		✓		✓
Нечітке визначення меж інтеграції інструменту і задач, котрі він вирішує	✓			✓	
Розмір інструменту	✓	✓			
Якість гарантійного та після гарантійного сервісу	✓	✓	✓	✓	
Активність підтримки	✓	✓	✓	✓	✓
План розвитку та стратегічне бачення			✓	✓	✓
Відповідність новизни інструменту рівню інноваційності цілей та завдань проєкту			✓	✓	✓
Продуктивність			✓	✓	
Рівень безпеки			✓	✓	
Стійкість до збоїв			✓	✓	

Таблиця 2 – Відповідність людських критеріїв інструменту цілям оцінювання ефективності його впровадження у проєкт

	Трудомісткість	Витрати	Якість	Ризики	Технологічний ефект
Досвід використання	✓		✓	✓	
Знання та сприйняття	✓		✓	✓	
Брак зацікавленості співробітників у проєкті	✓		✓	✓	
Згода з вибором	✓			✓	
Можливість знайти фахівців для підтримки інструменту	✓	✓		✓	✓
Складність вивчення	✓	✓		✓	
Компетентність команди	✓	✓	✓	✓	
Рейтинг розробника		✓		✓	✓
Популярність		✓		✓	✓
Відповідність корпоративній культурі та політиці				✓	

Таблиця 3 – Відповідність економічних критеріїв інструменту цілям оцінювання ефективності його впровадження у проєкт

	Трудомісткість	Витрати	Якість	Ризики	Технологічний ефект
Нереалістичність встановлених термінів впровадження	✓			✓	
Необхідний час на впровадження	✓	✓			
Витрати на використання та інтеграцію		✓			
Витрати на ліцензію для володіння		✓			
Зручність умов придбання		✓		✓	
Перевищення проєктних витрат над запланованим бюджетом		✓		✓	
Невигідний профіль грошового потоку		✓		✓	
Потрібні інвестиції перевищують очікуваний прибуток		✓		✓	
Вигода від використання (якщо можливо оцінити, то – дохід від використання)		✓		✓	✓
Розповсюдженість інструменту		✓		✓	✓
Підприємства (організації) розробника інструменту		✓		✓	✓
Імідж інструменту				✓	✓
Привабливість для споживачів				✓	✓
Країни походження інструменту				✓	
Відсутність схвалення фінансування з боку клієнта				✓	

Цільові функції. У моделі вибору інструменту розробки критерії згруповані у п'ять цільових функцій, кожна з яких відображає окремий вимір прийняття рішень. Припустимо розглядаються n інструментів, $j = 1, n$.

Трудомісткість – це сукупність людських ресурсів і зусиль, необхідних для освоєння, інтеграції/міграції, налаштування, документування, навчання та подальшого використання інструменту. Вони включають обсяг часу роботи. На трудомісткість впливають чинники (див. табл. 1 та 2), що визначають швидкість і ефективність адаптації: рівень досвіду та розподіл компетенцій у команді, потреба й обсяг навчання, якість і доступність документації/прикладів, а також складність вивчення та інтеграції інструменту в наявну архітектуру. Вагомими є й організаційно-соціальні чинники: готовність і мотивація команди, погодженість змін, реалістичність строків і доступність ресурсів. Серед технічних чинників — зрілість і стабільність інструменту, зручність API та наборів програмних інструментів, сумісність і наявність інтеграцій, можливості автоматизації, а також його розмір і зручність повсякденного використання.

