

DOI: <https://doi.org/10.36910/6775-2524-0560-2023-53-04>

УДК: 004.415.53

Міскевич Оксана Іванівна, асистент

<https://orcid.org/0000-0002-5009-2391>

Омельчук Дмитро Юрійович, бакалавр

Луцький національний технічний університет, м.Луцьк, Україна

## РУЧНЕ ТА АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ БЛОКЧЕЙНУ ЗА ДОПОМОГОЮ POSTMAN

**Міскевич О.І., Омельчук Д.Ю. Ручне та автоматизоване тестування блокчейну за допомогою Postman.**

Людство створило дуже велику кількість різноманітних винаходів для облегшення та покращення життя. З часом, винаходи стають все більш складними, так само і з їх тестуванням. Блокчейн – це важлива частина світу сьогодення. Ця технологія застосовується в логістиці, охороні здоров'я, фінансових транзакція та багатьох інших сферах. І щоб забезпечити високу відмовостійкість, продуктивність та інші важливі аспекти програмного забезпечення кожен блокчейн потребує ретельного тестування на кожному етапі циклу його розробки. Дякуючи багатьом розробникам, які зробили вклад в блокчейн та програмне забезпечення, яке відноситься до нього, сьогодні ми маємо багато методів для створення тестових випадків для блокчейну. Зараз велика кількість компаній та організацій досліджують та впроваджують технологію блокчейну. Також, це створює попит на тестувальників, які забезпечують якість та надійність розподіленої системи. В цій статті ми розглянемо процес автоматизованого тестування блокчейну, його важливість та виклики, які виникають під час розробки блокчейн-продуктів. Ми будемо використовувати Postman для проведення потрібного тестування. Також, ми будемо використовувати автоматизоване тестування для виключення людського фактору з процесу.

В цій статті буде розказано про стадії автоматизованого тестування блокчейну і створимо один тестовий випадок.

**Ключові слова:** автоматизоване тестування, postman, блокчейн.

**Miskevych O., Omelchuk D. Manual and automated blockchain testing with Postman.** Humanity created an immense amount of various inventions to make life easier and better. As time passes, inventions become more and more complicated and so does its testing. Blockchain is an important part of today's world. This technology serves its purpose in logistics, healthcare, financial transactions, and many more sectors. And in order to provide resilient and high-performing product every single blockchain requires thorough testing at every point of its development life cycle. Thanks to the many developers who made their contribution to blockchain and other software that is related to it, today we have many ways to create test cases for blockchain. With time, software becomes more complex and so do tests: all things need to be in properly-working state. Blockchain testing has its challenges, such as performance and work under load, integration testing and security. In this article, we will be using Postman for conducting required testing. We will be using automated test to exclude human factor from our testing.

In this article we will cover all phases of automatic testing of our blockchain and create one test case.

**Key words:** automatic testing, postman, blockchain.

**Вступ.** В реальному світі та в світі інформаційних технологій існують численні системи, які використовуються для обробки даних та забезпечення їхньої надійності та цілісності. Однією з найбільш перспективних та обіцяючих технологій є блокчейн. Ця технологія революціонує спосіб обміну даними та фінансовими операціями, забезпечуючи високий рівень безпеки та відкритості.

Блокчейн - це розподілена система, яка зберігає та керує даними за допомогою ланцюга блоків. Кожен блок містить інформацію про певну кількість транзакцій, та всі блоки зв'язані між собою. Основна ідея блокчейну полягає в тому, що дані є незмінними та недоступними для втручання після того, як вони були збережені. Ця технологія була розроблена для забезпечення безпеки та надійності фінансових операцій, але вона також знаходить застосування в різних інших галузях.

