ANNA KOVALOVA

ШТУЧНИЙ ІНТЕЛЕКТ У ТЕСТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

НАВЧАЛЬНО-МЕТОДИЧНИЙ ПОСІБНИК

МЕТОДИКА ЗАСТОСУВАННЯ З ПРАВИЛАМИ ПЕРЕВІРКИ РЕЗУЛЬТАТІВ ТА КРИТЕРІЯМИ ОЦІНЮВАННЯ

2025

ШТУЧНИЙ ІНТЕЛЕКТ У ТЕСТУВАННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Навчально-методичний посібник

Методика застосування з правилами перевірки результатів та критеріями оцінювання

Видавничий дім «Інтернаука»
2025

УДК 004.41:004.05
DOI:

Автор:
Анна Ковальова
співвласниця та генеральна директорка Anbosoft LLC, Irvine, CA, USA, 92618

Рецензенти:
Розломій Інна, PhD, доцент кафедри інформаційної безпеки та комп’ютерної інженерії, Черкаський державний технологічний університет.

Наконечна Юлія, PhD, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського".

Штучний інтелект у тестуванні програмного забезпечення: навчально-методичний посібник: методика застосування з правилами перевірки результатів та критеріями оцінювання / Анна Ковальова. - Київ: Видавничий дім «Інтернаука», 2025. - 41 с.

ISBN

У посібнику подано практичну методику застосування штучного інтелекту у тестуванні програмного забезпечення з акцентом на контроль якості результатів, правила їх перевірки та вимірювані показники ефективності. Запропоновано стандартизовані сценарії використання, контрольні списки перевірки, критерії оцінювання та практикум з оцінювання зрілості процесу забезпечення якості на основі доказів. Посібник призначений для студентів, викладачів і практиків у сфері тестування та забезпечення якості.

UDC 004.41:004.05
DOI:

Author:
Anna Kovalova
Co-owner and CEO, Anbosoft LLC, Irvine, CA, USA, 92618

Reviewers:

Inna Rozlomii, PhD, Associate Professor, Department of Information Security and Computer Engineering, Cherkasy State Technological University

Yuliia Nakonechna, PhD, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Artificial intelligence in software testing: study guide: application methodology with result verification rules and assessment criteria / Anna Kovalova. - Kyiv: Internauka, 2025. - 41 p.

ISBN

The guide presents a practical methodology for applying artificial intelligence in software testing with a focus on result quality control, verification rules, and measurable performance indicators. It includes standardized usage scenarios, verification checklists, assessment criteria, and a capstone practicum for evidence-based QA process maturity evaluation. The guide is intended for students, instructors, and practitioners in software testing and quality assurance.

Зміст

ДОДАТКИ

Вступ

Мета посібника, сфера застосування, результати навчання
Метою посібника є надати практичну методику використання штучного інтелекту у тестуванні програмного забезпечення з фокусом на керованість, перевірюваність результатів і вимірюваний ефект, а також підтримати навчання на програмах підвищення кваліфікації для фахівців-практиків у сфері тестування та забезпечення якості. Посібник може застосовуватися у навчальних дисциплінах з тестування програмного забезпечення, забезпечення якості, програмної інженерії та суміжних курсах, а також як основа для практичних занять і підсумкового практикуму.

Цільова аудиторія посібника включає студентів бакалаврату 3-4 курсів і магістратури за спеціальностями, пов'язаними з програмною інженерією, комп'ютерними науками, кібербезпекою та інформаційними технологіями, а також слухачів курсів підвищення кваліфікації, корпоративного навчання або професійних програм перекваліфікації у напрямі тестування програмного забезпечення.

Очікувані результати навчання: після опрацювання матеріалу студент або слухач зможе формувати коректні задачі для штучного інтелекту, отримувати результат у структурованому вигляді, перевіряти його за контрольним списком, підтверджувати доказами та оцінювати якість через метрики. Також здобувач зможе підготувати набір тестів для типових сценаріїв, виконати аналіз дефектів і сформувати пропозиції щодо покращення процесу забезпечення якості.

Новизна методики та відмінності від типових підходів
Новизна запропонованої методики полягає у поєднанні практичних сценаріїв застосування штучного інтелекту з чіткою системою перевірки результатів і оцінюванням якості виконаної роботи. На відміну від типових підходів, які зводяться до переліку можливостей або прикладів запитів, у посібнику запропоновано керований процес роботи: запит - відповідь - перевірка - фіксація. Методика подана у вигляді відтворюваної системи правил і шаблонів, що дозволяє отримувати порівнювані результати у різних навчальних групах та проєктах.

Ядром методики є матриця "Сценарій - Ризик - Контроль - Метрика", яка дозволяє заздалегідь визначити ризики використання штучного інтелекту, обрати контрольні дії для перевірки та встановити метрики для оцінювання ефекту. Додатково методика включає правила перевірки результатів, контрольний список та критерії приймання або відхилення результату. Така структура робить підхід повторюваним, придатним для навчального процесу і впровадження у різних командах.

Практичне значення: показники ефекту та сфери застосування
Практичне значення посібника полягає у тому, що він пропонує готову до використання систему навчальних матеріалів, таблиць, контрольних списків і критеріїв оцінювання, які можна інтегрувати у навчальну програму без додаткової розробки. Матеріали посібника можуть бути безпосередньо включені до робочих програм дисциплін і програм підвищення

кваліфікації, а стандартизовані критерії оцінювання забезпечують однаковий підхід до перевірки результатів у різних групах та закладах. Матеріали орієнтовані на задачі, які студенти і практики виконують у реальних проєктах: аналіз вимог, формування критеріїв приймання, тестовий дизайн, тестування програмних інтерфейсів, підтримка автоматизації та аналіз дефектів.

Ефект від застосування методики оцінюється через вимірювані показники, зокрема: скорочення часу на підготовку тестових сценаріїв і документації, підвищення повноти покриття вимог тестами, зменшення кількості пропущених критичних сценаріїв, покращення якості опису дефектів, зниження частки нестабільних тестів в автоматизації, а також підвищення якості звітів про ризики. Посібник може використовуватися як основа для практичних занять, лабораторних робіт, курсових проєктів, програм підвищення кваліфікації та підсумкового практикуму з оцінювання зрілості процесу забезпечення якості на основі доказів.

