DOI: https://doi.org/10.36910/6775-2524-0560-2025-58-24
УДК 004.41:519.876.5
Васьків Роман Ігорович, аспірант
https://orcid.org/0000-0002-8549-5035
Веретеннікова Наталія Вячеславівна, к.н.соц.к., доцент
https://orcid.org/0000-0001-9564-4084
Національний університет «Львівська політехніка», м Львів, Україна

МЕТОДОЛОГІЯ «ІНЖЕНЕРІЯ ХАОСУ» ТА СТІЙКІСТЬ РОЗПОДІЛЕНИХ ІТ-КОМАНД

Васьків Р.І., Веретеннікова Н.В. Методологія «Інженерія хаосу» та стійкість розподілених ІТ-команд. У статті досліджено застосування принципів методології «Інженерія хаосу» для підвищення стійкості розподілених ІТ-команд шляхом вдосконалення управління ризиками та оптимізації проєктних процесів. Розроблено математичну модель, яка дозволяє кількісно оцінювати ключові параметри, зокрема часову затримку, комунікаційні бар'єри, стабільність інформаційних систем, функціональну розподіленість, ефективність проксі-ролей та рівень змінності команди. Модель передбачає дискретний аналіз у часових інтервалах, що дає змогу оцінювати динаміку змін і вплив ризиків на продуктивність розподілених команд. Особлива увага приділена методології моделювання збоїв, яка включає імітацію критичних сценаріїв для тестування стійкості системи в умовах високої невизначеності. Запропоновані теоретичні та практичні підходи сприяють вдосконаленню методів управління ризиками в умовах сучасних динамічних ІТ-середовищ.
Ключові слова: методологія «Інженерія хаосу», управління ризиками, розподілені ІТ-команди, комунікаційні бар'єри, стабільність систем, проксі-ролі, оптимізація проєктних процесів.

Vaskiv R., Veretennikova N. Chaos Engineering methodology and the resilience of distributed IT teams. The paper explores the application of Chaos Engineering methodology to enhance the resilience of distributed IT teams through improved risk management and project process optimization. A mathematical model has been developed to quantitatively assess key parameters, including latency, communication barriers, information system stability, functional distribution, proxy role efficiency, and team variability. The model employs discrete interval analysis, enabling the evaluation of dynamic changes and the impact of risks on the performance of distributed teams. Special attention is given to failure modeling methodology, which incorporates the simulation of critical scenarios to test system resilience under high uncertainty conditions. The proposed theoretical and practical approaches contribute to advancing risk management methods in contemporary dynamic IT environments.
Keywords: Chaos engineering methodology, risk management, distributed IT teams, communication barriers, system stability, proxy roles, project process optimization.

Постановка наукової проблеми. У сучасному трансформаційному середовищі розподілені ІТ-команди набувають дедалі більшого значення, забезпечуючи доступ до кращих талантів, оптимізацію витрат та гнучкість у реалізації проєктів. За даними Баффер, після пандемії COVID-19 понад 70% компаній впровадили гібридну або повністю віддалену модель роботи, причому в ІТ-секторі ця тенденція проявляється особливо яскраво [1]. Дослідження Гартнера показує, що технологічні компанії є лідерами у впровадженні розподілених команд та віддалених форматів роботи [2]. Водночас такі команди стикаються з унікальними викликами, зокрема соціокультурними, часовими, географічними, технічними та функціональними бар'єрами, які можуть негативно впливати на їхню продуктивність і стійкість. Важливою складовою успіху розподілених команд є здатність забезпечувати ефективне використання інформаційних та комунікаційних технологій, які відіграють ключову роль у взаємодії учасників та координації проєктних процесів.

Ризики, з якими стикаються розподілені команди, мають багатовимірний характер. Географічна розподіленість команди може спричиняти затримки у прийнятті рішень через різницю в часових поясах, що обмежує можливість синхронної взаємодії. Соціокультурні відмінності — різниця у ментальності, мовах, традиціях і робочих підходах — можуть викликати непорозуміння, напругу та зниження рівня взаємодії. За даними Інституту проєктного менеджменту (Project Management Institute), до 40% проєктів у розподілених командах зазнають значних затримок саме через комунікаційні бар'єри [3].

Функціональна розподіленість є ще одним важливим викликом. У межах таких команд кожен учасник має чітко визначені ролі, зони відповідальності та завдання, які можуть виконуватись паралельно або послідовно. Це підвищує ефективність роботи, але водночас створює складнощі в координації, узгодженні пріоритетів і запобіганні конфліктам між функціональними підрозділами. Невдале управління функціональною розподіленістю може призводити до дублювання зусиль, або помилок через недостатню комунікацію та значного зниження продуктивності команди.

Окрім цього, залежність від стабільності інформаційних систем та надійності комунікаційних технологій залишається критичним аспектом. Збої в цих системах можуть посилювати негативні

наслідки розподіленої роботи, спричиняючи зриви у виконанні проєктних завдань. Технічні збої та неефективна комунікація можуть суттєво впливати на продуктивність розподілених команд, спричиняючи затримки та перевищення бюджетів. Згідно з дослідженням Ш. Фукс, Дж. Новікке, Г. Струбе, великі проєкти в різних галузях зазвичай завершуються на 20% пізніше запланованого терміну та з перевищенням бюджету на 80% [4]. Крім того, за даними Інституту проєктного менеджменту, неефективна комунікація є причиною невдачі проєктів у третині випадків [5].

Важливою складовою цих викликів є управління ризиками, які невід'ємно супроводжують проєкти в умовах високої невизначеності та складності. Ризик в управлінні IT-проєктами визначається як невизначена подія або умова, яка, якщо виникне, матиме позитивний або негативний вплив на принаймні одну з цілей проєкту, таких як час, вартість, якість або обсяг робіт [6].

Індентифікація ризиків та їхніх чинників вважається ключовим етапом управління ризиками [7, 8], який широко використовують як у гнучких, так і в традиційних підходах до розробки програмного забезпечення [9]. Фактор ризику визначається як умова, яка може становити серйозну загрозу для успішного завершення проєкту з розробки програмного забезпечення [10]. Неефективність процесу ідентифікації ризиків при розробці складних систем вважається однією з основних причин невдач проєктів [11]. Низка досліджень вказує на те, що фактично існує прямий зв'язок між управлінням ризиками та успіхом або покращенням ефективності проєктів з розробки програмного забезпечення [12, 13, 14, 15, 16, 17]. Ці дослідження показують, що фактори ризику повинні бути ідентифіковані та добре контрольовані для того, щоб проєкти досягли своїх цілей. Таким чином, ідентифікація факторів ризику відіграє вирішальну роль в успіху та ефективності проєктів з розробки програмного забезпечення.

Посібник із зведення знань з управління проєктами (PMBOK Guide) підкреслює важливість проактивного підходу до управління ризиками, що включає ідентифікацію, аналіз, планування заходів реагування та моніторинг ризиків протягом життєвого циклу проєкту [6]. Характерною рисою ризиків в IT-проєктах є їхній ефект взаємозалежності, коли один ризик може спричинити каскад інших.

Методологія «Інженерія хаосу» (Chaos Engineering), як дисципліна, спрямована на тестування систем шляхом моделювання реальних збоїв, відкриває нові можливості для підвищення стійкості розподілених команд. Завдяки цілеспрямованим експериментам, які дозволяють імітувати непередбачувані сценарії, можна ідентифікувати вузькі місця у використанні інформаційних систем і комунікаційних технологій, а також оцінити здатність команди до адаптації в умовах невизначеності. Методологія «Інженерія хаосу» дозволяє не лише виявити потенційні проблеми, але й оцінити ефективність розроблених рішень, створюючи основу для формування стійких і надійних проєктних процесів навіть за умов географічної, часової та функціональної розподіленості.