Трудомісткість для інструменту j дорівнює:

$$E_j = \sum_{i=1}^{m_j} (E_{edu,ij} + E_{impl,ij} + E_{del,id}) + E_{use,j} \quad (1)$$

де m_j – кількість осіб, залучених до процесу впровадження інструменту j ;

$E_{edu,ij}$ – трудомісткість для особи i на освоєння інструменту j : вивчення принципів його використання, аналіз документації, дослідницькі активності, пов'язані з тим, як саме інструмент має бути інтегрований у проєкт;

$E_{impl,ij}$ – трудомісткість для особи i на впровадження інструменту j в проєкт: інтеграція в кодову базу, налаштування середовищ, CI/CD, міграція конфігурацій тощо;

$E_{del,id}$ – трудомісткість для особи i на видалення існуючого інструменту d , що замінюється інструментом j ;

$E_{use,j}$ – інтегральна трудомісткість подальшого використання інструменту j у типових повсякденних або періодичних повторюваних завданнях у процесі розробки продукту після його впровадження, розрахована для типового циклу розробки, прийнятого як період оцінювання.

Перша цільова функція описує мінімізацію трудомісткості, пов'язаної із впровадженням інструменту, і має вигляд:

$$\sum_{j=1}^n E_j x_j \rightarrow \min_{x_j} \quad (2)$$

де x_j – бінарна змінна, що відображає вибір інструменту, $x_j \in \{0,1\}$.

Якщо проєкт вимагає вибору лише одного інструменту з множини доступних, додається обмеження:

$$\sum_{j=1}^n x_j = 1$$

Для врахування обмеження часових ресурсів команди, яке визначає обсяг людських ресурсів, доступних для впровадження інструменту, вводиться умова:

$$\sum_{j=1}^n E_j x_j \leq E_{max}, \quad (3)$$

де E_{max} – максимальна трудомісткість, з якою зможе впоратися команда, тобто сукупний обсяг часових ресурсів команди, який може бути виділений на впровадження інструменту без перевищення можливостей команди.

Витрати – це сукупність фінансових і матеріальних ресурсів, необхідних для придбання, впровадження та подальшої підтримки інструменту. Вони охоплюють придбання ліцензії/підписки, оплату за навчання використанню, оплату консультацій та послуг постачальника інструменту, оплату праці при впровадженні інструменту. На витрати впливають такі чинники: бюджетні обмеження, ризик перевищення запланованих витрат, профіль грошових потоків і співвідношення очікуваних інвестицій та вигід від використання, умови придбання, зрілість постачальника, доступність фахівців для підтримки, валютні ризики, а також потенційний економічний ефект або дохід від застосування інструменту.

Витрати на оплату праці при впровадженні інструменту j визначаються як:

$$C_{wj} = \sum_{i=1}^{m_j} (E_{edu,ij} + E_{impl,ij} + E_{del,id}) r_i \quad (4)$$

де r_i – погодинна ставка особи i , яка бере участь у впровадженні інструменту.

Витрати на придбання прав на використання інструменту j командою розробки виконавця визначається як $C_{dev,j}$.

Витрати замовника на ліцензії або права користування інструментом j протягом періоду t років визначається як $C_{prod,j}$.

Витрати при експлуатації інструменту у замовника протягом періоду t років визначається як $C_{use,j}$.

Сумарні витрати на впровадження інструменту j становлять:

$$C_j = C_{wj} + C_{dev,j} + C_{prod,j} + C_{use,j} \quad (5)$$

Друга цільова функція для мінімізації витрат, пов'язаних із впровадженням інструменту, має вигляд:

$$\sum_{j=1}^n C_j x_j \rightarrow \min_{x_j} \quad (6)$$

Для запобігання перевищенню бюджету вводиться обмеження:

$$\sum_{j=1}^n (C_{wj} + C_{dev,j}) x_j \leq C_{max,int} \quad (7)$$

де $C_{max,int}$ – максимально допустимі сукупні витрати на впровадження інструменту.

$$\sum_{j=1}^n (C_{prod,j} + C_{use,j}) x_j \leq C_{max,use} \quad (8)$$

де $C_{max,use}$ – максимально допустимі сукупні витрати на експлуатацію інструменту.