Оцінювання ефекту виконується шляхом порівняння базових значень показників із результатами після застосування методики, а кожний висновок підтверджується зафіксованими доказами відповідно до правил розділу 4. Такий підхід забезпечує відтворюваність результатів і придатність методики до масштабування в різних командах та навчальних середовищах.

1. Основи застосування штучного інтелекту у тестуванні

Розділ окреслює базові принципи застосування штучного інтелекту у тестуванні програмного забезпечення, щоб забезпечити коректне використання інструментів, уникнути типової помилкової довіри до результатів і сформувати єдині правила роботи для навчальних і практичних задач. У межах цього посібника штучний інтелект розглядається як інструмент підтримки рішень, а не джерело істини. Будь-який результат, сформований із застосуванням штучного інтелекту, потребує перевірки, підтвердження доказами та оцінювання за критеріями приймання.

1.1 Можливості та межі застосування у задачах тестування та забезпечення якості

Штучний інтелект може суттєво підвищувати продуктивність у задачах тестування, особливо там, де потрібні швидке структурування інформації, генерація варіантів або попередній аналіз. Найбільш ефективні напрями застосування:

- Аналіз вимог і критеріїв приймання
  Штучний інтелект допомагає структурувати опис функціональності, виділяти правила, формувати уточнювальні питання, виявляти неоднозначності та прогалини. Корисним результатом є чернетка критеріїв приймання у перевірюваному форматі, яка надалі підтверджується першоджерелами і тестами.

- Генерація тестового дизайну
  Штучний інтелект може пропонувати варіанти позитивних, негативних і граничних сценаріїв, підказувати комбінації даних, допомагати з покриттям критичних правил. Це особливо корисно для стартового набору тестів або для перевірки повноти покриття.

- Тестування програмних інтерфейсів
  Штучний інтелект може формувати структуру наборів перевірок для інтерфейсів, пропонувати граничні значення і негативні варіанти, систематизувати перевірки валідності відповіді. Важливо, що поля, статуси і правила перевірки мають братися лише зі специфікації, а не припускатися.

- Підтримка автоматизації тестування
  Штучний інтелект може допомогти з підготовкою структури тестів, формулюванням перевірок результату, підказками щодо стабільності, синхронізації та типових причин нестабільності. При цьому код, створений за допомогою інструментів, потребує рецензування і перевірки відповідності стандартам якості, вимогам безпеки та стилю проєкту.

- Аналіз дефектів і підготовка звітів
  Штучний інтелект корисний для первинного розбору дефектів, класифікації, формування гіпотез першопричини та шаблонізації якісного опису. Найважливіше правило: висновки формуються лише на основі наданих фактів, журналів подій і

відтворюваних кроків.

- Формування артефактів управління якістю
  Штучний інтелект може допомагати створювати карти ризиків, структурувати план покращень, формувати короткі підсумки і рекомендації. У цьому випадку особливо важливі докази, метрики та критерії приймання результату.

Межі застосування
Штучний інтелект не може замінити джерела істини, експертизу домену та перевірки в середовищі. Типові обмеження:

- відсутність доступу до реального стану системи, якщо не надано дані або артефакти
- схильність до вигаданих деталей, якщо контекст неповний
- помилкова впевненість формулювань, яка може створювати хибне відчуття правильності
- неможливість підтвердити виконання перевірок без реальних доказів

Практичний висновок: штучний інтелект використовується для підготовки ідей, структури і варіантів, а остаточний результат приймається лише після перевірки за правилами розділу 4.

1.2 Ризики та обмеження результатів, короткі правила безпеки

Застосування штучного інтелекту у тестуванні має типові ризики, які можуть призвести до некоректних рішень, пропуску критичних дефектів або витоку даних. Нижче наведено ключові ризики та короткі правила безпеки.

Основні ризики

- Вигадані вимоги або вигадані деталі
  Можлива поява правил, полів, статусів або поведінки, яких немає у вимогах чи специфікації.
- Неповний контекст
  Результат може виглядати коректним, але містити пропуски, якщо не передано обмеження, винятки або критичні умови.
- Підміна перевірки припущенням
  Результат може бути логічним, але не підтвердженим доказами або виконаними перевірками.

- Неврахування граничних умов і помилок
  Штучний інтелект часто пропонує типові позитивні сценарії, але може пропустити граничні та негативні перевірки, якщо це не вимагати явно.

- Ризик конфіденційності
  Передача чутливих даних у зовнішні інструменти може порушувати політики організації, угоди або нормативні вимоги.

Короткі правила безпеки і якості

- Використовувати лише дозволені джерела даних і не передавати конфіденційну інформацію без дозволу

- Завжди надавати достатній контекст: мета, обмеження, джерела істини, формат очікуваного результату

- Вимагати фіксації припущень та уточнювальних питань, якщо даних недостатньо

- Перевіряти результат на відсутність вигаданих деталей і суперечностей

- Для критичних правил обов’язково вимагати негативні і граничні перевірки

- Приймати результат лише після підтвердження доказами та перевірки за контрольним списком розділу 4

- Фіксувати докази: посилання на вимоги, приклади даних, результати прогонів, журнали подій

Зв’язок із методикою посібника
Перелічені ризики нейтралізуються авторською методикою керованого застосування: мінімальні вимоги до контексту, процес "запит - відповідь - перевірка - фіксація", матриця "Сценарій - Ризик - Контроль - Метрика", а також правила приймання або відхилення результату і метрики ефективності.

2. Авторська методика керованого застосування штучного інтелекту у тестуванні

Цей розділ описує авторську методику керованого застосування штучного інтелекту у тестуванні програмного забезпечення як відтворювану систему правил, перевірок і вимірюваних результатів. Методика створена для того, щоб використання штучного інтелекту було стандартизованим, контрольованим і придатним до впровадження в різних командах та навчальних групах. Ключова відмінність методики полягає у поєднанні практичних сценаріїв застосування з обов'язковою перевіркою результатів, підтвердженням доказами та фіксацією метрик ефективності.

2.1 Принципи методики та вимоги до контексту

Методика базується на принципі керованості: результат, отриманий із застосуванням штучного інтелекту, має бути прив'язаним до конкретної задачі тестування, перевіреним і пояснюваним. Для цього визначається мінімально необхідний контекст, без якого результат вважається ненадійним. Контекст включає мету задачі, опис функціональності, критерії приймання, обмеження і ризики, вхідні дані та очікуваний формат результату.