Рис.1 – Етапи створення експерименту в методології «Інженерія хаосу» для тестування стійкості розподілених систем

Побудова експерименту в межах методології «Інженерія хаосу» (рис. 1) є структурованим процесом, спрямованим на моделювання потенційних збоїв у системах для перевірки їхньої стійкості. Даний процес включає такі етапи:

1. Створення нового експерименту (гіпотези). Ініціація експерименту полягає у визначенні основної мети дослідження та ключових параметрів для перевірки.
2. Вибір збою з бібліотеки по методології «Інженерія хаосу». З бібліотеки можливих збоїв обирається той, який найбільше відповідає цілям експерименту. Це можуть бути збої в комунікації, перевантаження системи тощо.
3. Налаштування параметрів збою. На цьому етапі задаються ключові характеристики збою, такі як тривалість, серйозність і інші властивості. Це дозволяє точно змоделювати умови, наближені до реальних.
4. Додавання інших збоїв до того самого об'єкта. Для підвищення точності експерименту можна додати кілька збоїв для аналізу їхнього впливу на одну цільову систему або процес.
5. Додавання перевірки стійкості системи. Перевірка стійкості дозволяє оцінити, чи здатна система повернутися до стабільного стану після інциденту. Це критичний етап для виявлення потенційних слабких місць.
6. Додавання нових збоїв до експерименту. Після виконання первинних етапів можна поступово включати інші сценарії збоїв, щоб дослідити комплексні сценарії їхнього впливу.
7. Завершення експерименту. Після завершення всіх етапів аналізуються отримані дані, робляться висновки щодо виявлених вразливостей і ефективності методів усунення наслідків збоїв.

Ця методологія дозволяє забезпечити системний підхід до оцінки стійкості розподілених систем, використовуючи принципи «Інженерії хаосу», та оптимізувати роботу проєктних процесів у розподілених командах.

Метою цього дослідження є інтеграція принципів методології «Інженерія хаосу» із сучасними методами управління проєктами для забезпечення стійкості розподілених команд у сфері інформаційних систем та технологій. Особлива увага виділена аналізу ризиків, пов'язаних із соціокультурними бар'єрами, технічними обмеженнями, часовою, географічною та функціональною розподіленістю, а також складністю взаємодії у рамках комунікаційних платформ. Методологія «Інженерія хаосу» використовується для моделювання сценаріїв, які дозволяють оцінити ефективність інформаційних систем і технологій у критичних умовах, а також оптимізувати проєктні процеси, що проходять у розподіленій команді. Завдяки цьому підходу стає можливо не лише ідентифікувати вузькі місця в роботі команди, але й розробити стратегії для мінімізації ризиків, пов'язаних із затримками в комунікації, невідповідністю інструментів, некоректним розподілом ролей та іншими критичними аспектами. Моделювання потенційних збоїв у процесах дозволяє тестувати стійкість командних взаємодій, виявляти слабкі місця в управлінні проєктами та створювати підхід до їхньої оптимізації, що сприяє досягненню високої продуктивності навіть у складних умовах розподіленої роботи.

Дослідження зосереджене на розробці моделей управління ризиками, які враховують особливості роботи розподілених команд, та впровадженні практичних підходів до оптимізації проєктних процесів. В статті запропоновано теоретичні основи та практичні рекомендації, що можуть стати основою для майбутніх досліджень і вдосконаленні практик управління розподіленими командами в галузі ІТ.

Аналіз досліджень. Методологія «Інженерія хаосу» зародилася як відповідь на зростаючу складність розподілених систем, зокрема в контексті хмарних обчислень. Цей підхід, вперше застосований у компанії Netflix у 2010 році, передбачає створення контрольованих експериментів для виявлення слабких місць у системах до виникнення реальних збоїв. Застосування інструменту Chaos Monkey дозволило Netflix суттєво підвищити надійність своїх послуг у динамічному середовищі. Основна ідея полягала у створенні сценаріїв, які моделюють реальні відмови, щоб протестувати стійкість систем і водночас підготувати команди до вирішення потенційних проблем.

Методологія «Інженерія хаосу» є концептуально значущим підходом для аналізу та покращення стійкості складних систем, особливо у контексті сучасних розподілених ІТ-команд. Як зазначають Віллард та Гатсон, цей підхід базується на моделюванні контрольованих збоїв у реальних середовищах з метою виявлення слабких місць до того, як вони призведуть до критичних проблем. Основна ідея полягає у створенні сценаріїв, які дозволяють перевірити, як система реагує на відмови та чи здатна вона відновлюватися без зовнішнього втручання [18].

Однією з ключових переваг методології «Інженерія хаосу» є можливість виявлення прихованих вразливостей. Наприклад, за даними Польтроньєрі, інтеграція цифрових двійників (Chaostwin) у експерименти з методологією «Інженерія хаосу» дозволяє тестувати стійкість систем

у віртуальних середовищах, що мінімізує ризики для реальних інфраструктур [19]. Це відкриває нові можливості для аналізу надійності ІТ-процесів, включно із соціальною та технічною взаємодією розподілених команд.

Інструменти у методології «Інженерія хаосу», такі як Chaos Monkey, не лише імітують можливі збої, але й дають змогу створювати складні сценарії відмов. Наприклад, використання підходу «Fail-Fast, Fail-Small» дозволяє ізолювати точки відмови та поступово знижувати функціональність, зберігаючи критично важливі сервіси [18]. У контексті розподілених систем цей підхід підтримує адаптивність, необхідну для мінімізації впливу збоїв.

У кіберфізичних системах (Cyber Physical System - CPS) методологія «Інженерія хаосу» застосовується для оцінки стійкості до таких загроз, як природні катастрофи, техногенні аварії, кібератаки, мережеві збої, а також помилки користувачів. За словами Костандіну, такі експерименти дозволяють оптимізувати взаємодію між апаратними та програмними компонентами, забезпечуючи працездатність критичних функцій навіть у складних умовах [20]. Дана особливість може бути адаптована для ІТ-команд, які працюють у багатозадачних і розподілених середовищах.

Інтеграція штучного інтелекту (ШІ) у методологію «Інженерія хаосу» також розширює можливості цієї методології. Наприклад, Чжу, Хау та ін. стверджують, що алгоритми машинного навчання можуть прогнозувати потенційні збої, аналізуючи великі набори даних у режимі реального часу. Це дозволяє організаціям переходити від реактивного до проактивного управління ризиками, тим самим підвищуючи загальну стійкість систем [21].

Ще однією перспективою методології «Інженерія хаосу» є її використання для оптимізації гібридних хмарних середовищ. Як зазначає Діаманті, поєднання методології «Інженерія хаосу» із штучним інтелектом дозволяє автоматизувати процеси управління ресурсами, забезпечуючи їх динамічний розподіл залежно від поточних умов. Це допомагає уникати перевантажень систем і гарантує стабільність навіть під час пікових навантажень [22].

Методологія «Інженерія хаосу» демонструє значний потенціал для адаптації та оптимізації процесів в ІТ-командах. Її інтеграція з інструментами ШІ, цифровими двійниками та гібридними хмарними платформами дозволяє забезпечувати стійкість та адаптивність сучасних розподілених систем. Завдяки цьому підходу команди отримують ефективні засоби для виявлення ризиків і розробки стратегій їхнього усунення, що підвищує продуктивність та стабільність проєктних процесів.

Методологія «Інженерія хаосу» базується на формуванні гіпотез про очікувану поведінку системи у стабільному стані, моделюванні реальних сценаріїв збоїв і детальному аналізі впливу таких сценаріїв на ключові показники ефективності. Особлива увага виділяється проведенню експериментів у продуктивних середовищах, що дає змогу тестувати не лише технічну, а й організаційну стійкість. Це забезпечує можливість адаптувати методологію до умов, у яких працюють сучасні розподілені команди.

Розподілені ІТ-команди стали невід'ємною частиною сучасних проєктів, але їхня структура породжує низку викликів. Географічна розподіленість, яка дозволяє залучати фахівців із різних регіонів, ускладнює синхронізацію через різницю в часових поясах. Це створює бар'єри для оперативного прийняття рішень та ефективної координації. Водночас функціональна розподіленість, коли кожен учасник має чітко визначені ролі, дозволяє виконувати завдання паралельно, але водночас створює ризики розривів у комунікації між різними функціональними групами. Така структура потребує особливого підходу до управління, який враховує ці особливості.

Методологія «Інженерія хаосу» дозволяє моделювати сценарії, що враховують географічну, часову та функціональну розподіленість команд, і аналізувати, як ці фактори впливають на ефективність проєктних процесів. Наприклад, імітація затримок у комунікації або збоїв у координації між функціональними групами дає змогу визначити слабкі місця у взаємодії команди й підготувати рекомендації щодо їх усунення.