Під якість стороннього інструменту будемо розуміти похідну від загального визначення якості [21]. Якість стороннього інструменту – це його здатність задовольняти команди розробки та замовників продукту проекту.

Якість інструменту охоплює як технічні, так і організаційно-людські показники якості, що впливають на ефективність його використання у проекті (див. табл. 1 – 3).

Таким чином, якість інструменту визначає його технічну досконалість, зручність експлуатації та зрілість як продукту, що використовується у програмному середовищі проекту.

Якість інструменту j оцінюється як зважена сума його показників якості:

$$Q_j = \sum_{k=1}^{s_j} w_{kj} q_{kj}, \quad (9)$$

де s_j – кількість показників якості, що використовуються для оцінки інструменту j .

w_{kj} – ваговий коефіцієнт для показника якості k інструменту j .

q_{kj} – показник якості k інструменту j .

Третя цільова функція, що відповідає максимізації якості інструменту, має вигляд:

$$\sum_{j=1}^n Q_j x_j \rightarrow \max_{x_j} \quad (10)$$

Ризики – це сукупність невизначеностей і потенційних вразливостей, пов'язаних із використанням інструменту протягом усього його життєвого циклу. Вони охоплюють технічні чинники (рівень безпеки, стійкість до збоїв, стабільність і частота випусків), організаційні аспекти (узгодженість із корпоративною культурою, підтримку з боку розробника чи спільноти, можливість знайти кваліфікованих фахівців, реалістичність термінів впровадження), а також економічні та зовнішні умови (ризик перевищення бюджету, невідповідний профіль грошових потоків, залежність від конкретної країни чи компанії-постачальника, відсутність фінансування з боку клієнта). Таким чином, ризики відображають імовірність того, що інструмент стане джерелом проблем чи обмежень у середньо- та довгостроковій перспективі (див. табл. 4).

Таблиця 4 – Визначення ризиків

Для команди розробника	Для замовника
Припустимо для інструменту j виявлено l_{dev} ризикових подій, $k = 1, l_{dev}$. Наслідки ризикової події k , оцінюються як $o_{dev,k}$. Ймовірність ризикової події k оцінюється як $p_{dev,k}$. Таким чином, ризик для команди розробника дорівнюватиме $R_{dev,j} = \sum_{k=1}^{l_{dev}} o_{dev,k} * p_{dev,k}$ (11)	Припустимо для інструменту j виявлено l_{prod} ризикових подій, $k = 1, l_{dev}$. Наслідки ризикової події k оцінюються як $o_{prod,k}$. Ймовірність ризикової події k оцінюється як $p_{prod,k}$. Таким чином, ризик у замовника дорівнюватиме $R_{prod,j} = \sum_{k=1}^{l_{prod}} o_{prod,k} * p_{prod,k}$ (12)

Четверта цільова функція, що відповідає мінімізації ризиків від впровадження інструменту, має вигляд:

$$\sum_{j=1}^n (R_{dev,j} + R_{prod,j}) x_j \rightarrow \min_{x_j} \quad (13)$$

Технологічний ефект (див. табл. 5) – це стратегічна цінність інструменту, що виходить за межі його локального функціонального застосування. Він включає здатність інструменту забезпечувати

інноваційність, кросплатформеність, сумісність із іншими системами, відповідність архітектурі проекту й довгостроковим цілям розвитку організації, та відповідність трендам технологічного розвитку галузі. До цього напряму належать також план розвитку й стратегічне бачення постачальника, активність підтримки та репутація розробника. Важливими є й «м'які» чинники: популярність, імідж інструменту, його поширеність на ринку та привабливість для спільноти й кінцевих користувачів.