**Основна частина.** Тестування блокчейну є невід'ємною частиною процесу розробки та впровадження цієї технології. Воно виконує низку важливих функцій, серед яких:

Верифікація смарт-контрактів - багато блокчейнів використовують смарт-контракти - програми, які автоматизовано виконують операції при виконанні певних умов і ручне тестування допомагає переконатися, що смарт-контракти працюють правильно та не містять помилок, які можуть призвести до втрати коштів або порушення угод;

Перевірка безпеки - безпека є однією з ключових переваг блокчейну, і ручне тестування допомагає виявити потенційні уразливості та ризики в системі, бо тестувальники перевіряють, чи існують можливості для атак та визначають, чи відповідає даний блокчейн стандартам безпеки;

Тестування мережі: розробники блокчейну створюють власні мережі для тестування та вдосконалення системи перед впровадженням, ручне тестування мережі дозволяє переконатися, що вона працює належним чином та витримує навантаження;

Тестування інтеграції - блокчейн часто використовується як складова багатьох інших систем. Тестувальники переконуються, що інтеграція блокчейну з іншими додатками або сервісами працює коректно.

Postman надає потужну платформу для ручного автоматизованого тестування додатків блокчейну. Він дуже ефективний у автоматизації тестування API, яке є обов'язковим для додатків блокчейну, оскільки вони зазвичай використовують API для взаємодії. Postman автоматизує тестування цих API, надсилаючи серію запитів із заданими тестовими даними та перевіряючи відповіді.

Тестування на основі даних - ще одна важлива функція Postman. Ця програма підтримує тестування на основі даних, де використовуються різні набори тестових даних для виконання одних і тих самих тестових скриптів. Це особливо важливо в блокчейн-додатках з численними вхідними даними та сценаріями транзакцій.

Тестування продуктивності - ще одна сильна сторона Postman. Тести можуть бути налаштовані так, щоб симулювати велику кількість одночасних запитів до мережі блокчейну, оцінюючи наскільки добре додаток впорається з навантаженням.

Під час ручного тестування тестувальники повинні вдаватися до відповідних дій для перевірки роботи системи. Наприклад, при перевірці транзакцій тестувальники повинні переконатися, що транзакції виконуються правильно та без помилок, вони перевіряють коректність передачі коштів та відстежують рух активів між користувачами. При тестуванні смарт-контрактів вони повинні бути ретельно протестовані, включаючи випадки виконання, невиконання та помилкові сценарії, а також потрібно перевірити як контракти взаємодіють між собою. Ручне тестування також включає в себе перевірку масштабованості системи та її продуктивності, де тестувальники навантажують систему та слідкують за реакцією системи на велику кількість операцій. Ще потрібно переконатися, що дані в блокчейні є достовірними та правильними, тому додатково тестувальники повинні перевіряти інформацію, яка зберігається у блоках, на відповідність стандартам. Ще задачею ручного тестування є тестування інтерфейсу - якщо система має інтерфейс для користувачів, тестувальники повинні перевірити його на відповідність дизайну та функціональним вимогам.

Ручне тестування займає багато часу, особливо в розподілених системах. Тому при розробці блокчейн-продуктів також активно використовують автоматизоване тестування.

Раніше у цій статті було згадано тестування транзакцій, у цій статті ми створимо тестовий випадок та протестуємо саме цю функцію нашого блокчейну.

### **Тестовий випадок №1**

**Назва тестового випадку:** Відправлення криптовалюти з гаманця

**Опис:** Цей тест перевіряє правильність відправлення криптовалюти з одного гаманця на інший у блокчейн-мережі.

#### **Кроки:**

1. В Postman перейти по адресі localhost:5000/mine і натиснути кнопку Send
2. Далі перейти по адресі localhost:5000/transactons/new
3. Тип запиту обрати POST, дані виду raw, формат – JSON
4. Ввести в поле запиту наступне:

```
{  
  "sender": "*хеш користувача*"  
  "recipient": "someone-with-other-address",  
  "amount": 1000  
}
```

5. Натиснути на кнопку "Send"
6. Обрати тип запиту GET, перейти по адресі localhost:5000/mine
7. Перейти по адресі localhost:5000/chain

Тут ми перевіримо як і можливість проведення самої транзакції, так і те, чи буде вона проведена якщо у користувача недостатня кількість монет.