Розглянемо детальніше поняття ризиків та їх класифікації. Міжнародна організація зі стандартизації [23], визначає ризик, як «вплив невизначеності на досягнення цілей». У свою чергу, Інститут управління проєктами [6] визначає його як «невизначену подію або стан, які, у разі настання, можуть мати позитивний або негативний вплив на одну чи кілька цілей проєкту». Згідно підходу Хілсона [24], ризик це як «невизначеність, що має значення». Це визначення акцентує увагу на тому, що ризиком вважаються лише ті невизначеності, які безпосередньо впливають на цілі проєкту. Дослідники Елмс та Франк [25, 26] пропонують класифікацію невизначеностей на два основні типи: алеаторні, які характеризуються варіативністю і широким діапазоном можливих

значень, та епістемічні, що виникають через неоднозначність або брак знань. Хілсон [24] деталізує ці категорії, виокремлюючи чотири види невизначеностей: алеаторні, що пов'язані з надійністю процесів; стохастичні, які є потенційними подіями ризику; епістемічні, що зумовлені браком інформації; і онтологічні, які є невідомими («чорні лебеді»).

Варто зазначити, що онтологічна невизначеність залишається поза можливістю моделювання через повну відсутність інформації про ризик. Натомість інші види невизначеності інтегруються у модель проєкту, де ймовірність та вплив кожного типу моделюються у вигляді функцій розподілу. Це дозволяє їх врахувати під час проведення симуляцій методом Монте-Карло. Симуляція на основі методу Монте-Карло, замінює матрицю ймовірність-вплив. Методологія забезпечує кількісну оцінку ризиків, дозволяючи визначати їхній вплив на тривалість та вартість проєкту [27].

Управління ризиками передбачає ідентифікацію можливостей і загроз, здатних вплинути на цілі проєкту, а також розробку відповідних заходів реагування. Головна мета цього процесу полягає у збільшенні шансів реалізації можливостей та зменшенні ймовірності прояву визначених загроз.

Управління ризиками в розподілених командах вимагає адаптивних стратегій. Одним із ключових підходів є використання гнучких методологій, таких як Scrum чи Kanban. Ці методології забезпечують регулярну комунікацію, короткі цикли зворотного зв'язку та можливість швидкої адаптації до змін. Водночас моделювання ризиків у контексті методології «Інженерія хаосу» дозволяє не лише тестувати технічну стійкість, а й аналізувати вплив організаційних факторів, таких як взаємодія між учасниками, ефективність використання комунікаційних інструментів і адаптація до змін.

Сучасні інформаційні системи та комунікаційні платформи відіграють ключову роль у забезпеченні функціонування розподілених команд. Інструменти управління проєктами, такі як Jira чи Trello, сприяють координації завдань і забезпечують прозорість процесів, тоді як платформи для комунікації, як-от Slack чи Microsoft Teams, підтримують ефективний обмін інформацією. Проте використання цих інструментів не позбавлене викликів. Затримки в мережевих з'єднаннях, перевантаження серверів або проблеми з інтеграцією різних систем можуть впливати на ефективність роботи. Методологія «Інженерія хаосу» забезпечує можливість моделювати ці сценарії, тестувати стійкість платформ і розробляти заходи для усунення виявлених недоліків.

Отже, комплексний аналіз літератури підтверджує значний потенціал методології «Інженерія хаосу» як інструменту для підвищення стійкості розподілених ІТ-команд. Інтеграція цього підходу з сучасними методами управління ризиками та оптимізації проєктних процесів відкриває нові можливості для ефективної роботи команд у сучасному динамічному середовищі.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Методологія «Інженерія хаосу» створює можливість моделювання збоїв у реальних умовах для вивчення реакції системи та команд, що забезпечує глибоке розуміння слабких місць. У розподілених ІТ-командах, де процеси є багатошаровими і залежними від комунікаційних та інформаційних систем, застосування цього підходу може виявити потенційні ризики та сприяти підвищенню стійкості. З огляду на складність таких систем, запропонуємо розширений набір гіпотез.

Таблиця 1

Гіпотези для дослідження стійкості розподілених ІТ-команд з використанням принципів методології «Інженерія хаосу»

Категорія гіпотез	Формулювання гіпотез
Стійкість розподілених команд	Впровадження принципів методології «Інженерія хаосу» дозволяє виявити вузькі місця у функціональній, часовій та комунікаційній складових розподіленої команди.
	Розподіл ролей у команді за функціональною ознакою підвищує ефективність виконання завдань та мінімізує втрати через дублювання функцій.
	Використання локальних проксі-ролей, таких як локальний Власник продукту (Product Owner), знижує час на прийняття рішень в умовах географічної розподіленості.
	Динамічна адаптація до змін у проєктах, включаючи зміну чи оновлення складу команди, дозволяє зменшити ризики затримок.

	Інтеграція принципів методології «Інженерія хаосу» із Scrum допомагає оптимізувати короткострокові завдання та підвищити продуктивність команди.
	Рівень змінності команди та її вплив на проєктні ризики вимагає врахування додаткових сценаріїв, зокрема кадрових ротацій та їх наслідків.
	Функціональна розподіленість команд напряму впливає на розподіл завдань і зниження ризиків дублювання функцій.
Комунікаційні бар'єри	Збільшення часових зон між учасниками команди пропорційно підвищує ризик затримок у виконанні завдань.
	Використання автоматизованих комунікаційних інструментів (Slack, Microsoft Teams) зменшує вплив мовних та часових бар'єрів.
	Імітація збоїв у комунікаційних технологіях допомагає визначити критичні залежності між командами.
	Планування синхронних зустрічей у спільних часових слотах знижує ризики неправильного розуміння завдань.
	Скорочення кількості комунікаційних вузлів між учасниками команди зменшує ризик втрати важливої інформації.
Ефективність інформаційних систем	Стійкість проєкту залежить від інтеграції інформаційних систем, які використовуються у розподіленій команді, із загальними платформами управління проєктами.
	Впровадження методології «Інженерія хаосу» дозволяє оцінити реакцію інформаційних систем на раптові збої та їх вплив на проєктні процеси.
	Частота технічних збоїв у системах управління завданнями пропорційна втратам у продуктивності команди.
	Автоматизація звітності у системах управління проєктами мінімізує ризики, пов'язані з людським фактором.
	Імітація надмірного навантаження на інформаційні системи дозволяє оцінити їхню продуктивність у критичних умовах.
Управління змінами в командах	Планування та управління виходами учасників команди (через відпустки, хвороби, звільнення) знижує ризик затримок у виконанні завдань.
	Імітація сценаріїв несподіваного виходу ключового учасника проєкту дозволяє оцінити стійкість команди до кадрових змін.
	Оптимізація завантаження учасників команди сприяє стабільності проєкту у довгостроковій перспективі.
Функціональний розподіл та ризики	Чітке визначення зон відповідальності у команді мінімізує ризики конфліктів та дублювання роботи.
	Імітація затримок виконання функціонально залежних завдань дозволяє ідентифікувати критичні вузли проєктного процесу.

Кожна з запропонованих гіпотез дозволяє оцінити окремий аспект роботи розподілених команд та виявити ключові ризики, пов'язані із взаємодією учасників, комунікаційними технологіями, інформаційними системами та проєктними процесами.

У розподілених IT-командах управління ризиками є ключовим завданням для забезпечення стабільності та продуктивності проєктних процесів. Зважаючи на географічну, функціональну та часову розподіленість, а також використання різних інформаційних систем і комунікаційних технологій, виникає потреба у кількісному аналізі впливу цих факторів. Для цього розроблено систему параметрів, яка дозволяє оцінити ризики та ефективність їхнього управління.

Описані параметри враховують ключові аспекти функціонування розподілених команд, включаючи комунікаційні та інформаційні бар'єри, час реагування на завдання, стабільність системи, ефективність використання проксі-ролей і рівень виконання завдань. Наведені формули забезпечують основу для математичного аналізу цих параметрів, що дозволяє систематизувати підхід до управління ризиками в динамічних умовах.