Таблиця 5 – Визначення технологічного ефекту

Для команди розробника	Для замовника
Нехай для інструменту j виявлено $\tau_{dev,j}$ технологічних ефектів, що очікуються після впровадження, $h = 1, \tau_{dev,j}$. $b_{dev,h,j}$ – ваговий коефіцієнт для ефекту h від інструменту j , $\pi_{dev,h,j}$ – оцінка ефекту h від інструменту j . Таким чином, технологічний ефект для команди розробника дорівнюватиме $T_{dev,j} = \sum_{h=1}^{\tau_{dev,j}} b_{dev,h,j} \times \pi_{dev,h,j}$ (14)	Нехай для інструменту j виявлено $\tau_{prod,j}$ технологічних ефектів, що очікуються після впровадження, $h = 1, \tau_{prod,j}$. $b_{prod,h,j}$ – ваговий коефіцієнт для ефекту h від інструменту j , $\pi_{prod,h,j}$ – оцінка ефекту h від інструменту j . Таким чином, технологічний ефект для команди розробника дорівнюватиме $T_{prod,j} = \sum_{h=1}^{\tau_{prod,j}} b_{prod,h,j} \times \pi_{prod,h,j}$ (15)

Сумарний технологічний ефект інструменту визначається як:

$$T_j = T_{dev,j} + T_{prod,j} \quad (16)$$

П'ята цільова функція для вибору інструменту з урахуванням технологічного ефекту має вигляд:

$$\sum_{j=1}^n (T_{dev,j} + T_{prod,j}) x_j \rightarrow \max_{x_j} \quad (17)$$

Для розв'язання задачі вибору інструменту розробки пропонується використати метод багатокритеріального аналізу TOPSIS [6], який забезпечує гнучку процедуру визначення альтернативи, «найближчої до ідеалу», за наперед заданим користувачем набором критеріїв. Після нормалізації вихідних даних метод є критеріально незалежним, тобто не залежить від одиниць вимірювання та шкал оцінювання.

Проте застосування класичного методу TOPSIS дає змогу лише ранжувати альтернативи за сукупністю критеріїв, повністю ігноруючи наявні обмеження. Інакше кажучи, метод передбачає попередній відбір рішень із припустимої області: спочатку кожне рішення перевіряється на допустимість, а вже потім серед допустимих альтернатив визначається оптимальна.

Для усунення зазначеної прогалини доцільно модифікувати базовий підхід, інтегрувавши механізм урахування обмежень безпосередньо у процес розрахунків.

Одним із найгнучкіших способів реалізації такого механізму є використання штрафних функцій. Цей підхід передбачає модифікацію початкової матриці рішень до етапу нормалізації в алгоритмі TOPSIS. Альтернативи, що порушують задані обмеження, отримують штраф до своїх показників, що гарантує: підсумковий вибір буде не лише оптимальним за сукупністю критеріїв, а й припустимим з погляду бізнес-вимог. Запровадження штрафних функцій є

особливо доцільним у ситуаціях, коли незначне порушення обмеження (наприклад, перевищення бюджету на 2%) вважається прийнятним, тоді як істотні відхилення (перевищення на 50%) – ні.

Інтеграція способу полягає в модифікації початкового значення критерію x_{ij} до нового значення:

$$x_{pen,ij} = x_{ij} + P(x_{ij}), \quad (18)$$

де i – сторонній інструмент розробки як альтернатива, $i = \overline{1, m}$;

j – критерій оцінювання, $j \in \{E, C, R, Q, T\}$;

x_{ij} – оцінка значення критерію j для альтернативи i ;

$P(x_{ij})$ – штрафна функція. Для оцінки критерію x_{ij} з обмеженням x_{max} штрафна функція має вигляд:

$$P(x_{ij}) = \begin{cases} 0, & x_{ij} \leq x_{max} \\ k_{x_{ij}} \times (x_{ij} - x_{max}), & x_{ij} > x_{max} \end{cases} \quad (19)$$

де $(x_{ij} - x_{max})$ – абсолютна величина порушення,

$k_{x_{ij}}$ – коефіцієнт штрафу, що визначає наскільки критичним є кожна одиниця порушення. Велике значення коефіцієнта наближає до ситуації жорсткого штрафу.