Таблиця 2.1 Мінімальний контекст для запиту до штучного інтелекту

Елемент контексту	Навіщо потрібен	Приклад формулювання
Мета задачі	Щоб результат був цільовим і перевірюваним	Сформувати негативні перевірки для реєстрації користувача
Опис функціональності	Щоб уникнути хибних припущень	Реєстрація через електронну пошту з підтвердженням кодом
Критерії приймання	Щоб перевірити правильність результату	Пароль не менше 8 символів, обов'язково цифра та літера
Обмеження і ризики	Щоб врахувати критичні умови	Після 5 невдалих спроб вхід блокується на 10 хвилин

Вхідні дані та формати	Щоб не вигадувалась структура даних	Поля: електронна пошта, пароль, підтвердження пароля
Очікуваний формат результату	Щоб результат було зручно оцінювати	Таблиця: кроки, дані, очікуваний результат
Пріоритети	Щоб зосередитись на важливому	Спочатку ризики безпеки і втрати даних
Відомі проблемні зони	Щоб врахувати реальні дефекти	Раніше були помилки у перевірці формату пароля

У подальшій роботі рекомендується використовувати мінімальний набір контексту, наведений у Таблиці 2.1, щоб підвищити відтворюваність результатів та зменшити ризик хибних припущень.

Другим принципом є доказовість: результати підтверджуються посиланням на джерела або перевіркою через тестові запуски, аналіз вимог, застосування технік тестового дизайну та інші контрольні дії. Третім принципом є вимірюваність: для кожного сценарію застосування визначаються метрики, які дозволяють оцінити якість результату і практичний ефект. Четвертим принципом є відтворюваність: порядок роботи та правила оцінювання описані так, щоб отримані результати були порівнюваними між різними групами, проєктами та виконавцями.

2.2 Процес роботи: запит - відповідь - перевірка - фіксація

Методика реалізується через чотири послідовні кроки, які забезпечують контроль якості та зменшують ризик помилкових висновків: запит, відповідь, перевірка, фіксація.

Таблиця 2.2 Процес роботи "запит - відповідь - перевірка - фіксація"

Крок	**Вхід**	**Дія**	**Вихідний артефакт**	**Мінімальна перевірка**

Запит	Контекст, мета, формат	Формулювання задачі і вимог до результату	Структурований запит	Є мета, критерії приймання, формат результату
Відповідь	Попередній результат	Уточнення структури, усунення неоднозначностей	Чернетка артефакту	Відповідає формату, немає суперечностей
Перевірка	Артефакт і джерела	Контрольні дії та контрольний список	Підтверджений результат	Зіставлено з вимогами, перевірено критичні ризики
Фіксація	Підтверджений результат	Збереження артефактів, доказів і метрик	Готовий артефакт	Є докази перевірки і визначені метрики ефекту

Процес роботи та мінімальні вимоги до перевірки кожного кроку наведено у Таблиці 2.2.

2.3 Ядро методики: матриця "Сценарій - Ризик - Контроль - Метрика"

Центральним елементом методики є матриця "Сценарій - Ризик - Контроль - Метрика", яка стандартизує застосування штучного інтелекту у тестуванні. Для кожного сценарію матриця задає типові ризики, конкретні контрольні дії для перевірки результату та метрики, які показують якість і практичний ефект. Завдяки матриці методика є масштабованою, придатною для навчальних програм і повторного використання у проєктах.

Матриця "Сценарій - Ризик - Контроль - Метрика" і процес роботи "запит - відповідь - перевірка - фіксація" є авторськими артефактами методики, придатними для повторного використання в навчальних програмах та командах тестування.

Таблиця 2.3 Приклад заповнення матриці "Сценарій - Ризик - Контроль - Метрика"

Сценарій	Ризик	Контроль (як перевіряємо)	Метрика (як вимірюємо)
Формування критеріїв приймання	Неповнота вимог	Зіставити з першоджерелом, перевірити винятки і граничні випадки	Кількість прогалин у вимогах; частка критеріїв, що відповідають першоджерелу
Генерація тестових сценаріїв	Пропуск критичних перевірок	Перевірити покриття ризиків, застосувати техніки тестового дизайну	Покриття вимог тестами; кількість пропусків після перевірки
Тестування програмних інтерфейсів	Хибні припущення про структуру	Зіставити зі специфікацією, перевірити схему і статуси	Покриття запитів; кількість дефектів, знайдених до випуску

Приклад заповнення матриці наведено у Таблиці 2.3, а повна версія матриці подана у Додатку 1.

3. Практичні сценарії використання

У розділі подано практичні сценарії застосування штучного інтелекту у типових задачах тестування та забезпечення якості. Кожен сценарій описано у логіці авторської методики: чітка мета, мінімальний контекст, очікуваний результат, контрольні дії та метрики. Для стандартизації роботи рекомендовано використовувати матрицю "Сценарій - Ризик - Контроль - Метрика" з Додатка 1 і фіксувати докази перевірки відповідно до розділу 4. Усі результати, отримані в межах сценаріїв 3.1-3.5, приймаються лише після перевірки за правилами розділу 4, з фіксацією доказів і виконанням контрольного списку.

Таблиця 3.1 Шаблон опису практичного сценарію використання штучного інтелекту

Поле	Що вказати	Приклад заповнення
Назва сценарію	Один з підрозділів 3.1-3.5	Тестовий дизайн: негативні та граничні сценарії
Мета	Який результат потрібно отримати	Сформувати набір негативних і граничних перевірок для критичних полів
Мінімальний контекст	4-6 пунктів контексту	Критерії приймання, правила валідації, винятки, пріоритети ризиків, формат результату
Запит до штучного інтелекту	Короткий структурований запит	Згенеруй таблицю тестів: кроки, дані, очікуваний результат. Включи негативні та граничні перевірки. Не вигадуй вимоги.
Очікуваний результат	Який артефакт має бути на виході	Таблиця тестових сценаріїв з конкретними даними та очікуваннями
Контрольні дії	Як перевіряємо результат	Зіставити з критеріями приймання, перевірити граничні значення, прибрати дублікати, перевірити відтворюваність
Докази перевірки	Що додаємо як підтвердження	Посилання на вимоги; приклад виконання 2-3 ключових перевірок; коментарі до виправлень
Метрика	Як вимірюємо ефект	Покриття критеріїв приймання; кількість пропущених сценаріїв після перевірки; час підготовки

Висновок	2-3 речення про результат	Додано 12 негативних і 8 граничних перевірок. Виявлено 3 прогалини у вимогах. Час підготовки скоротився на 30 відсотків.