Далі наведено опис кожного параметра, його тлумачення та математичну інтерпретацію.

Часова затримка (TL_t) є одним із ключових параметрів, що відображає ефективність комунікації між учасниками розподіленої команди. Вона визначається часом, необхідним для отримання відповіді на запит, і залежить від часових зон, швидкості реакції та доступності учасників.

$$TL_t = \frac{\sum_{i=1}^{n} t_{response,i} - t_{request,i}}{n}$$

де $t_{response,i}$ — час отримання відповіді на i-й запит, $t_{request,i}$ — час відправлення запиту, n — кількість запитів.

Комунікаційні бар'єри (CB_t) включають соціокультурні, мовні та технічні перешкоди, які можуть впливати на взаєморозуміння між членами команди. Вони є важливим фактором ризику, оскільки спричиняють затримки або нерозв'язаність завдань.

$$CB_t = \frac{N_{unresolved}}{N_{total}}$$

де $N_{unresolved}$ — кількість завдань, що залишилися нерозв'язаними через комунікаційні труднощі, N_{total} — загальна кількість завдань у вибірці.

Інформаційні бар'єри (IB_t) виникають через несумісність інформаційних систем або їхню низьку продуктивність, що призводить до затримок у виконанні запитів.

$$IB_t = \frac{\sum_{i=1}^{m} t_{execution,i}}{m}$$

де $t_{execution,i}$ — час виконання i-го запиту в інформаційній системі, m — кількість запитів.

Стабільність системи (S_t) є ключовим показником, що відображає частоту збоїв у роботі інформаційних систем і вплив цих збоїв на виконання завдань. Вона оцінює ймовірність успішного завершення завдань без порушень.

$$S_t = \frac{N_{failures}}{N_{total}}$$

де $N_{failures}$ — кількість збоїв за визначений період, N_{total} — загальна кількість завдань.

Ефективність проксі-ролей (PR_t) оцінює здатність локальних представників, таких як проксі-Власник продукту (Product Owner) вирішувати завдання без залучення головного Власника продукту (Product Owner). Це дозволяє мінімізувати затримки у прийнятті рішень.

$$PR_t = \frac{N_{proxy\_resolved}}{N_{total\_tasks}}$$

де $N_{proxy\_resolved}$ — кількість завдань, вирішених проксі-ролями, N_{total_tasks} — загальна кількість завдань.

Цей показник (C_t) є індикативним параметром, що дозволяє кількісно оцінити рівень досягнення командою поставлених цілей протягом визначеного періоду часу.

Це співвідношення завершених завдань до запланованих.

$$C_t = \frac{N_{completed}}{N_{planned}}$$

де $N_{completed}$ — кількість завершених завдань, $N_{planned}$ — кількість запланованих завдань. Цей параметр виконує допоміжну функцію в системі управління ризиками, оскільки його значення опосередковано формується на основі таких ключових показників моделі, як стабільність системи(S_t), часова затримка (TL_t) та ефективність проксі-ролей (PR_t). Таким чином, (C_t) використовується для загальної оцінки продуктивності команди, проте не є безпосередньо інтегрованим у математичну модель, зважаючи на його вторинний характер і залежність від інших базових параметрів.

Рівень змінності команди (HR_t) – параметр оцінює частоту виходу учасників із проєкту через відпустки, хвороби чи інші причини, що впливає на стабільність проєкту.

$$HR_t = \frac{N_{exits}}{T}$$

де N_{exits} — кількість виходів учасників за період, T — тривалість періоду (у днях).

Функціональна розподіленість (FD_t) параметр характеризує взаємодоповнюваність функціональних ролей у команді, що впливає на ефективність розподілу завдань.

$$FD_t = \frac{\sum_{i=1}^{k} R_{efficiency,i}}{k}$$

де $R_{efficiency,i}$ — ефективність виконання завдань i-ї ролі, k — кількість функціональних ролей у команді.

Перераховані вище параметри формують основу для кількісної оцінки ефективності управління ризиками в розподілених командах. Їх використання дозволяє моделювати та прогнозувати вплив ризиків, пов'язаних із комунікацією, інформаційними системами та організаційними змінами.

Розподілені IT-команди функціонують у динамічному середовищі, що постійно піддається впливу внутрішніх і зовнішніх змін. Ефективне управління ризиками в таких командах вимагає врахування динамічних факторів, таких як часові, функціональні, комунікаційні бар'єри, зміни в складі команди, а також вплив технічних збоїв. У цьому контексті використання принципів методології «Інженерія хаосу» стає ключовим елементом для моделювання сценаріїв, які допомагають ідентифікувати та мінімізувати ризики у супроводі проєктних процесів.

Робота розподілених команд в умовах динамічного середовища потребує інтеграції методів методології «Інженерія хаосу» із математичними моделями для оцінки ризиків. Основним завданням є врахування змінних параметрів системи у дискретні моменти часу, що дозволяє точніше відобразити динаміку процесів. Застосування дискретних часових індексів t_j дає змогу точно відображати стан системи на визначених етапах її функціонування, що забезпечує детальний аналіз динаміки змін та оцінку ефективності управління ризиками в межах кожного окремого інтервалу часу. Модель для оцінки ефективності управління ризиками в розподілених IT-командах із інтеграцією принципів методології «Інженерія хаосу» має наступний вигляд (фор. 1). Вона забезпечує аналіз параметрів системи у дискретних часових моментах t_j, враховуючи динамічний характер процесів у команді та реакцію на збої, моделюванні у межах методології «Інженерія хаосу».

Загальний рівень ефективності управління ризиками R_t розраховується як середньозважене значення параметрів у дискретні моменти часу:

$$R(t) = \frac{\sum_{j=1}^{n}\left(\omega_{TL}\cdot TL(t_j) + \omega_{CB}\cdot CB(t_j) + \omega_{IB}\cdot IB(t_j) + \omega_S\cdot S(t_j) + \omega_{PR}\cdot PR(t_j) + \omega_{HR}\cdot HR(t_j) + \omega_{FD}\cdot FD(t_j)\right)\cdot \Delta t_j}{\sum_{j=1}^{n} \Delta t_j} \quad (1)$$

де:

- t_j – часові індекси, які позначають стан системи в момент часу j;
- $\Delta t_j = t_j - t_{j-1}$ – тривалість інтервалу між двома послідовними моментами часу;
- ω_{TL} – ваговий коефіцієнт комунікаційної ефективності, що відображає значення параметрів часу затримки (TL) у загальній моделі (коефіцієнт затримки відповіді);
- ω_{CB} – ваговий коефіцієнт комунікаційних бар'єрів, що враховує, наскільки критичними є комунікаційні бар'єри (CB) для роботи команди (коефіцієнт комунікаційних бар'єрів);
- ω_{IB} – ваговий коефіцієнт інформаційної ефективності, що відображає вплив інформаційних бар'єрів (IB) на продуктивність (коефіцієнт стабільності інформаційних систем);
- ω_S – ваговий коефіцієнт стійкості системи, що показує значення стабільності (S) для загальної оцінки (коефіцієнт стійкості системи);
- ω_{PR} – ваговий коефіцієнт ефективності проксі-ролей, що враховує вплив проксі-ролей (PR) на управління ризиками (коефіцієнт проксі-ефективності).
- ω_{HR} – ваговий коефіцієнт рівня змінності команди, що враховує частоту виходів учасників із проєкту (HR) та їх вплив на стабільність процесів (коефіцієнт змінності);
- ω_{FD} – ваговий коефіцієнт функціональної розподіленості, що оцінює узгодженість функціональних ролей у команді (FD) та вплив цього показника на ефективність виконання завдань (коефіцієнт функціональної розподіленості).
- C_t – показник виконання завдань, який є додатковим індикатором продуктивності команди.

Значення вагових коефіцієнтів мають бути встановлені залежно від особливостей конкретного проєкту, його вимог та ризиків.

Основні методи визначення.

Експертна оцінка. У цьому підході залучаються експерти з управління проєктами, які на основі свого досвіду визначають пріоритетність параметрів. Значення коефіцієнтів залежать від контексту проєкту.

Наприклад:

- для проєктів із високою залежністю від комунікацій: $\omega_{TL} = 0.4$, $\omega_{CB} = 0.3$.
- для проєктів із складною IT-інфраструктурою: $\omega_{IB} = 0.5$, $\omega_S = 0.3$.