Результат:

```
{  
  "chain": [  
    {  
      "sender": "*хеш користувача*"  
      "recipient": "someone-with-other-address",  
      "amount": 1000  
    }  
  ]  
}
```

```
    "index": 1,  
    "previous_hash": 1,  
    "proof": 100,  
    "timestamp": 1699201050.1327708,  
    "transactions": []  
  },  
  {  
    "index": 2,  
    "previous_hash": "ef485514d8ef0b1e2087eefa6cb9c10424914179712db3d9f311b7de468a  
3c0f",  
    "proof": 35293,  
    "timestamp": 1699208789.292067,  
    "transactions": [  
      {  
        "amount": 1,  
        "recipient": "f88d61f52800481eb60a07510011038e",  
        "sender": "0"  
      }  
    ]  
  },  
  {  
    "index": 3,  
    "previous_hash": "09ae4a3034c92c2d1e7511738e1f477a6f06ac0a44b5fce6e42521239a7b  
f1e0",  
    "proof": 35089,  
    "timestamp": 1699209072.607282,  
    "transactions": [  
      {  
        "amount": 1000,  
        "recipient": "someone-with-other-address",  
        "sender": "f88d61f52800481eb60a07510011038e"  
      },  
      {  
        "amount": 1,  
        "recipient": "f88d61f52800481eb60a07510011038e",  
        "sender": "0"  
      }  
    ]  
  }  
],  
"length": 3  
}
```

Тут ми можемо бачити те, що транзакція дійсно проводиться і зберігається в ланцюгу, але наш блокчейн не слідкує за балансом і дозволяє проводити транзакції з меншою кількістю монет ніж є у користувача. Звідси можна створити баг-репорт:

#### **Баг-репорт №1:**

Опис: потрібно перевіряти чи сума яка буде відправлена комусь є меншою або рівною поточному балансу користувача.

Очікуваний результат: блокчейн зупиняє транзакцію та виводить повідомлення про недостатню кількість монет на балансі.

Результат: транзакція проводиться і зберігається в ланцюгу.

Критичність багу: максимальна

Кроки для відтворення:

1. В Postman перейти по адресі localhost:5000/mine і натиснути кнопку Send
2. Далі перейти по адресі localhost:5000/transactions/new

3. Тип запиту обрати POST, дані виду raw, формат – JSON
4. Ввести в поле запиту наступне:

```
{  
  "sender": "*хеш користувача*",  
  "recipient": "someone-with-other-address",  
  "amount": 1000  
}
```
5. Натиснути на кнопку “Send”
6. Обрати тип запиту GET, перейти по адресі localhost:5000/mine
7. Перейти по адресі localhost:5000/chain

Тепер приступимо до автоматизованого тестування. Кроки для створення автоматизованого тесту:

Створення тестового скрипту - Тестувальники створюють тестові скрипти в Postman, вказуючи запити API та очікувані результати. Ці скрипти організовані в колекції.

Налаштування автоматизації - Postman може бути налаштованим на автоматичний запуск. Це включає визначення змінних, наборів даних та середовищ тестування.

Виконання - Як тільки автоматизовані тести налаштовані, їх можна виконувати автоматично. Postman відправляє запити, обробляє дані та перевіряє результати відповідно до заданих скриптів.

Звітність - Під час виконання автоматизованих тестів Postman генерує докладні звіти. Ці звіти включають статуси (пройшов/не пройшов), докладні журнали, часи відповідей та інші корисні дані про процес тестування. Будь-які помилки повідомляються для подальшого аналізу.

Під час ручного тестування тестувальники повинні вдаватися до відповідних дій для перевірки роботи системи. Наприклад: перевірка транзакцій - тестувальники повинні переконатися, що транзакції виконуються правильно та без помилок, вони перевіряють коректність передачі коштів та відстежують рух активів між користувачами; тестування смарт-контрактів: смарт-контракти мають бути ретельно протестовані, включаючи випадки виконання, невиконання та помилкові сценарії, а також потрібно перевірити як контракти взаємодіють між собою; тестування масштабованості та продуктивності - ручне тестування включає в себе перевірку масштабованості системи та її продуктивності., а тестувальники можуть навантажувати систему та вимірювати її реакцію на велику кількість операцій; валідація даних - важливо переконатися, що дані в блокчейні є достовірними та правильними, тестувальники повинні перевіряти інформацію, яка зберігається у блоках, на відповідність стандартам; тестування інтерфейсу - якщо система має інтерфейс для користувачів, тестувальники повинні перевірити його на відповідність дизайну та функціональним вимогам.