Потрібно обрати один із трьох альтернативних інструментів розробки A_i для IT-проекту. Вибір ґрунтується на п'яти інтегральних критеріях, визначених у моделі, та обмеженнях, визначених у таблиці 6.

Таблиця 6 – Визначення інтегральних показників

Критерій	Трудомісткість	Витрати	Ризики	Якість	Техн. ефект
Позначення критерію	E	C	R	Q	T
Напрямок оптимізації	min	min	min	max	max
Одиниця вимірювання	людино-годин	у.о.	оцінка 0-10	оцінка 0-10	оцінка 0-10
Обмеження	550	190	-	-	-
Коефіцієнт штрафу	1	10	-	-	-

Згідно з таблицею, значення критеріїв ризиків, якості та технологічного ефекту подаються у вигляді числових оцінок у діапазоні від 0 до 10. Для критеріїв якості та технологічного ефекту: чим нижчим є значення критерію (і чим ближче воно до 0), тим вищою є ймовірність того, що впровадження або заміна стороннього інструменту матиме потенційно негативний вплив на проєкт, тобто така оцінка фактично є негативним індикатором. Для критерію ризики: навпаки, чим нижчим є значення критерію (і чим ближче воно до 0), тим вищою є ймовірність того, що впровадження або заміна стороннього інструменту не матиме негативний вплив на проєкт.

Припустимо, що була проведена оцінка для всіх трьох інструментів за метриками та консолідовано значення метрик. Це дає змогу сформулювати початкову матрицю рішень.

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} E & C & R & Q & T \\ 500 & 150 & 6 & 7 & 8 \\ 350 & 200 & 4 & 9 & 7 \\ 600 & 120 & 8 & 6 & 6 \end{pmatrix}$$

Першим кроком є пошук критеріїв, які порушують обмеження, та застосування для них функції штрафів. Для знайдених критеріїв необхідно провести перерахунок. Наприклад, для альтернативи A_2 критерій C порушує встановлене обмеження. Відповідно перерахунок значення критерію C для альтернативи A_2 буде мати наступний вигляд:

$$P(x_{2C}) = 10 \times (200 - 190) = 100$$

$$x_{pen,2C} = x_{2C} + P(x_{2C}) = 200 + 100 = 300$$

Для альтернативи A_3 критерій E порушує встановлене обмеження. Відповідно перерахунок значення критерію E для альтернативи A_3 буде мати наступний вигляд:

$$P(x_{3E}) = 1 \times (600 - 550) = 50$$

$$x_{pen,3E} = x_{3E} + P(x_{3E}) = 600 + 50 = 650$$

Оновлена матриця рішень після застосування штрафів має вигляд:

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} E & C & R & Q & T \\ 500 & 150 & 6 & 7 & 8 \\ 350 & 300 & 4 & 9 & 7 \\ 650 & 120 & 8 & 6 & 6 \end{pmatrix}$$

Наступний крок передбачає нормалізацію матриці рішень (оцінок метрик) $\frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}$ із використанням

розрахованих коефіцієнтів (див. табл. 7) і перехід до зваженої нормалізованої матриці v_{ij} за допомогою ваг важливості (див. табл. 8).

$$v_{ij} = w_j \times \frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}, \quad (21)$$

де w_j – вага важливості метрики j , $w_j = \overline{0,1}$, $\sum_{j=1}^n w_j = 1$;

v_{ij} – зважена нормалізована оцінка метрики j для альтернативи i .