Аналіз історичних даних. Цей підхід базується на аналізі попередніх проєктів для визначення значущості параметрів. Якщо історичні дані показують, що комунікаційні бар'єри є частою причиною затримок, коефіцієнт ω_{CB} збільшується. Аналогічно, якщо стабільність інформаційних систем відіграє ключову роль у проєктах, підвищується ω_{IB}. В проєктах із високою змінністю учасників значення ω_{HR} отримує пріоритет. Додатково, у проєктах із низьким рівнем завершення завдань (C_t) можуть підвищуватися вагові коефіцієнти ω_s або ω_{FD}.

Метод аналізу ієрархій. Метод передбачає побудову матриці парного порівняння параметрів за важливістю. На основі цих порівнянь розраховується нормалізована шкала ваг. Наприклад:

- Якщо затримка відповіді (TL) важливіша за інформаційні бар'єри (IB), то коефіцієнт $\omega_{TL} > \omega_{IB}$. Результати матриці дозволяють отримати обґрунтовані вагові коефіцієнти.
- Якщо функціональна розподіленість (FD) є критичною для виконання завдань, ω_{FD} має вищий пріоритет, ніж інші коефіцієнти.

Динамічний підхід. У цьому підході ваги змінюються залежно від стадії проєкту та актуальних ризиків.

Наприклад:

- на початкових етапах проєкту значення ω_{PR} (ефективність проксі-ролей) може бути низьким, але зростає в міру збільшення залежності від локальних ролей;
- у проєктах із високою змінністю команди значення ω_{HR} зростає на завершальних етапах для врахування ризиків кадрових змін;
- у проєктах із швидкими змінами значущість стабільності систем ω_S зростає у критичних фазах.
- у проєктах із низькими показниками завершення завдань (C_t) можуть переглядатися вагові коефіцієнти для стабільності (ω_S) та комунікаційної ефективності (ω_{TL}).

Для полегшення вибору значень вагових коефіцієнтів, пропонується орієнтовна таблиця 2.

Таблиця 2

Рекомендовані значення вагових коефіцієнтів для моделі оцінки ефективності

Коефіцієнт	Оптимальне значення	Ризикове значення	Критичне значення
ω_{TL} – коефіцієнт затримки відповіді	0.2	0.4	0.6
ω_{CB} – коефіцієнт комунікаційних бар'єрів	0.3	0.5	0.7
ω_{IB} – коефіцієнт стабільності інформаційних систем	0.4	0.6	0.8
ω_S – коефіцієнт стійкості системи	0.3	0.5	0.7
ω_{PR} – коефіцієнт проксі-ефективності	0.3	0.4	0.6
ω_{HR} – коефіцієнт змінності команди	0.2	0.3	0.5
ω_{FD} – коефіцієнт функціональної розподіленості	0.2	0.3	0.5

Методологія «Інженерія хаосу» створює умови для динамічної адаптації вагових коефіцієнтів моделі управління ризиками шляхом аналізу чутливості ключових параметрів до змодельованих збоїв. У дискретний момент часу t_j моделюється комунікаційний збій, після чого виконуються кількісні оцінки параметрів $TL(t_j)$, $CB(t_j)$, $IB(t_j)$ та інших параметрів моделі, таких як $HR(t_j)$, $FD(t_j)$, до і після інциденту. Результати такого аналізу слугують основою для корекції вагових

коефіцієнтів залежно від їх впливу на загальну стійкість системи. Наприклад, якщо результати експерименту демонструють, що стабільність системи S є домінантним фактором, ваговий коефіцієнт ω_S збільшується. Водночас, якщо ефективність комунікації критично залежить від часових затримок, ваговий коефіцієнт ω_{TL} підвищується пропорційно їх впливу. Таким чином, вагові коефіцієнти не лише відображають специфіку конкретного проєкту, а й забезпечують можливість гнучкої адаптації моделі до реальних умов, підвищуючи ефективність і релевантність управління ризиками в умовах розподілених IT-систем.

Запропонована модель дозволяє оцінювати ефективність управління ризиками в розподілених IT-командах на основі параметрів, що відображають різні аспекти роботи системи. Ключовими компонентами є часова затримка ($(TL(t_j))$), комунікаційні бар'єри ($CB(t_j)$), інформаційні бар'єри ($IB(t_j)$), стабільність системи ($S(t_j)$), рівень змінності команди $\left(HR(t_j)\right)$, функціональна розподіленість $\left(FD(t_j)\right)$ та ефективність проксі-ролей ($PR(t_j)$), що оцінюються у дискретні моменти часу (t_j).

Часова затримка ($(TL(t_j))$). Параметр оцінює ефективність комунікації між членами команди, визначаючи середній час відгуку на запити:

$$TL(t_j) = \frac{1}{n}\sum_{i=1}^{n}\left(t_{response,i} - t_{request,i}\right),$$

де $t_{response,i}$ — час відповіді на i-й запит, $t_{request,i}$ — час надсилання запиту, n — кількість запитів у часовому інтервалі.

Комунікаційні бар'єри ($(CB(t_j))$). Цей параметр враховує вплив мовних, культурних або технологічних бар'єрів на ефективність виконання завдань:

$$CB(t_j) = \frac{N_{unresolved}}{N_{total}},$$

де $N_{unresolved}$ — кількість завдань, які залишилися невирішеними через бар'єри, а N_{total} — загальна кількість завдань.

Інформаційні бар'єри ($(IB(t_j))$). Параметр оцінює затримки або несумісність інформаційних систем:

$$IB(t_j) = \frac{1}{m}\sum_{i=1}^{m} t_{execution,i},$$

де $t_{execution,i}$ — час виконання i-го запиту, m — кількість запитів у вибраному інтервалі.

Стабільність системи ($(S(t_j))$). Цей показник відображає частку успішно завершених завдань:

$$S(t_j) = \frac{N_{success}}{N_{total}},$$

де $N_{success}$ — кількість успішно завершених завдань, N_{total} — загальна кількість завдань.

Ефективність проксі-ролей ($(PR(t_j))$). Цей параметр характеризує здатність локальних ролей, наприклад проксі-Власник продукту (Product Owner), вирішувати завдання без участі основних виконавців:

$$PR(t_j) = \frac{N_{proxy\_resolved}}{N_{total\_tasks}},$$

де $N_{proxy\_resolved}$ — кількість завдань, вирішених проксі-ролями, N_{total_tasks} — загальна кількість завдань.

Для забезпечення систематичного підходу до оцінки ефективності управління ризиками в розподілених IT-командах було розроблено схему алгоритму, яка ілюструє ключові етапи аналізу та прийняття рішень при використанні моделі згідно методології «Інженеріїя хаосу». Схема алгоритму є інструментом для візуалізації аналізу параметрів, їх перевірки та розробки оптимізаційних стратегій (рис. 2).