Ручне тестування займає багато часу, особливо в розподілених системах. Тому при розробці блокчейн-продуктів також активно використовують автоматизоване тестування.

Автоматизоване тестування є важливою складовою процесу розробки блокчейну. Воно дозволяє виконувати тести швидше та ефективніше, забезпечуючи високий рівень достовірності та точності результатів. Основні переваги автоматизованого тестування включають:

Автоматизоване виконання тестів: Автоматизовані тести можуть бути запущені автоматично при кожній зміні в коді або при певних подіях, що дозволяє виявляти помилки швидше та зменшувати ризик людських помилок.

Швидкість виконання: Автоматизовані тести виконуються набагато швидше, ніж ручні, що дозволяє збільшити ефективність розробки та впровадження.

Повторюваність та послідовність: Автоматизовані тести виконуються послідовно та точно кожен раз, що дозволяє виявляти проблеми, які можуть бути непередбачувані під час ручного тестування.

Тестування на великому обсязі даних: Автоматизовані тести дозволяють виконувати тестування на великому обсязі даних, таких як масштабованість та продуктивність, без значних витрат часу та зусиль.

Автоматизоване тестування блокчейну передбачає використання спеціалізованих інструментів та підходів для створення тестів та їх автоматичного виконання. Ось деякі з важливих аспектів автоматизованого тестування блокчейну:

Сценарії тестування: Визначення сценаріїв тестування є важливим етапом автоматизації. Тестувальники розробляють сценарії, які покривають різні аспекти роботи системи, включаючи взаємодію з смарт-контрактами, обробку транзакцій, перевірку безпеки та інші функціональність.

Вибір інструментів: Для автоматизованого тестування блокчейну використовуються різні інструменти, такі як Truffle, Mocha, та інші. Вибір інструментів залежить від конкретних потреб проекту та технологій, що використовуються.

Розробка тестового середовища: Важливо створити тестове середовище, яке відтворює умови роботи реальної мережі блокчейну. Це дозволяє виконувати тести в ідентичних умовах та переконатися, що система працює як очікувалося.

Автоматизоване виконання тестів: Створені тестові сценарії виконуються автоматично з використанням вибраних інструментів. Результати тестів аналізуються для виявлення помилок та невідповідностей.

Навчання моделей тестування: В деяких випадках використовуються машинне навчання та штучний інтелект для автоматизації тестування. Моделі можуть виявляти аномалії та підказувати тестувальникам, де шукати проблеми.

Автоматизоване відстеження помилок: Система відстеження помилок дозволяє ефективно керувати виявленими дефектами та вирішувати їх у найкоротший термін. Після виявлення проблем автоматизований процес може генерувати звіти та повідомлення для розробників.

Автоматизоване тестування дозволяє значно скоротити час виконання тестів та збільшити їхню покриття. Проте важливо розуміти, що автоматизоване тестування не може замінити ручне тестування повністю, оскільки ручне тестування важливе для перевірки аспектів, які не можуть бути автоматизовані, такі як інтуїція користувача та інші аспекти, які потребують від людини креативного підходу.

Виклики тестування блокчейну

Тестування блокчейну має свої власні виклики та особливості, які варто враховувати при розробці тестових стратегій. Деякі з них включають:

Складність системи: Блокчейн - це розподілена система зі складними механізмами. Тестувальники повинні мати глибоке розуміння цих механізмів, щоб ефективно визначити потенційні проблеми.

Невизначеність стандартів: На ринку існує багато різних реалізацій блокчейну, і стандарти ще не повністю визначені. Це ускладнює розробку тестових кейсів та сценаріїв.

Приватність та безпека: Безпека та приватність є важливими аспектами блокчейну, і вони вимагають особливої уваги при тестуванні. Тестувальники повинні переконатися, що конфіденційні дані не витікають та що система захищена від зловмисних атак.