3.1 Вимоги та критерії приймання

Мета сценарію: застосувати штучний інтелект для аналізу вимог, виявлення прогалин і суперечностей, а також формування критеріїв приймання, які є перевірюваними та однозначними.

Мінімальний контекст для роботи

- опис функціональності або історії користувача
- обмеження і винятки
- визначення даних і форматів (поля, типи, правила)
- приклади очікуваної поведінки
- пріоритети ризиків

Очікуваний результат

- перелік уточнювальних питань до стейкхолдера
- список припущень, які впливають на тестування
- критерії приймання у формі, придатній для перевірки
- перелік ризиків і зон підвищеної уваги

Контрольні дії

- зіставити сформовані критерії з першоджерелом вимог і перевірити, що немає вигаданих правил
- перевірити, що критерії приймання містять умову та очікуваний результат і не містять двозначних формулювань
- перевірити наявність винятків, помилкових сценаріїв і граничних умов

Метрики

- кількість виявлених прогалин або суперечностей у вимогах
- частка критеріїв приймання, які можна напряму перевірити тестом
- кількість уточнень, які були погоджені до початку тестування

3.2 Тестовий дизайн: позитивні, негативні та граничні сценарії

Мета сценарію: створити повний і збалансований набір тестових сценаріїв з урахуванням ризиків, граничних значень і типових помилок.

Мінімальний контекст для роботи

- критерії приймання
- перелік полів, правил і валідацій
- класи еквівалентності та граничні значення (якщо відомі)
- пріоритети ризиків та критичність функції
- дані для позитивних і негативних перевірок

Очікуваний результат

- структурований перелік тестових сценаріїв: кроки, дані, очікуваний результат
- окремий блок негативних сценаріїв
- окремий блок граничних перевірок
- покриття ризиків і винятків

Контрольні дії

- перевірити, що кожен критерій приймання має відповідні сценарії
- перевірити наявність негативних і граничних перевірок для критичних полів і правил
- прибрати дублікати та сценарії без чіткої перевірюваної цінності
- перевірити відтворюваність кожного сценарію: дані і очікування мають бути конкретними

Метрики

- покриття критеріїв приймання тестовими сценаріями
- кількість критичних граничних випадків, доданих після перевірки
- зменшення кількості дефектів, що потрапляють у пізні стадії через пропущені сценарії

3.3 Тестування програмних інтерфейсів: набори сценаріїв і валідація

Мета сценарію: застосувати штучний інтелект для створення набору перевірок програмних інтерфейсів з акцентом на валідацію структури, помилкові запити, обмеження і стабільність.

Мінімальний контекст для роботи

- опис доступних запитів і відповідей, правила авторизації
- структура полів, обов’язковість, типи даних, допустимі значення
- приклади коректних і некоректних запитів
- очікувані статуси у типових та помилкових випадках
- обмеження: ліміти, частота, розміри повідомлень

Очікуваний результат

- перелік сценаріїв перевірки: позитивні, негативні, граничні
- перевірки структури відповіді і обов’язкових полів
- перевірки помилок вхідних даних і обробки винятків
- шаблон звітування про дефекти інтерфейсів

Контрольні дії

- зіставити кожен сценарій із специфікацією та перевірити, що не додано вигаданих полів
- перевірити коректність статусів і повідомлень про помилки
- виконати вибіркові запуски ключових сценаріїв у тестовому середовищі

Метрики

- покриття ключових запитів сценаріями тестування
- кількість дефектів інтерфейсів, знайдених до випуску
- кількість невідповідностей між специфікацією і фактичною поведінкою

3.4 Автоматизація тестування: підготовка тестів, перевірка коду, аналіз нестабільності

Мета сценарію: застосувати штучний інтелект для прискорення підготовки автоматизованих тестів, підвищення якості перевірок і зменшення нестабільності.

Мінімальний контекст для роботи

- опис тестового сценарію, очікувані результати
- структура сторінок або компонентів, стабільні ідентифікатори
- правила очікувань і синхронізації
- вимоги до стилю коду і структури тестів
- приклади типових збоїв або нестабільної поведінки

Очікуваний результат

- заготовка тесту з чіткими перевірками
- рекомендації щодо покращення структури коду тестів
- перелік причин нестабільності і способів усунення
- мінімальний набір перевірок для надійності

Контрольні дії

- перевірити наявність перевірок результату, а не лише виконання кроків
- запустити тест повторно кілька разів у однакових умовах, зафіксувати відхилення
- перевірити коректність очікувань, обробки помилок і звільнення ресурсів
- провести перевірку коду на відповідність стандартам

Метрики

- частка нестабільних тестів у наборі
- час на підтримку автоматизованих тестів
- стабільність прогонів у безперервній інтеграції
- кількість правок після перевірки коду

3.5 Аналіз дефектів: первинний розбір, класифікація, пошук першопричин

Мета сценарію: застосувати штучний інтелект для прискорення первинного розбору дефектів, формування якісного опису і попередніх гіпотез щодо першопричин.

Мінімальний контекст для роботи

- кроки відтворення і фактичний результат
- очікуваний результат і критерій приймання
- журнали подій, повідомлення про помилки, скріншоти
- версія збірки, середовище, налаштування
- історія подібних дефектів (за наявності)

Очікуваний результат

- уточнені кроки відтворення і мінімальний приклад
- класифікація дефекту за типом і критичністю
- попередні гіпотези причин і що перевірити для підтвердження
- шаблон якісного звіту про дефект

Контрольні дії

- перевірити відтворюваність дефекту і коректність середовища
- зіставити дефект з вимогами і критеріями приймання
- перевірити, що гіпотези не суперечать журналам подій і фактам

- узгодити класифікацію та критичність з командою розробки за потреби

Метрики

- час первинного розбору дефекту
- частка дефектів з коректною класифікацією після підтвердження
- покращення якості звітів про дефекти за контрольним списком
- зменшення повторних дефектів у тій самій категорії

Таблиця 3.2 узагальнює сценарії 3.1-3.5 та показує зв’язок між метою роботи, очікуваним результатом, контрольними діями та метриками. Це забезпечує порівнюваність отриманих результатів у різних групах та проєктах.