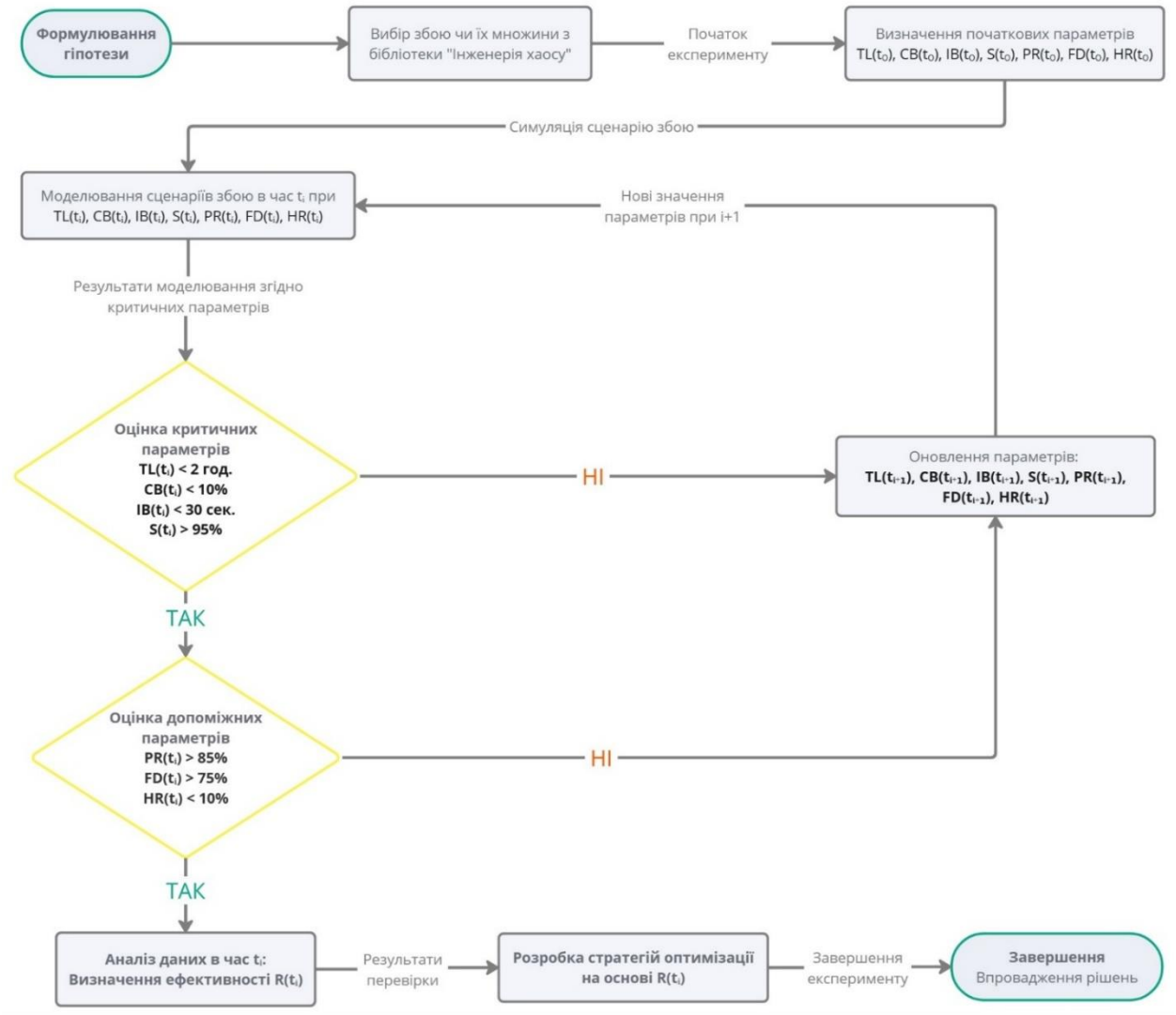

Рис. 2 – Алгоритм оцінки та оптимізації ефективності управління ризиками в розподілених IT-командах

Подана схема алгоритму (рис. 2) відображає етапи моделювання та перевірки параметрів у рамках методології управління ризиками розподілених IT-команд. Алгоритм базується на поетапній оцінці ключових параметрів моделі, з подальшим оновленням їх значень у відповідності до отриманих результатів моделювання.

На першому етапі здійснюється оцінка критичних параметрів — часу затримки (TL), комунікаційних бар'єрів (CB)), інформаційних бар'єрів (IB) і стабільності системи (S). Ці параметри є фундаментальними для оцінки загального стану системи, оскільки їх відхилення можуть суттєво впливати на ефективність командної роботи. Критерії оцінки визначені на основі порогових значень, представлених у таблиці 3.

Таблиця 3

Критерії оцінки критичних параметрів моделі

Параметр	Оптимальне значення	Допустиме значення	Критичне значення
$TL(t)$	< 2 год.	2–4 год.	> 4 год.
$CB(t)$	< 10%	10–20%	> 20%
$IB(t)$	< 30 сек.	30–60 сек.	> 60 сек.
$S(t)$	> 95%	80–95%	< 80%

У разі, якщо значення критичних параметрів виходять за межі допустимих рівнів, здійснюється корекція моделі із внесенням змін до параметрів і повторною перевіркою.

На другому етапі оцінюються допоміжні параметри, зокрема ефективність проксі-ролей (PR), функціональна розподіленість (FD) і рівень змінності команди (HR). Ці параметри є важливими для

деталізації оцінки та вдосконалення моделі, оскільки вони відображають гнучкість і стійкість командних процесів. Їх порогові значення наведено в таблиці 4.

Таблиця 4

Критерії оцінки допоміжних параметрів моделі

Параметр	Оптимальне значення	Допустиме значення	Критичне значення
$PR(t)$	> 85%	70–85%	< 70%
$FD(t)$	> 75%	50–75%	< 50%
$HR(t)$	< 10%	10–20%	> 20%

Результати аналізу параметрів використовуються для розробки оптимізаційних стратегій, що враховують виявлені ризики та потенційні покращення. Інтеграція моделі з принципами методології «Інженерія хаосу» дозволяє підвищити її адаптивність до динамічних умов роботи розподілених команд.

Вибір вагових коефіцієнтів у моделі оцінки ефективності управління ризиками в розподілених IT-командах залежить від специфіки проєкту, пріоритетності ризиків і еталонних значень параметрів. Розглянемо основні аспекти формування вагових коефіцієнтів.

Пріоритетність вагових коефіцієнтів формується відповідно до критичних аспектів проєкту.

До прикладу у проєктах із високою залежністю від синхронної комунікації між командами ваговий коефіцієнт ω_{CB} може отримати найвищий пріоритет. Це пов'язано з необхідністю оперативного вирішення задач, де комунікаційні бар'єри є ключовим фактором ризику. Проте у технічно складних проєктах, де критичною є стабільність інформаційних систем, домінуючим стає коефіцієнт ω_{IB}, що відображає вплив інформаційних бар'єрів на продуктивність. Для проєктів із частими змінами складу команди або складною функціональною структурою, пріоритет можуть отримати коефіцієнти ω_{HR} та ω_{FD}, які відображають змінність команди та її функціональну узгодженість.

Еталонні значення допомагають встановити допустимі та критичні межі для кожного параметра.

Для параметра часова затримка $TL(t_j)$:

- **Оптимальний рівень**: $TL(t_j) < 5$ хвилин. Такий показник досяжний для задач, які виконуються в синхронному режимі або в межах близьких часових поясів.
- **Допустимий рівень**: 5 хвилин $\leq TL(t_j) \leq 2$ години. Цей рівень враховує взаємодію між учасниками, які працюють у різних часових зонах, і допускає час на аналіз запиту.
- **Ризиковий рівень**: $TL(t_j) > 2$ години. Затримка понад цей час вказує на суттєві комунікаційні чи організаційні проблеми, які можуть впливати на продуктивність і синхронізацію команд.

Параметр $TL(t_j)$ визначає середній час між моментом надсилання запиту та отриманням відповіді. Він враховує кілька важливих аспектів, які суттєво впливають на ефективність взаємодії в розподілених командах. Першим аспектом є технічна складова, яка охоплює затримки в передачі даних через інформаційні системи або мережеву інфраструктуру. Наприклад, затримки можуть виникати через перевантаження серверів або недостатню швидкість мережевого з'єднання.

Другим ключовим аспектом є час реакції людини. Цей період включає час, необхідний учаснику команди для обробки запиту. Він залежить від поточного робочого навантаження, рівня доступності, а також складності завдання. Додатково враховується вплив контексту, у якому працює команда, зокрема можливість одночасної участі в інших процесах.

Третім аспектом є організаційні чинники. Вони включають затримки, що виникають через необхідність уточнення деталей запиту, ескалацію до відповідальної особи або перенаправлення завдання між різними учасниками команди. Такі затримки можуть бути спричинені недостатньо чіткими регламентами або складністю структури команди.

У розподілених командах, де учасники можуть перебувати в різних часових поясах і залежати від складних інформаційних систем, параметр $TL(t_j)$ забезпечує комплексну оцінку оперативності взаємодії. Він дозволяє визначити потенційні проблеми у комунікації, які можуть уповільнити прийняття рішень та знизити продуктивність проєкту. Цей показник є критичним для розподілених команд, де часові пояси й мережеві залежності створюють додаткові бар'єри для оперативної взаємодії.