Великий обсяг даних: Блокчейн може містити великий обсяг даних, особливо якщо він використовується для зберігання історії транзакцій. Тестувальники повинні мати можливість ефективно керувати цим обсягом та виконувати тести на різних масштабах.

Сценарії розгортання: Блокчейн може бути розгорнутий у різних сценаріях, включаючи публічні та приватні мережі, мережі спеціального призначення та інші. Тестувальники повинні враховувати ці різні сценарії при створенні тестових кейсів.

Незмінність даних: Однією з ключових особливостей блокчейну є незмінність даних, тобто дані не можуть бути видалені чи змінені після додавання до блокчейну. Це ускладнює виправлення помилок та робить важливим тестування перед впровадженням.

Синхронізація мережі: У розподіленій мережі блокчейну важливо переконатися, що всі учасники мережі синхронізовані та отримують дані вчасно. Тестувальники повинні виконувати сценарії синхронізації для визначення потенційних проблем.

Заключні думки

Тестування блокчейну є складним та важливим процесом для забезпечення якості та надійності розподіленої системи. Ручне та автоматизоване тестування мають свої власні переваги та виклики, і їх комбінація дозволяє досягти найкращих результатів.

Важливо мати глибоке розуміння технології блокчейну та його особливостей для ефективного тестування. Тестувальники повинні визначати важливі аспекти для конкретного проекту, розробляти відповідні стратегії тестування та виявляти проблеми, які можуть виникнути під час роботи системи.

Блокчейн може бути використаний для різних цілей, включаючи фінансові транзакції, розподілені реєстри, обмін медичними даними та багато інших. Звідси, тестування повинно бути спрямоване на конкретні потреби та вимоги проекту. Це технологія майбутнього, яка вже змінює спосіб обміну даними та фінансовими операціями. Тестування грає важливу роль у забезпеченні надійності та безпеки цієї технології, і воно буде продовжувати розвиватися та вдосконалюватися разом із блокчейном.

Ця стаття надає загальний огляд тестування блокчейну та його важливості в сучасному інформаційному суспільстві. Блокчейн вже знайшов своє застосування в різних галузях, включаючи фінанси, логістику, медицину та багато інших. Однак ця технологія не є панацеєю і також вимагає тестування для забезпечення її надійності та безпеки.

Важливо розуміти, що блокчейн - це не лише технологія, але і концепція розподіленої бази даних, яка вимагає нових підходів до розробки та тестування. Ручне тестування залишається важливою складовою процесу, оскільки воно дозволяє перевіряти аспекти, які не можуть бути автоматизовані, такі як інтуїція користувача та реакція на непередбачені ситуації.

Автоматизоване тестування, у свою чергу, допомагає виконувати тести швидше та ефективніше, зменшуючи ризик людських помилок та забезпечуючи вищу точність результатів. Використання відповідних інструментів та підходів до автоматизації дозволяє розробникам та тестувальникам вдосконалити процес розробки та впровадження блокчейн-продуктів.

Незважаючи на виклики, пов'язані з тестуванням блокчейну, ця технологія обіцяє змінити багато аспектів сучасного суспільства. Вона може покращити безпеку фінансових транзакцій, забезпечити прозорість управління даними та сприяти розвитку нових галузей, таких як інтернет речей та цифрова медицина.

У майбутньому тестування блокчейну буде лише набирати обертів і розвиватися разом із самою технологією. Тестувальники та розробники матимуть нагоду вдосконалювати свої навички та вивчати нові методи тестування, щоб забезпечити високу якість блокчейн-продуктів.

Таким чином, тестування блокчейну відіграє важливу роль у розвитку цієї технології та допомагає забезпечити її надійність та безпеку. Воно вимагає спеціалізованих знань та навичок, але відкриває нові можливості для інновацій та розвитку. Тестування блокчейну є важливою складовою цього цифрового майбутнього і продовжуватиме грати ключову роль в його розвитку.

