УДК 004.4:004.8:004.052
DOI https://doi.org/10.32689/maup.it.2025.4.8

Віталій ЖЕПЛІНСЬКИЙ
здобувач вищої освіти кафедри інформаційних систем та мереж,
Національний університет «Львівська політехніка», vitalii.zheplinskyi.mnitm.2024@lpnu.ua
***ORCID:** 0009-0003-5267-4402*

Михайло ЛУЧКЕВИЧ
кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних систем та мереж,
Національний університет «Львівська політехніка», mykhailo.m.luchkevych@lpnu.ua
***ORCID:** 0000-0002-2196-252X*

ГІБРИДНА МОДЕЛЬ ГЕНЕРАЦІЇ ТЕСТОВИХ СЦЕНАРІЇВ НА ОСНОВІ LLM ТА АНАЛІЗУ ІСТОРІЇ ДЕФЕКТІВ

Анотація. *У статті представлено концепцію гібридної моделі автоматизованого тестування програмного забезпечення, що поєднує можливості великих мовних моделей із аналізом історії дефектів. Модель орієнтована на формування ризик-орієнтованих тестових сценаріїв, здатних адаптивно реагувати на зміни у програмних системах та підвищувати ефективність процесів тестування. Основна увага приділяється інтеграції генеративних можливостей LLM із механізмами аналітики логів, що дозволяє формувати релевантні набори тестів у контексті CI/CD середовищ.*

Мета. *Метою дослідження є розробка концепції інформаційної системи, яка інтегрує технології штучного інтелекту та методи аналізу дефектів для забезпечення більш точного та своєчасного виявлення помилок.*

Методологія. *Особливістю підходу є побудова універсальної гібридної моделі, що може бути масштабована для різних предметних областей розробки ПЗ і не обмежується конкретним класом систем чи технологій. У перспективі модель підтримуватиме механізми пріоритизації тестів за рівнем ризику, що враховуватимуть частоту та критичність дефектів, а також здатність до автоматичного відновлення сценаріїв при змінах у вимогах.*

Наукова новизна. *Наукова новизна дослідження полягає у поєднанні генеративних можливостей LLM з ризик-орієнтованим аналізом історії дефектів, що дозволяє створювати адаптивні та релевантні сценарії тестування.*

Висновок. *Практична цінність полягає у можливості інтеграції запропонованої моделі у сучасні процеси CI/CD з метою зменшення витрат на підтримку автотестів, підвищення точності виявлення критичних помилок і загальної надійності програмного забезпечення. Запропонована модель демонструє потенціал для розвитку інтелектуальних механізмів автоматизованого тестування та побудови гнучкої інфраструктури забезпечення якості, що сприяє ефективному поєднанню інженерних практик із сучасними AI-технологіями.*

Ключові слова: *автоматизоване тестування, великі мовні моделі, історія дефектів, гібридна модель, ризик-орієнтовані сценарії, CI/CD.*

Vitalii ZHEPLINSKYI, Mykhailo LUCHKEVYCH. HYBRID MODEL OF TEST SCENARIO GENERATION BASED ON LLM AND DEFECT HISTORY ANALYSIS

Abstract. *The article presents the concept of a hybrid model of automated software testing that combines the capabilities of large language models with defect history analysis. The model is focused on forming risk-based test scenarios that can adaptively respond to changes in software systems and improve the efficiency of testing processes. Particular attention is paid to integrating the generative capabilities of LLMs with log analytics mechanisms, which enables the creation of relevant test sets in the context of CI/CD environments.*

Purpose. *The purpose of the study is to develop a concept of an information system that integrates artificial intelligence technologies and defect analysis methods to ensure more accurate and timely error detection.*

Methodology. *The key feature of the approach is the construction of a universal hybrid model that can be scaled for various software development domains and is not limited to a specific class of systems or technologies. In the future, the model will support test prioritization mechanisms based on risk levels, taking into account defect frequency and criticality, as well as the ability to automatically restore scenarios when requirements change.*

Scientific novelty. *The scientific novelty of the research lies in combining the generative capabilities of LLMs with risk-based defect history analysis, which enables the creation of adaptive and relevant testing scenarios.*

Conclusion. *The practical value lies in the possibility of integrating the proposed model into modern CI/CD processes to reduce the cost of maintaining automated tests, increase the accuracy of detecting critical errors, and improve overall software reliability. The proposed model demonstrates the potential for developing intelligent mechanisms of automated testing and building a flexible quality assurance infrastructure that promotes the effective combination of engineering practices with modern AI technologies.*

Key words: *automated testing, large language models, defect history, hybrid model, risk-based scenarios, CI/CD.*

Постановка проблема. Розвиток сучасних інформаційних технологій призвів до суттєвого ускладнення програмних систем та зростання вимог до їхньої якості. В умовах постійного скорочення життєвого циклу програмного забезпечення та широкого застосування практик Continuous Integration / Continuous Deployment (CI/CD) ключовим завданням стає забезпечення надійного та ефективного тестування [1]. Водночас традиційні підходи до автоматизованого тестування демонструють низку обмежень.