Таблиця 7 – Розрахований коефіцієнт $\frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}$

	E	C	R	Q	T
$\sqrt{\sum_i x_{ij}^2}$	891.63	356.23	10.77	12.88	12.21

Таблиця 8 – Ваги важливості метрик

	E	C	R	Q	T
w_j	0.15	0.30	0.10	0.35	0.10

Нормалізована матриця $\frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}$ має наступний

вигляд:

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} E & C & R & Q & T \\ 0.561 & 0.421 & 0.557 & 0.543 & 0.655 \\ 0.393 & 0.842 & 0.371 & 0.698 & 0.573 \\ 0.729 & 0.337 & 0.743 & 0.466 & 0.491 \end{pmatrix}$$

Зважена нормалізована матриця v_{ij} має наступний вигляд:

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} E & C & R & Q & T \\ 0.084 & 0.126 & 0.056 & 0.190 & 0.066 \\ 0.059 & 0.253 & 0.037 & 0.244 & 0.057 \\ 0.109 & 0.101 & 0.074 & 0.163 & 0.049 \end{pmatrix}$$

Визначення найкращих альтернатив для вибору v_j^* , при якому ми бажаємо мінімізувати E , C , R та максимізувати Q , T .

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} E & C & R & Q & T \\ 0.084 & 0.126 & 0.056 & 0.190 & \mathbf{0.066} \\ \mathbf{0.059} & 0.253 & \mathbf{0.037} & \mathbf{0.244} & 0.057 \\ 0.109 & \mathbf{0.101} & 0.074 & 0.163 & 0.049 \end{pmatrix}$$

Визначення найгірших альтернатив для вибору v_j' , при якому ми бажаємо максимізувати E , C , R та мінімізувати Q , T .

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} E & C & R & Q & T \\ 0.084 & 0.126 & 0.056 & 0.190 & 0.066 \\ 0.059 & \mathbf{0.253} & 0.037 & 0.244 & 0.057 \\ \mathbf{0.109} & 0.101 & \mathbf{0.074} & \mathbf{0.163} & \mathbf{0.049} \end{pmatrix}$$

Розрахунок відстані кожної альтернативи до ідеального рішення щодо вибору:

$$S_i^+ = \sqrt{\sum_j (v_j^* - v_{ij})^2}, \quad (22)$$

$$S_i^- = \sqrt{\sum_j (v_j' - v_{ij})^2} \quad (23)$$

Відстані кожної альтернативи до ідеального рішення представлені у таблиці 9.

Таблиця 9 – Відстані до ідеального рішення

S_i^+	S_i^-
$A_1 \begin{pmatrix} 0.067 \\ 0.152 \\ 0.103 \end{pmatrix}$	$A_1 \begin{pmatrix} 0.134 \\ 0.102 \\ 0.152 \end{pmatrix}$

Вибір найкращої альтернативи здійснюється як $C_i^* = \frac{S_i^-}{S_i^+ + S_i^-}$, $0 < C_i^* < 1$:

$$\begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \begin{pmatrix} \mathbf{0.666} \\ 0.402 \\ 0.595 \end{pmatrix}$$

Інструмент A_1 є найкращим вибором. Це демонструє, що Інструмент A_1 є найбільш збалансованим та раціональним рішенням у цьому прикладі, оскільки він є єдиною альтернативою, що повністю відповідає встановленим бізнес-обмеженням. Його відносно високі трудомісткість та посередня якість є його головними недоліками, але вони повністю компенсуються відсутністю штрафів за порушення критичних обмежень. Водночас інструменти A_2 та A_3 отримали суттєві штрафи, які значно погіршили їхні підсумкові позиції, незважаючи на їхні переваги за окремими критеріями.

Висновки. У статті розглянуто багатокритеріальну оптимізаційну задачу вибору сторонніх інструментів розробки, для якої попередньо у науковій площині не було розроблено формальних оптимізаційних моделей з чітко визначеними цільовими функціями та обмеженнями.

Було згруповано систематизований набір критеріїв у п'ять інтегральних показників: трудомісткість, витрати, якість, ризики та технологічний ефект. Така класифікація дає змогу пов'язати окремі метрики з ключовими цілями проекту, а також забезпечує можливість подальшої формалізації задачі вибору інструменту як багатокритеріальної. Для кожного інтегрального показника запропоновано кількісне подання, що уніфікує різномірні оцінки в числовому діапазоні та дозволяє виконувати подальшу математичну обробку.