В код блокчейну були внесені корективи і тепер там присутня перевірка балансу користувача, помилка при недостатній кількості коштів буде мати код 51. Код який потрібно ввести в вкладку Tests в Postman:

```
pm.sendRequest({
  url: 'http://localhost:5000/mine',
  method: 'GET'
}, function (err, response) {
  if (err) {
    pm.test('GET запит /mine має помилку', function () {
      pm.expect.fail('GET /mine request failed');
    });
  } else {
    pm.test('GET запит /mine спрацював', function () {
      pm.expect(response).to.have.status(200);
    });

    var transactions = pm.response.json().transactions;
    var lastTransaction = transactions[transactions.length - 1];
    var recipient = lastTransaction.recipient;
    pm.environment.set('recipient', recipient);

    pm.sendRequest({
      url: 'http://localhost:5000/transactions/new',
      method: 'POST',
      header: 'Content-Type: application/json',
      body: {
```

```
mode: 'raw',
raw: JSON.stringify({
  "sender": recipient,
  "recipient": "someone-else",
  "amount": 10
})
}, function (err, response) {
  if (response.code !== 51) {
    pm.test('POST запит /transactions/new має помилку', function () {
      pm.expect.fail('POST /transactions/new request failed');
    });
  } else {
    pm.test('POST запит /transactions/new спрацював, помилка 51', function () {
      pm.expect(response).to.have.status(51);
    });
  }
});
});
```

В ньому ми майнемо 1 монету, беремо наш хеш з результату і відправляємо її на адресу "someone-else". Якщо з'явиться помилка то Postman повідомить, що транзакція не була проведена і виведе код помилки, якщо помилка 51 то тест буде успішним.

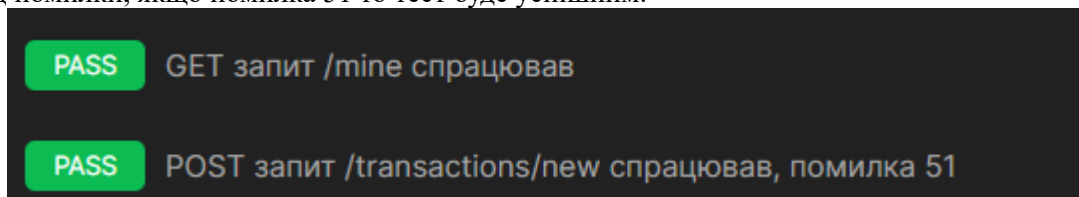


Рис. 1. Інтерфейс вкладки Test Results в Postman

Отже, баг був вирішений та успішно протестований.

### Результат дослідження

Почнемо з переваг ручного тестування:

Гнучкість: Тестувальники мають можливість досліджувати додаток, адаптувати тестові випадки під час тестування та вивчати можливу неочікувану поведінку. Загалом, ручне тестування дає можливість тестувальникам реагувати на зміни та вносити корективи до тестових сценаріїв під час розробки.

Моделювання сценаріїв реального світу: Ручне тестування дозволяє тестувальникам імітувати сценарії реального світу для перевірки того що додаток відповідає очікуванням користувачів.

Оцінка безпеки: Тестувальники можуть активно шукати вразливості з точки зору безпеки, що є важливим аспектом тестування додатків блокчейну.

Можливість використовувати творчий підхід: Тестувальники можуть використовувати свій досвід та інтуїцію для пошуку помилок, які можуть бути не знайденими автоматизованими інструментами.

Пошук нових видів помилок: Ручне тестування дозволяє виявити нові види помилок, які не були передбачені в автоматизованих сценаріях.

У ручного тестування є і недоліки:

Час і затрати: Ручне тестування вимагає багато часу і зусиль, особливо при великих проектах.

Людські помилки: Люди можуть допускати помилки та пропускати потенційні проблеми незважаючи на їх досвід та кваліфікацію.

Повторюваність: Ручні тести важко повторювати, і це може призвести до недостатнього покриття тестами.