Таблиця 3.2 Узагальнення сценаріїв 3.1-3.5

Підрозділ	**Мета**	**Очікуваний результат**	**Ключовий контроль**	**Метрика**
3.1 Вимоги та критерії приймання	Уточнити вимоги і сформувати перевірювані критерії приймання	Перелік питань, припущень, критеріїв приймання, ризиків	Зіставлення з першоджерелом, перевірка винятків і граничних умов	Кількість прогалин у вимогах; частка критеріїв, що можна перевірити тестом
3.2 Тестовий дизайн	Сформувати повний набір позитивних, негативних і граничних перевірок	Таблиця тестових сценаріїв з кроками, даними і очікуваним результатом	Покриття критеріїв приймання, перевірка граничних значень, усунення дублікатів	Покриття критеріїв приймання тестами; кількість доданих критичних сценаріїв

3.3 Тестування програмних інтерфейсів	Створити набір перевірок інтерфейсів і валідацій	Перелік сценаріїв: позитивні, негативні, граничні; перевірки структури відповіді	Зіставлення зі специфікацією, перевірка статусів і обов'язкових полів	Покриття ключових запитів; кількість невідповідностей специфікації
3.4 Автоматизація тестування	Підготувати якісні автоматизовані тести і зменшити нестабільність	Заготовка тесту з перевірками; перелік причин нестабільності та рекомендації	Перевірка наявності перевірок результату, повторні запуски, перевірка очікувань	Частка нестабільних тестів; стабільність прогонів; час підтримки
3.5 Аналіз дефектів	Прискорити первинний розбір і підвищити якість опису дефекту	Уточнені кроки, класифікація, гіпотези причин, що перевірити	Відтворюваність, зіставлення з вимогами, узгодження фактів з журналами подій	Час первинного розбору; точність класифікації після підтвердження

Приклади структурованих запитів у Таблиці 3.3 можуть використовуватися як шаблони для практичних завдань. Перед застосуванням необхідно замінити фрагменти у квадратних дужках на дані конкретної задачі. Запити є інструментом підготовки результату, але остаточний результат підлягає перевірці та прийманню відповідно до розділу 4.

Таблиця 3.3 Приклади структурованих запитів для сценаріїв 3.1-3.5

Підрозділ	Структурований запит (для копіювання)
3.1 Вимоги та критерії приймання	Проаналізуй опис функціональності: [опис функціональності]. Сформуй: 1) перелік уточнювальних питань, 2) перелік припущень, 3) критерії приймання у форматі "умова - очікуваний результат", 4) перелік винятків і ризиків. Не додавай вимог, яких немає в описі.
3.2 Тестовий дизайн	На основі критеріїв приймання: [критерії приймання] згенеруй таблицю тестових сценаріїв з колонками "кроки", "дані", "очікуваний результат". Обов’язково додай негативні та граничні перевірки для критичних правил. Уникай дублікатів і загальних формулювань.
3.3 Тестування програмних інтерфейсів	На основі опису інтерфейсу: [опис інтерфейсу або специфікація] сформуй набір перевірок: позитивні, негативні, граничні. Для кожної перевірки вкажи запит (параметри), очікуваний статус, ключові поля відповіді та правила валідації. Не вигадуй поля і статуси, якщо їх немає у специфікації.
3.4 Автоматизація тестування	Для сценарію: [опис сценарію] запропонуй структуру автоматизованого тесту з чіткими перевірками результату. Додай рекомендації щодо синхронізації і стабільності. Сформуй короткий список типових причин нестабільності і як їх перевірити.
3.5 Аналіз дефектів	Ось дані дефекту: кроки [кроки відтворення], фактичний результат [фактичний результат], очікуваний результат [очікуваний результат], середовище [середовище], журнали подій або повідомлення [журнали подій або повідомлення]. Сформуй: 1) уточнені кроки відтворення, 2) класифікацію і критичність, 3) 3-5 гіпотез першопричини з перевірками для підтвердження, 4) короткий шаблон якісного опису дефекту. Використовуй лише надані факти.

4. Правила перевірки результатів та контроль якості

Розділ визначає правила перевірки результатів, отриманих із застосуванням штучного інтелекту, та підхід до контролю якості таких результатів у навчальних і практичних завданнях. Мета розділу - забезпечити доказовість, відтворюваність і порівнюваність результатів, а також зменшити ризик помилкових висновків через неповний контекст або хибні припущення. У межах методики результат вважається прийнятним лише тоді, коли він підтверджений доказами, пройшов перевірку за контрольним списком і відповідає визначеним критеріям приймання.

4.1 Правила підтвердження результатів: що вважається доказом і як його фіксувати

Доказ у цьому посібнику - це артефакт або спостереження, яке підтверджує правильність, відтворюваність і практичну цінність результату. Докази потрібні для того, щоб відокремити обґрунтований результат від припущення, а також забезпечити можливість незалежної перевірки.

До доказів можуть належати:

- посилання на першоджерело вимог або критерії приймання, на підставі яких сформовано результат
- результати виконання перевірок у тестовому середовищі або протоколи запусків
- приклади вхідних даних і отриманих відповідей з підтвердженням правил валідації
- журнали подій або повідомлення про помилки, що обґрунтовують висновок
- скріншоти або інші артефакти, які демонструють фактичну поведінку
- результати рецензування коду тестів або набору тестових сценаріїв
- результати порівняння з історією дефектів або ризиковими зонами продукту

Правила фіксації доказів

- доказ фіксується разом із результатом, а не окремо
- кожен ключовий висновок має мати хоча б один доказ
- докази мають бути відтворюваними, тобто інша особа може повторити перевірку або переглянути артефакт

- якщо доказ неможливо надати, у результаті робиться позначка "потребує підтвердження" із зазначенням плану перевірки
- докази зберігаються у форматі, який дозволяє швидко знайти їх за посиланням або ідентифікатором

  Таблиця 4.1 визначає мінімальний набір доказів для типових сценаріїв і формат їх фіксації.