Запропоновано модель вибору стороннього інструменту розробки, у якій показники трудомісткості, витрат, якості, ризиків та технологічного ефекту інтегровані в єдину формалізовану схему прийняття рішень. Модель реалізовано на основі адаптованого методу TOPSIS, доповненого механізмом штрафних функцій для

безпосереднього урахування бізнес-обмежень (зокрема бюджетних, часових та обмежень на ризики чи мінімально припустимий рівень якості). Це дозволяє ранжувати альтернативи за сукупністю критеріїв та враховувати, контролювати порушення критичних обмежень.

На числовому прикладі з чіткими даними та трьома альтернативними інструментами продемонстровано застосування запропонованого підходу. Показано, що найкращою альтернативою стає інструмент, який не є домінуючим за всіма окремими критеріями, але забезпечує збалансований профіль за інтегральними показниками та повністю відповідає заданим бізнес-обмеженням. Інструменти з кращими значеннями окремих показників, але з порушенням обмежень, отримують суттєві штрафи, що змінює їх на нижчі позиції в підсумковому ранжуванні. Це підтверджує спроможність моделі відображати реальні компроміси між ефективністю та допустимістю рішень.

Практичне значення роботи полягає в тому, що запропонована модель та система інтегральних показників можуть бути використані як основа для побудови процедур і програмних засобів підтримки прийняття рішень щодо вибору сторонніх інструментів розробки в ІТ-проектах.