Для параметра комунікаційні бар'єри $CB(t_j)$:

- Допустимий рівень: до 10% завдань із бар'єрами.
- Ризиковий рівень: понад 20% завдань, що свідчить про необхідність корекції процесів.

Для стабільності системи $S(t_j)$:

- Оптимальний рівень: понад 95% успішно виконаних завдань.
- Ризиковий рівень: нижче 80%, що свідчить про значні технічні чи організаційні ризики.

Для ефективності проксі-ролей $PR(t_j)$:

- Оптимальний рівень: понад 85%, що відображає високу автономність локальних команд.
- Ризиковий рівень: нижче 70%, що може свідчити про затримки в прийнятті рішень.

Дискретна модель оцінки ефективності управління ризиками має низку важливих переваг, які підкреслюють її наукову та практичну цінність. Однією з ключових особливостей є її точність, що забезпечується можливістю аналізу впливу окремих інцидентів у визначені часових інтервалів. Це дозволяє глибше оцінити динаміку змін у системі та ідентифікувати критичні вузькі місця.

Ще однією суттєвою перевагою моделі є її гнучкість, яка проявляється у здатності адаптуватися до різних масштабів і стадій проєкту. Це дозволяє враховувати специфіку не лише короткострокових завдань, але й довгострокових проєктів із високою кадровою змінністю або функціональною складністю.

Окрім того, модель вирізняється високою практичністю. Її простота в реалізації сприяє ефективному використанню в реальних умовах, особливо для аналізу великих обсягів даних. Ця властивість робить її зручною для інтеграції в існуючі системи управління ризиками, забезпечуючи високу оперативність у прийнятті рішень.

Проте, незважаючи на переваги дискретної моделі оцінки ефективності управління ризиками, існує потреба у подальшій оптимізації підходів, які могли б не лише кількісно оцінювати ризики, але й забезпечувати практичну перевірку стійкості систем у реальних умовах. У цьому контексті в майбутніх дослідженнях буде розглянуто інтегрування методів кількісного аналізу, таких як Monte Carlo Simulation (MCS), із практичними експериментами, що пропонує методологія «Інженерія хаосу».

Висновки. Запропонована в межах дослідження модель управління ризиками для розподілених IT-команд, інтегрована з принципами методології «Інженерія хаосу», надає можливість цілеспрямовано виявляти критичні ризики, оптимізувати проєктні процеси та підвищувати загальну стійкість команд. Її математичне підґрунтя охоплює низку ключових параметрів, специфічних для розподілених середовищ: часова затримка $TL(t)$, комунікаційні бар'єри $CB(t)$, інформаційні бар'єри $IB(t)$, стабільність системи $S(t)$, ефективність проксі-ролей $PR(t)$, функціональна розподіленість $FD(t)$ та рівень змінності команди $HR(t)$. Для інтегральної оцінки цих показників використовується загальний рівень ефективності управління ризиками $R(t)$, що зважує значущість кожного параметра та дозволяє відстежувати динаміку змін стійкості команди впродовж усього життєвого циклу проєкту.

Важливо, що модель передбачає динамічну адаптацію вагових коефіцієнтів, які визначають внесок окремих параметрів, залежно від результатів експериментів у межах методології «Інженерія хаосу» (Chaos Engineering). Це сприяє оперативному виявленню «вузьких місць» під час виконання завдань та підвищує ефективність керування проєктами. Крім того, модель не лише виявляє теоретичні загрози, а й практично перевіряє стійкість системи в продуктивному режимі: різнопланові імітації збоїв дають змогу комплексно оцінити вплив відмов як на технічні, так і на організаційні аспекти.

Незважаючи на очевидні переваги, подальші дослідження є доцільними для більш комплексного охоплення ймовірнісних сценаріїв, зокрема «каскадних» або багатофакторних збоїв. У цьому контексті інтегрування методу Monte Carlo Simulation (MCS) зі сценаріями, сформованими на засадах підходу методології «Інженерія хаосу», розглядається як перспективний напрям. Такий синергетичний підхід забезпечить імовірнісний прогноз впливу потенційних ризик-подій на тривалість і вартість проєктів, а також на динаміку робочих процесів у розподілених IT-командах. Як наслідок, удосконалена модель сприятиме гнучкішому коригуванню підходів до управління проєктними процесами з урахуванням специфіки розподілених команд, адже поєднує імовірнісну оцінку ризиків із практичною перевіркою стійкості у межах методології «Інженерія хаосу». Водночас, подальше розширення бібліотеки сценаріїв цієї методології дасть змогу імітувати ще ширший спектр нетипових (аномальних) подій, у тому числі зі складною каскадною природою, що

збагатить можливості кількісного аналізу ризиків та посилить готовність розподілених IT-команд до непередбачуваних збоїв. Така інтеграція підвищує прогнозованість і ефективність роботи розподілених проєктних команд, водночас формуючи надійне підґрунтя для продуктивної діяльності в умовах високої невизначеності сучасних IT-середовищ.

Список бібліографічного опису

1. Buffer (2022). State of Remote Work 2022: The Annual Buffer Report. Buffer Publishing. https://buffer.com/state-of-remote-work/2022
2. Gartner (2022). Future of Work Trends Post-COVID-19: Long-Term Impact & Actions for HR. *Gartner Research*. https://www.gartner.com/en/human-resources/trends/future-of-work-trends-post-covid-19
3. Project Management Institute (2021). Pulse of the Profession: Beyond Agility. PMI Publishing.
4. Steffen Fuchs, James Nowicke, Gernot Strube (2017). Navigating the digital future: The disruption of capital projects. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/navigating-the-digital-future-the-disruption-of-capital-projects
5. Project Management Institute (PMI) (2021). PMI study reveals poor communication leads to project failure one-third of the time. Ascertra Blog. https://www.ascertra.com/blog/pmi-study-reveals-poor-communication-leads-to-project-failure-one-third-of-the-time
6. Project Management Institute (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition*.
7. De Bakker, K., Boonstra, A., Wortmann, H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management*, 28(5), 493–503. https://doi.org/10.1016/j.ijproman.2009.07.002
8. Salmeron, J. L., Lopez, C. (2012). Forecasting risk impact on ERP maintenance with augmented fuzzy cognitive maps. *IEEE Transactions on Software Engineering*, 38(2), 439–452. https://doi.org/10.1109/TSE.2011.8
9. Neves, S. M., da Silva, C. E. S., Salomon, V. A. P., da Silva, A. F., Sotomonte, B. E. P. (2014). Risk management in software projects through knowledge management techniques: cases in Brazilian incubated technology-based firms. *International Journal of Project Management*, 32(1), 125–138. https://doi.org/10.1016/j.ijproman.2013.02.007
10. March, J. G., Shapira, Z. (1987). Managerial perspectives on risk and risk taking. Management Science, 33(11), 1404–1418. https://doi.org/10.1287/mnsc.33.11.1404
11. Reeves, J. D., Eveleigh, T., Holzer, T. H., Sarkani, S. (2013). Risk identification biases and their impact to space system development project performance. *Engineering Management Journal*, 25(2), 3–12. https://doi.org/10.1080/10429247.2013.11431970
12. Jiang, J., Klein, G. (2000). Software development risks to project effectiveness. *The Journal of Systems and Software*, 52(1), 3–10. https://doi.org/10.1016/S0164-1212(99)00128-4
13. Jiang, J., Klein, G., & Discenza, R. (2001). Information systems success as impacted by risks and development strategies. *IEEE Transactions on Engineering Management*, 48(1), 46–55. https://doi.org/10.1109/17.913165
14. Raz, T., Shenhar, A. J., Dvir, D. (2002). Risk management, project success, and technological uncertainty. *R&D Management*, 32(2), 101–109. https://doi.org/10.1111/1467-9310.00243
15. Wallace, L., Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM*, 47(4), 68–73. https://doi.org/10.1145/975817.9758
16. Wallace, L., Keil, M., Rai, A. (2004). Understanding software project risk: a cluster analysis. *Information Management*, 42(1), 115–125. https://doi.org/10.1016/j.im.2003.12.007
17. Han, W. M., Huang, S. J. (2007). An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software*, 80(1), 42–50. https://doi.org/10.1016/j.jss.2006.04.030
18. Jill Willard, James Hutson (2024). Fail Fast, Fail Small: Designing Resilient Systems for the Future of Software Engineering. *International Journal of Recent Engineering Science*, 11 (5), 51-58. https://doi.org/10.14445/23497157/IJRES-V11I5P106
19. Filippo Poltronieri, Mauro Tortonesi, Cesare Stefanelli (2021). ChaosTwin: A Chaos Engineering and Digital Twin Approach for The Design of Resilient IT Services. *2021 17th International Conference on Network and Service Management (CNSM)*, 234-238. https://doi.org/10.23919/CNSM52442.2021.9615519
20. C. Konstantinou, G. Stergiopoulos, M. Parvania, P. Esteves-Verissimo (2021). Chaos Engineering for Enhanced Resilience of Cyber-Physical Systems. *2021 Resilience Week (RWS)*, 1-10. https://doi.org/10.1109/RWS52686.2021.9611797
21. Yongqing Zhu, Kiam Cheng How, Horng Jyh Wu, Qi Cao (2023). AI-based Proactive Storage Failure Management in Software-Defined Data Centres. In Proceedings of the 2023 6th International Conference on Information Science and Systems (ICISS '23), 231–237. https://doi.org/10.1145/3625156.3625190
22. Alessio Diamanti, José Manuel Sánchez Vílchez, Stefano Secci (2022). An AI-Empowered Framework for Cross-Layer Softwarized Infrastructure State Assessment. IEEE Transactions on Network and Service Management, 19 (4), 4434-4448. https://doi.org/10.1109/TNSM.2022.3161872
23. ISO. (2018). ISO 31000:2018. Risk management – Guidelines. International Organization for Standardization.
24. Hillson D (2014). How to manage the risks you didn't know you were taking. PMI® Global Congress 2014—North America, Phoenix, AZ. Newtown Square, PA: Project Management Institute.
25. Elms D.G. (2004) Structural safety: Issues and progress. Progress in Structural Engineering and Materials, 6 (2), 116–126. https://doi.org/10.1002/pse.176
26. Frank M. (1999) Treatment of uncertainties in space nuclear risk assessment with examples from Cassini mission implications. Reliability Engineering & System Safety, 66 (3), 203–221. https://doi.org/10.1016/S0951-8320(99)00002-2