Таблиця 4.1 Мінімальний набір доказів і формат фіксації

Тип результату	**Що вважається доказом**	**Мінімальний набір доказів**	**Де фіксувати**
Критерії приймання	Посилання на вимоги і перевірка відповідності	Посилання на першоджерело; перелік винятків; перевірка однозначності	Робочий документ завдання або звіт
Тестові сценарії	Перевірювані кроки, дані, очікування, прив’язані до вимог	Матриця покриття; 2-3 приклади даних; граничні перевірки	Таблиця тестів або звіт
Програмні інтерфейси	Узгодженість зі специфікацією і вибіркове виконання запитів	Посилання на специфікацію; приклади запитів і відповідей; підтвердження статусів	Звіт або журнал перевірок
Автоматизовані тести	Код і підтвердження стабільності	Посилання на сценарій; результати повторних запусків; перевірки результату	Репозиторій коду і звіт
Аналіз дефекту	Факти з кроків і журналів подій	Мінімальні кроки; очікування і факт; фрагмент журналів подій	Картка дефекту або звіт

Звіт про ризики	Зв’язок ризику з фактом і рекомендацією	Ризик; вплив; доказ; рекомендація; пріоритет	Звіт або підсумок роботи

4.2 Контрольний список перевірки результатів штучного інтелекту

Контрольний список використовується для швидкої перевірки якості результату перед його прийманням. Він застосовується до будь-якого артефакту: критеріїв приймання, тестових сценаріїв, перевірок програмних інтерфейсів, автоматизованих тестів, описів дефектів або звітів про ризики. Контрольний список призначений для виявлення типових проблем: вигадані припущення, неповний контекст, нечіткі формулювання, пропуск граничних умов, відсутність доказів.

Таблиця 4.2 Контрольний список перевірки результатів штучного інтелекту

Пункт перевірки	**Так**	**Ні**	**Коментар**
Результат відповідає меті задачі			
Контекст достатній для висновків			
Відсутні суперечності або дублікати			
Немає вигаданих правил або даних			
Формулювання однозначні і перевірювані			
Враховано винятки, помилки, граничні випадки			

Докази наявні або визначено план підтвердження			
Визначено метрику або спосіб оцінювання			

Контрольний список у табличному форматі наведено в Таблиці 4.2. Його рекомендовано додавати до звіту або заповнювати під час перевірки результату.

4.3 Критерії приймання або відхилення результату

Результат приймається, якщо виконується сукупність умов:

- результат відповідає задачі та заданому формату
- ключові твердження підтверджені доказами або визначено чіткий план підтвердження
- відсутні критичні помилки, що можуть призвести до неправильного тестування або хибних висновків
- контрольний список перевірки виконано і немає невирішених пунктів високої критичності
- визначено метрики або критерії оцінювання ефекту

Результат відхиляється або повертається на доопрацювання, якщо:

- виявлено вигадані вимоги або припущення, які суперечать наданому контексту
- результат неможливо перевірити через нечіткість або відсутність очікуваного результату
- відсутні докази для ключових висновків, а план підтвердження не визначено
- пропущено критичні негативні або граничні перевірки, що впливають на якість
- є суперечності або дублікати, що знижують практичну цінність

Короткі критерії приймання і відхилення результату наведено в Таблиці 4.3.

Таблиця 4.3 Критерії приймання або відхилення результату

Критерій	Прийняти	Доопрацювати	Відхилити
Відповідність задачі та формату	Так	Частково	Ні
Наявність доказів для ключових висновків	Так	Частково	Ні
Відсутність критичних помилок	Так	Потрібні виправлення	Ні
Виконання контрольного списку	Так	Є невирішені пункти	Ні
Наявність метрик або критеріїв оцінювання ефекту	Так	Частково	Ні

4.4 Метрики якості та ефективності: що і як вимірювати

Метрики потрібні для об’єктивної оцінки якості результатів і практичного ефекту від застосування методики. Метрики обираються так, щоб їх можна було зібрати без надмірних витрат часу, і щоб вони дозволяли порівнювати результати між різними роботами, командами або навчальними групами. Для кожного сценарію рекомендується обирати 1-3 метрики якості результату та 1-2 метрики ефективності.

Підхід до вибору та фіксації метрик

- метрики прив’язуються до конкретного сценарію та ризиків
- метрики мають бути повторюваними і зрозумілими для вимірювання
- метрики фіксуються разом із результатом і доказами перевірки
- за можливості визначається базове значення, з яким порівнюється результат після застосування методики

Рекомендовані метрики для ключових сценаріїв наведено в Таблиці 4.4.

Таблиця 4.4 Рекомендовані метрики якості та ефективності

Сценарій	Метрики якості результату	Метрики ефективності
Вимоги і критерії приймання	Кількість прогалин; частка критеріїв, що є однозначними	Час узгодження; зменшення переробок
Тестовий дизайн	Покриття критеріїв; кількість пропущених критичних сценаріїв після перевірки	Час підготовки; зменшення дефектів через пропуски
Програмні інтерфейси	Частка перевірок, узгоджених зі специфікацією; кількість невідповідностей	Дефекти до випуску; час підготовки перевірок
Автоматизація тестування	Частка нестабільних тестів; кількість зауважень після перевірки коду	Стабільність прогонів; час підтримки
Аналіз дефектів	Частка коректної класифікації після підтвердження	Час первинного розбору; зменшення повторних дефектів

5. Підсумковий практикум: оцінювання зрілості процесу забезпечення якості на основі доказів

Розділ описує підсумковий практикум, у межах якого застосовується авторська методика для комплексної оцінки зрілості процесу забезпечення якості. Практикум побудований на доказовому підході: висновки формуються лише на основі підтверджених фактів, артефактів і результатів перевірок. Результатом практикуму є три взаємопов'язані артефакти: оцінка рівня зрілості 0-4, карта ризиків із пріоритизацією та план покращень 30-60-90.

5.1 Модель зрілості 0-4 та критерії рівнів

Модель зрілості 0-4 використовується для структурованої оцінки стану процесів тестування та забезпечення якості. Вона дозволяє зафіксувати поточний стан, виявити прогалини та сформувати план покращень із вимірюваними цілями. Оцінка виконується за критеріями, які підтверджуються доказами з артефактів процесу: тестової документації, результатів прогонів, звітів про дефекти, показників стабільності, практик роботи з ризиками та плануванням. Критерії рівнів зрілості 0-4 та розширені приклади доказів наведені у Додатку 2. Під час визначення рівня зрілості кожне твердження має спиратися на зафіксований доказ відповідно до правил розділу 4.