References

1. Larios-Vargas E., Aniche M., Treude C., Bruntink M., Gousios G. Selecting third-party libraries: The practitioners' perspective. *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20)*, 8–13 November 2020, Virtual Event, USA. New York: ACM, 2020. P. 245–256. DOI: 10.1145/3368089.3409711.
2. Tabassum M., Butt W. H., Muzaffar A. W., Lashari S. A. Selecting Third-Party Libraries: The Web Developers' Perspective. *International Journal of Software Engineering and Knowledge Engineering*. 2024. Vol. 34, No. 12. P. 1857–1893. DOI: 10.1142/S0218194024500402.
3. Nadi S., Sakr N. Selecting third-party libraries: the data scientist's perspective. *Empirical Software Engineering*. 2023. Vol. 28. Art. 15. – DOI: 10.1007/s10664-022-10241-3.
4. Nguyen P. T., Di Rocco J., Di Ruscio D., Di Penta M. CrossRec: Supporting software recommendation by recommending third-party libraries. *Journal of Systems and Software*. 2020. Vol. 161. Art. 110460. DOI: 10.1016/j.jss.2019.110460.
5. Jamwal A., Kumar S. A Multi-Scale Hypergraph-Based Approach for Third-Party Library Recommendation in Mobile App Development. *ACM Transactions on Software Engineering and Methodology*. 2025. DOI: 10.1145/3764114.
6. Lysenko A., Kononenko I. How to select third-party library: harnessing visual insights and systematic evaluation for informed decisions. *Radioelectronic and Computer Systems*. 2025. No. 1(113). P. 314–326. DOI: 10.32620/reaks.2025.1.20.
7. Jung H.-W., Choi B. Optimization models for quality and cost of modular software systems. *European Journal of Operational Research*. 1999. Vol. 112, No. 3. P. 613–619. – DOI: 10.1016/S0377-2217(98)00169-6.
8. Zachariah B., Rattihalli R. N. A multicriteria optimization model for quality of modular software systems. *Asia-Pacific Journal of Operational Research*. 2007. Vol. 24, No. 6. P. 797–811. DOI: 10.1142/S0217595907001541.
9. Kwong C. K., Mu L. F., Tang J. F., Luo X. G. Optimization of software components selection for component-based software system development. *Computers and Industrial Engineering*. 2010. Vol. 58, No. 4. P. 618–624. DOI: 10.1016/j.cie.2010.01.003.
10. Tang J. F., Mu L. F., Kwong C. K., Luo X. G. An optimization model for software component selection under multiple applications development. *European Journal of Operational Research*. 2011. Vol. 212, No. 2. P. 301–311. DOI: 10.1016/j.ejor.2011.01.045.
11. Rafsanjani M. K., Rakhshan N. Using a 0/1 Knapsack Algorithm for Software Components Selection in Component-based Software System Development. *Journal of American Science*. 2011. Vol. 7, No. 12. P. 641–648.
12. Verma S., Mehlawat M. K., Mahajan D. Software component evaluation and selection using TOPSIS and fuzzy interactive approach under multiple applications development. *Annals of Operations Research*. 2022. Vol. 312, No. 1. P. 441–471. DOI: 10.1007/s10479-018-3022-9.
13. Nabot A. Software Component Selection: An Optimized Selection Criterion for Component-based Software Engineering (CBSE). *The International Arab Journal of Information Technology*. 2024. Vol. 21, No. 2. P. 211–225. DOI: 10.34028/iajit/21/2/4.
14. Mahmoudi A., Feylizadeh M. R. A mathematical model for crashing projects by considering time, cost, quality and risk. *Journal of Project Management*. 2017. Vol. 2, No. 1. P. 27–36. DOI: 10.5267/j.jpmp.2017.5.002.
15. Van Wyngaard C. J., Pretorius J. H. C., Pretorius L. Theory of the triple constraint – a conceptual review. *Proceedings of the 2012 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2012)*, 10–13 December 2012, Hong Kong, China. Piscataway, NJ: IEEE, 2012. P. 1991–1997. DOI: 10.1109/IEEM.2012.6838095.
16. Kononenko I. V., Fadeyev V. A., Kolisnyk M. E. Project scope optimization model and method on criteria profit, time, cost, quality, risk. *Integrating Project Management Standards: Proceedings of the 26th IPMA World Congress*, 29–31 October 2012, Conference Centre Creta Maris, Hersonissos, Crete, Greece. 2012. P. 286–292.
17. Boehm B. W. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
18. *Open Textbooks for Hong Kong. Definitions of quality* [Electronic resource]. SCI S319 Quality Management For Science And Technology (Free Courseware). – 2016. – Available at: <https://www.opentextbooks.org.hk/ditatopic/10606> (accessed: 14 November 2025).
19. *ISO 9000:2015 Quality management systems – Fundamentals and vocabulary*. Geneva: International Organization for Standardization, 2015.
20. Marle F., Gidel T. A multi-criteria decision-making process for project risk management method selection. *International Journal of Multicriteria Decision Making*. 2012. Vol. 2, No. 2. P. 189–223.
21. Kozłowska J. Methods of multi-criteria analysis in technology selection and technology assessment: a systematic literature review. *Engineering Management in Production and Services*. 2022. Vol. 14, No. 2. P. 116–137. DOI: 10.2478/emj-2022-0021.

Надійшла (received) 05.11.2025

Відомості про авторів / About the Authors

Лисенко Олександр Олександрович (Lysenko Alexander) – аспірант кафедри управління проектами в інформаційних технологіях, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: alexander.lysenko.dev@gmail.com; ORCID: <https://orcid.org/0009-0009-4962-5881>.

Кононенко Ігор Володимирович (Kononenko Igor) – доктор технічних наук, професор, професор кафедри управління проектами в інформаційних технологіях, Національний технічний університет "Харківський політехнічний інститут", м. Харків, Україна; запрошений професор, Азербайджанський університет архітектури та будівництва, м. Баку, Азербайджанська Республіка, e-mail: igorvkononenko@gmail.com; ORCID: <https://orcid.org/0000-0002-1218-2791>.