Автоматизоване тестування, як і ручне, має свої переваги. Наприклад, автоматизовані тести можна виконувати швидко і дуже велику кількість разів, що прискорює процес тестування. Це корисно в сфері програмування блокчейн-програм, де важлива безперервна інтеграція. Ще автоматизоване тестування забезпечує послідовне виконання тестів, в цьому виді тестування неможливі помилки людського фактору та варіацій в тестуванні. Під час життєвого циклу розробки постійно вносяться нові зміни або оновлення. За допомогою автоматизованих тестів можна перевіряти новий код на предмет відповідності існуючим тестовим випадкам. Тестування навантаження є важливим для блокчейн-застосунків, які повинні обробляти велику кількість транзакцій. Автоматизоване тестування може симулювати велику навантаженість для оцінки продуктивності при стресовому тестуванні. Автоматизуючи рутинні тести, що включають велику кількість повторюваних взаємодій з мережею блокчейну, ресурси можна перерозподілити для більш складних тестів.

У автоматизованого тестування також є і недоліки. Створення автоматизованих тестів може бути витратним завданням, особливо на початковому етапі розробки. Автоматизовані тести потребують постійного обслуговування та оновлення під час змін в коді програми. Автоматизовані тести не можуть виявити помилки, які не були передбачені в сценаріях.

**Висновки.** У цій статті була пояснена потреба блокчейн-програм та як саме можна протестувати їх. Як ми дослідили у цій статті, Postman є корисним інструментом для забезпечення якості блокчейну.

Ручне тестування надає тестувальникам комплексне розуміння системи блокчейну. Воно дозволяє перевіряти процеси реєстрації користувачів, верифікацію транзакцій, взаємодію розумних контрактів та протоколи безпеки. Шляхом умісного введення помилок та детального аналізу ручне тестування розкриває вразливості та забезпечує надійність.

Під час ручного тестування тестувальники можуть імітувати сценарії реального життя, визначати граничні випадки та перевіряти коректність кожного процесу. Проте ручне тестування може бути витратним за часом і не завжди охоплює весь обсяг системи блокчейну.

Автоматизовані тести вносять ефективність та послідовність в процес тестування. Тестові скрипти полегшують виконання кількох тестових випадків однією командою. Здатність адаптувати середовища тестування та генерувати якісні звіти підвищує ефективність тестування.

Ручне та автоматизоване тестування з використанням Postman становлять потужну комбінацію для забезпечення надійності, якості та безпеки систем блокчейну. Тестувальники досліджують особливості технології блокчейну, виявляють вразливості та запобігають непередбаченим наслідкам завдяки засобам тестування як Postman. Оскільки технологія блокчейн продовжує революціонізувати різні галузі, ретельне тестування та інструменти, подібні Postman, залишатимуться ключовими для забезпечення якості та безпеки блокчейну.

#### Список бібліографічного опису

1. Христинець Н., Міскевич О., Мазуренко В. Технології Blockchain для оптимізації процесів документообігу. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2020. №40. С. 153-157.
2. Blockchain Testing - A Comprehensive Guide. Електронний ресурс. – URL: <https://www.guru99.com/blockchain-testing.html> (дата звернення 06.11.2023).
3. Introduction to Blockchain Testing. Електронний ресурс. – URL: <https://www.softwaretestinghelp.com/blockchain-testing/> (дата звернення 10.11.2023).
4. Blockchain Testing - A Complete Guide. Електронний ресурс. – URL: <https://www.edureka.co/blog/blockchain-testing> (дата звернення 12.11.2023).

#### References

1. Khrystinets N., Miskevich O., Mazurenko V. Blockchain technologies for optimizing document flow processes. Computer-integrated technologies: education, science, production. 2020. No. 40. P. 153-157.
2. Blockchain Testing - A Comprehensive Guide. Electronic resource. – URL: <https://www.guru99.com/blockchain-testing.html> (access date 06.11.2023).
3. Introduction to Blockchain Testing. Electronic resource. – URL: <https://www.softwaretestinghelp.com/blockchain-testing/> (access date 11.10.2023).
4. Blockchain Testing - A Complete Guide. Electronic resource. – URL: <https://www.edureka.co/blog/blockchain-testing> (access date 11.12.2023).