5.2 Карта ризиків і пріоритизація

Карта ризиків використовується для того, щоб пов'язати оцінку зрілості з практичними загрозами для продукту і випусків. Ризики визначаються на основі дефектів, прогалин у вимогах, нестабільності тестів, проблем з даними, інцидентів у виробничому середовищі, а також організаційних факторів. Для кожного ризику фіксуються вплив, імовірність, доказ, поточний контроль і рекомендовані дії. Пріоритет задається поєднанням впливу та імовірності, а для ризиків високого пріоритету визначаються метрики з розділу 4.

Шаблон карти ризиків із прикладом наведено в Таблиці 5.2.

Таблиця 5.2 Карта ризиків і пріоритизація

Поле	Приклад	Шаблон
Ризик	Пропуск граничних перевірок у критичному потоці	[назва ризику]

Вплив	Високий	[високий, середній, низький]
Імовірність	Середня	[висока, середня, низька]
Пріоритет	Високий	[високий, середній, низький]
Доказ	Дефект у випуску; відсутні граничні тести у наборі перевірок	[факт або артефакт]
Поточний контроль	Частковий набір тестів; перевірка виконується нерегулярно	[що вже є]
Рекомендована дія	Додати негативні і граничні сценарії; оновити контрольний список	[що зробити]
Метрика	Покриття критеріїв приймання; кількість дефектів до випуску	[що вимірювати]

5.3 План покращень 30-60-90: структура та приклад

План покращень 30-60-90 використовується для перетворення висновків практикуму на послідовні дії з вимірюваним ефектом. План будується від найбільш критичних ризиків до менш критичних і враховує залежності між задачами. Для кожної дії визначаються очікуваний результат, доказ виконання, метрика та критерій успіху. Практична логіка плану: 30 днів - стабілізація і закриття критичних прогалин, 60 днів - стандартизація та розширення покриття, 90 днів - оптимізація і профілактика повторних дефектів.

Таблиця 5.3 Шаблон плану покращень 30-60-90

Поле	30 днів (приклад)	60 днів (шаблон)	90 днів (шаблон)
Ціль	Стандартизувати критерії приймання і тестове покриття критичних правил	[ціль]	[ціль]
Дія	Узгодити критерії приймання для 3-5 ключових функцій; додати негативні і граничні сценарії для критичних правил; оновити контрольний список перевірки результатів	[дія]	[дія]
Очікуваний результат	Перевірювані критерії приймання; оновлений набір тестових сценаріїв; зменшення кількості пропущених критичних сценаріїв	[результат]	[результат]
Доказ виконання	Посилання на документ з критеріями; таблиця тестів із позначеним покриттям; протокол прогону	[доказ]	[доказ]
Метрика	Покриття критичних критеріїв; кількість уточнень після рев’ю; дефекти до випуску	[метрика]	[метрика]
Критерій успіху	Покриття не менше 90 відсотків для критичних критеріїв; зменшення пропусків критичних сценаріїв у порівнянні з базовим рівнем	[критерій]	[критерій]

Висновки

У посібнику систематизовано підхід до керованого застосування штучного інтелекту у тестуванні програмного забезпечення з акцентом на перевірюваність результатів, доказовість і вимірюваний ефект. Запропоновано авторську методику, що поєднує мінімальні вимоги до контексту, процес роботи "запит - відповідь - перевірка - фіксація" та матрицю "Сценарій - Ризик - Контроль - Метрика" для стандартизації практичних задач. Наведені практичні сценарії застосування охоплюють роботу з вимогами, тестовим дизайном, тестуванням програмних інтерфейсів, автоматизацією та аналізом дефектів, а також підсумковий практикум з оцінювання зрілості процесу забезпечення якості на основі доказів. Для підтримки впровадження та уніфікації перевірки практичних результатів подано таблицю критеріїв оцінювання виконаних робіт (Додаток 3). Оцінювання ефекту методики виконується на основі метрик якості та ефективності й підтверджується зафіксованими доказами відповідно до правил розділу 4, що забезпечує відтворюваність результатів і придатність методики до масштабування в різних командах та навчальних середовищах.

Список використаних джерел

1. ISO/IEC/IEEE 29119-1:2013. Software and systems engineering - Software testing - Part 1: Concepts and definitions.

2. ISO/IEC/IEEE 29119-2:2021. Software and systems engineering - Software testing - Part 2: Test processes.

3. ISO 9001:2015. Quality management systems - Requirements.

4. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.

5. ISTQB. Foundation Level Syllabus. (Офіційна програма базового рівня).

6. NIST. Secure Software Development Framework (SSDF).

7. OWASP. OWASP Testing Guide.

8. IEEE. Guide for Software Verification and Validation (IEEE 1012) або актуальна редакція стандарту з верифікації та валідації.

ДОДАТОК 1
Матриця "Сценарій - Ризик - Контроль - Метрика"

Сценарій	**Ризик**	**Контроль (як перевіряємо)**	**Метрика (як вимірюємо)**
Аналіз вимог	Неповнота або суперечності	Зіставлення з першоджерелом, перевірка винятків, уточнювальні питання	Кількість прогалин і суперечностей; кількість уточнень до погодження
Формування критеріїв приймання	Перекручення логіки	Перевірка відповідності вимогам, перевірка граничних умов	Частка критеріїв, що відповідають вимогам; кількість виправлень після перевірки
Тестовий дизайн: позитивні сценарії	Надмірна загальність	Перевірка на відтворюваність і однозначність, прив’язка до функцій	Кількість нечітких сценаріїв після перевірки; час підготовки
Тестовий дизайн: негативні і граничні сценарії	Пропуск критичних перевірок	Застосування граничних значень, еквівалентних класів, аналіз ризиків	Кількість доданих критичних сценаріїв; кількість дефектів до випуску
Регресійний контрольний список	Неоднозначні пункти	Перевірка формулювань, додавання очікуваного результату	Час виконання регресії; кількість неоднозначностей після перевірки
Підготовка тестових даних	Некоректні набори даних	Перевірка форматів, обмежень, унікальності, граничних значень	Кількість помилок у даних; час підготовки даних