27. Randy Heffernan (2021). Monte Carlo Simulation Provides Insights to Manage Risks. Risk Management Magazine. https://www.rmmagazine.com/articles/article//2021/05/28/monte-carlo-simulation-provides-insights-to-manage-risks

References

1. Buffer (2022). State of Remote Work 2022: The Annual Buffer Report. Buffer Publishing. https://buffer.com/state-of-remote-work/2022
2. Gartner (2022). Future of Work Trends Post-COVID-19: Long-Term Impact & Actions for HR. *Gartner Research*. https://www.gartner.com/en/human-resources/trends/future-of-work-trends-post-covid-19
3. Project Management Institute (2021). Pulse of the Profession: Beyond Agility. PMI Publishing.
4. Steffen Fuchs, James Nowicke, Gernot Strube (2017). Navigating the digital future: The disruption of capital projects. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/navigating-the-digital-future-the-disruption-of-capital-projects
5. Project Management Institute (PMI) (2021). PMI study reveals poor communication leads to project failure one-third of the time. Ascertra Blog. https://www.ascertra.com/blog/pmi-study-reveals-poor-communication-leads-to-project-failure-one-third-of-the-time
6. Project Management Institute (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition*.
7. De Bakker, K., Boonstra, A., Wortmann, H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management*, 28(5), 493–503. https://doi.org/10.1016/j.ijproman.2009.07.002
8. Salmeron, J. L., Lopez, C. (2012). Forecasting risk impact on ERP maintenance with augmented fuzzy cognitive maps. *IEEE Transactions on Software Engineering*, 38(2), 439–452. https://doi.org/10.1109/TSE.2011.8
9. Neves, S. M., da Silva, C. E. S., Salomon, V. A. P., da Silva, A. F., Sotomonte, B. E. P. (2014). Risk management in software projects through knowledge management techniques: cases in Brazilian incubated technology-based firms. *International Journal of Project Management*, 32(1), 125–138. https://doi.org/10.1016/j.ijproman.2013.02.007
10. March, J. G., Shapira, Z. (1987). Managerial perspectives on risk and risk taking. Management Science, 33(11), 1404–1418. https://doi.org/10.1287/mnsc.33.11.1404
11. Reeves, J. D., Eveleigh, T., Holzer, T. H., Sarkani, S. (2013). Risk identification biases and their impact to space system development project performance. *Engineering Management Journal*, 25(2), 3–12. https://doi.org/10.1080/10429247.2013.11431970
12. Jiang, J., Klein, G. (2000). Software development risks to project effectiveness. *The Journal of Systems and Software*, 52(1), 3–10. https://doi.org/10.1016/S0164-1212(99)00128-4
13. Jiang, J., Klein, G., & Discenza, R. (2001). Information systems success as impacted by risks and development strategies. *IEEE Transactions on Engineering Management*, 48(1), 46–55. https://doi.org/10.1109/17.913165
14. Raz, T., Shenhar, A. J., Dvir, D. (2002). Risk management, project success, and technological uncertainty. *R&D Management*, 32(2), 101–109. https://doi.org/10.1111/1467-9310.00243
15. Wallace, L., Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM*, 47(4), 68–73. https://doi.org/10.1145/975817.9758
16. Wallace, L., Keil, M., Rai, A. (2004). Understanding software project risk: a cluster analysis. *Information Management*, 42(1), 115–125. https://doi.org/10.1016/j.im.2003.12.007
17. Han, W. M., Huang, S. J. (2007). An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software*, 80(1), 42–50. https://doi.org/10.1016/j.jss.2006.04.030
18. Jill Willard, James Hutson (2024). Fail Fast, Fail Small: Designing Resilient Systems for the Future of Software Engineering. *International Journal of Recent Engineering Science*, 11 (5), 51-58. https://doi.org/10.14445/23497157/IJRES-V11I5P106
19. Filippo Poltronieri, Mauro Tortonesi, Cesare Stefanelli (2021). ChaosTwin: A Chaos Engineering and Digital Twin Approach for The Design of Resilient IT Services. *2021 17th International Conference on Network and Service Management (CNSM)*, 234-238. https://doi.org/10.23919/CNSM52442.2021.9615519
20. C. Konstantinou, G. Stergiopoulos, M. Parvania, P. Esteves-Verissimo (2021). Chaos Engineering for Enhanced Resilience of Cyber-Physical Systems. *2021 Resilience Week (RWS)*, 1-10. https://doi.org/10.1109/RWS52686.2021.9611797
21. Yongqing Zhu, Kiam Cheng How, Horng Jyh Wu, Qi Cao (2023). AI-based Proactive Storage Failure Management in Software-Defined Data Centres. In Proceedings of the 2023 6th International Conference on Information Science and Systems (ICISS '23), 231–237. https://doi.org/10.1145/3625156.3625190
22. Alessio Diamanti, José Manuel Sánchez Vílchez, Stefano Secci (2022). An AI-Empowered Framework for Cross-Layer Softwarized Infrastructure State Assessment. IEEE Transactions on Network and Service Management, 19 (4), 4434-4448. https://doi.org/10.1109/TNSM.2022.3161872
23. ISO. (2018). ISO 31000:2018. Risk management – Guidelines. International Organization for Standardization.
24. Hillson D (2014). How to manage the risks you didn't know you were taking. PMI® Global Congress 2014—North America, Phoenix, AZ. Newtown Square, PA: Project Management Institute.
25. Elms D.G. (2004) Structural safety: Issues and progress. Progress in Structural Engineering and Materials, 6 (2), 116–126. https://doi.org/10.1002/pse.176
26. Frank M. (1999) Treatment of uncertainties in space nuclear risk assessment with examples from Cassini mission implications. Reliability Engineering & System Safety, 66 (3), 203–221. https://doi.org/10.1016/S0951-8320(99)00002-2
27. Randy Heffernan (2021). Monte Carlo Simulation Provides Insights to Manage Risks. Risk Management Magazine. https://www.rmmagazine.com/articles/article//2021/05/28/monte-carlo-simulation-provides-insights-to-manage-risks