Тестування програмних інтерфейсів	Хибні припущення про поля і перевірки	Зіставлення зі специфікацією, перевірка схем, виконання запитів	Покриття запитів; кількість дефектів інтерфейсу до випуску
Автоматизація тестування	Нестабільні тести	Перевірка очікувань, синхронізації, повторні запуски	Частка нестабільних тестів; стабільність прогонів
Перевірка коду тестів	Логічні помилки у перевірках	Перевірка коду за стандартами, запуск у середовищі	Кількість зауважень після перевірки; кількість виправлень
Аналіз дефектів	Неправильна класифікація	Перевірка логів, кроків відтворення, узгодження з розробкою	Час первинного розбору; точність класифікації після підтвердження
Оцінка ризиків випуску	Пропуск ризиків	Структура "ризик - вплив - доказ - рекомендація", звірка з метриками	Кількість ризиків до випуску; кількість інцидентів після
План покращень 30-60-90	Нереалістичні дії	Пріоритизація за ризиком, перевірка залежностей, критерії успіху	Частка виконаних дій; зміна метрик якості після впровадження

ДОДАТОК 2

Модель зрілості 0-4 з прикладами доказів

У цьому додатку наведено критерії рівнів зрілості 0-4 та приклади доказів, які можуть використовуватися для підтвердження рівня зрілості процесу забезпечення якості. Докази добираються відповідно до правил розділу 4: кожне твердження має спиратися на відтворюваний артефакт або зафіксований факт.

Таблиця Д2.1 Рівні зрілості 0-4: критерії та приклади доказів

Рівень	**Короткий опис**	**Критерії рівня**	**Приклади доказів (артефакти та факти)**
0	Процес відсутній або хаотичний	Немає узгоджених правил і повторюваних практик; результати залежать від окремих людей; артефакти відсутні або випадкові	Відсутні критерії приймання; відсутні регулярні прогони; немає єдиного формату тестів; дефекти без кроків відтворення або без очікуваного результату
1	Базовий, частково повторюваний	Є окремі практики, але вони непослідовні; стандарти відсутні або застосовуються вибірково; контроль якості результатів нерівномірний	Часткові шаблони тестів; поодинокі контрольні списки; нерегулярні звіти; різні формати опису дефектів; відсутня системна робота з ризиками
2	Визначений процес і базова стандартизація	Встановлені мінімальні правила; є стабільні артефакти; контрольний список використовується регулярно; є базові метрики	Узгоджені критерії приймання; регулярні результати прогонів; стандартний шаблон дефекту; таблиця тестів з негативними і граничними перевірками; базові метрики (покриття, час циклу)
3	Керований і вимірюваний	Рішення приймаються на основі метрик і доказів; є управління ризиками; покращення плануються і перевіряється їх ефект	Карта ризиків із пріоритизацією; метрики стабільності і ефективності; звіти з посиланнями на докази; підтверджений ефект покращень у динаміці; регулярні огляди якості

4	Оптимізований і прогнозований	Є системні цикли покращень; профілактика дефектів; стабільна автоматизація; ризики прогнозуються і контролюються наперед	Стабільна позитивна динаміка метрик; низька частка повторних дефектів; висока стабільність прогонів; стандартизовані практики аналізу причин; підтверджені превентивні дії і їх ефект

Таблиця Д2.2 Мінімальний пакет доказів для визначення рівня зрілості

Область	Що перевіряється	Мінімальні докази	Приклади формату фіксації
Вимоги і критерії приймання	Перевірюваність і однозначність	3-5 прикладів вимог з критеріями приймання; перелік винятків	Посилання на документ; витяг критеріїв у таблиці
Тестовий дизайн	Наявність негативних і граничних перевірок	Фрагмент тестового набору з граничними кейсами; матриця покриття	Таблиця тестів; посилання на версію
Прогони і звітність	Регулярність і відтворюваність	2-3 звіти прогонів з датами і середовищем	Звіт; журнал прогонів
Дефекти	Якість опису і класифікації	3-5 дефектів з кроками, очікуваним і фактичним результатом	Картки дефектів; посилання на систему
Автоматизація	Стабільність і підтримуваність	Дані про нестабільні тести; приклад перевірки коду	Звіт про прогони; протокол перевірки коду
Ризики	Пріоритизація і контроль	Карта ризиків з доказами і діями	Таблиця ризиків; підсумок рішень
Покращення	Наявність плану і підтвердження ефекту	План 30-60-90; метрики до і після	План у таблиці; звіт з метриками

Таблиця Д2.3 Приклад правил присвоєння рівня зрілості на основі доказів

Правило	Опис
Пріоритет доказів	Перевага надається фактам і артефактам, які можна відтворити або перевірити незалежно
Мінімальний поріг	Рівень присвоюється лише тоді, коли наявні мінімальні докази для ключових областей з Таблиці Д2.2
Обережність у висновках	Якщо докази часткові або суперечливі, рівень знижується до підтвердженого мінімуму
Відхилення неперевірених тверджень	Твердження без доказу фіксуються як "потребує підтвердження" і не впливають на підсумковий рівень
Прив’язка до метрик	За можливості рівень підкріплюється метриками стабільності, покриття та ефективності з розділу 4

ДОДАТОК 3
Таблиця критеріїв оцінювання виконаних робіт

Таблиця Д3.1 Критерії оцінювання виконаних робіт (уніфікована таблиця)

Критерій	**Високий рівень**	**Достатній рівень**	**Потребує доопрацювання**
Постановка задачі та контекст	Мета чітка, контекст повний, формат результату визначено	Контекст частковий, але достатній для виконання	Контекст неповний, мета нечітка, формат не визначено
Якість результату	Результат однозначний, перевірюваний, без суперечностей	Є незначні неточності або прогалини	Результат нечіткий, суперечливий або неперевірюваний
Перевірка і контрольні дії	Виконано контрольний список, перевірено критичні ризики	Перевірку виконано частково	Перевірка відсутня або формальна
Докази	Докази надані для ключових висновків	Докази часткові або неповні	Докази відсутні
Метрики і висновок	Обрано метрики, сформульовано висновок про ефект	Метрики зазначено частково або без пояснення	Метрики не визначені, висновок відсутній або загальний

Таблиця Д3.2 Коротка шкала підсумкової оцінки (за потреби)

Результат перевірки	**Опис**
Прийнято	Відповідає критеріям, докази наявні, результат придатний до використання
Прийнято з доопрацюванням	Є недоліки середньої критичності, потрібні уточнення або виправлення

Відхилено	Є критичні помилки або відсутні докази, результат не можна використовувати