По-перше, процес створення тестових сценаріїв є трудомістким і вимагає значних часових та людських ресурсів. По-друге, тестові сценарії швидко втрачають актуальність у зв'язку зі змінами у вимогах та програмних інтерфейсах. По-третє, на практиці часто спостерігається проблема flaky-тестів, які характеризуються непередбачуваними результатами виконання, що знижує рівень довіри до автоматизованого тестування та потребує додаткового ручного втручання.

У відповідь на зазначені виклики у науковій спільноті зростає інтерес до застосування методів штучного інтелекту (AI) для підтримки та оптимізації процесів тестування. Окрему увагу приділяють використанню великих мовних моделей, які демонструють здатність до аналізу вимог, програмного коду та документації з метою автоматизованої генерації тестових сценаріїв [2]. Дослідження у сфері глибокого навчання підтверджують перспективність застосування таких підходів у створенні тестових випадків, проте здебільшого вони зосереджуються лише на одному аспекті – генерації тестів на основі текстових специфікацій.

Таким чином, постає проблема розробки гібридної моделі, яка б поєднувала можливості великих мовних моделей із аналізом історії дефектів для створення релевантних та адаптивних тестових сценаріїв. Реалізація такого підходу дозволить підвищити ефективність автоматизованого тестування, забезпечити пріоритизацію тестів за рівнем ризику та знизити витрати на підтримку тестового середовища.

Аналіз останніх досліджень та публікацій. Упродовж останнього десятиліття спостерігається стале зростання інтересу до застосування методів штучного інтелекту у процесах забезпечення якості програмного забезпечення, зокрема в автоматизованому тестуванні. Базові підходи до автоматичної генерації тестових сценаріїв опираються на використання глибокого навчання для інтерпретації текстових вимог і специфікацій та перетворення їх у формалізовані послідовності дій [3]. Доведено, що такі методи здатні підвищувати швидкість підготовки тестів і зменшувати частку ручної праці, однак їх ефективність суттєво залежить від повноти й якості вихідної текстової документації, а також від наявності чіткої структурованості вимог [10]. Додатковим обмеженням виступає недостатня зв'язаність автоматично згенерованих сценаріїв із реальними дефектами конкретних систем, що призводить до покриття здебільшого «щасливих шляхів» і пропуску критичних, але менш очевидних випадків [8].

У площині застосування великих мовних моделей у тестуванні накопичуються свідчення їх здатності інтерпретувати природномовні артефакти розроблення (user stories, технічні описи, issue-звіти) та синтезувати як тест-кейси, так і код автотестів [9]. Сучасні праці наголошують, що великі мовні моделі (LLM) формують новий клас інструментів підтримки інженерії програмного забезпечення, однак підкреслюють потребу в безпечному вбудовуванні таких моделей у процеси, з огляду на ризики галюцинацій, чутливість до формулювань запитів та відсутність гарантій коректності виводу [7]. Для тестування це означає необхідність додаткових механізмів валідації й пріоритизації, аби відсіяти нерелевантні чи надлишкові сценарії та сфокусуватися на найбільш значущих перевірках.

Сумарний аналіз показує наявність методологічного розриву між лінією досліджень, що зосереджена на генерації тестів із текстових артефактів, і лінією, орієнтованою на передбачення дефектів та аналіз експлуатаційних даних [5]. Перша пропонує високу продуктивність синтезу сценаріїв, але недостатньо враховує емпіричну дефектність конкретної системи; друга забезпечує знання про ризикові зони, проте рідко транслюється у безпосереднє конструювання тестів.

Підсумовуючи, сучасний стан наукових публікацій характеризується наявністю зрілих, але роз'єднаних напрямів: генерація тестів на основі тексту, дефект-предикція та лог-аналітика. Наявні роботи рідко пропонують конструктивний механізм злиття цих ліній у єдину технологічну схему, придатну для безперервних процесів CI/CD і масштабовану в різних доменах [6]. Саме ця прогалина і визначає наукову нішу, яку спрямовано заповнює запропонована у статті гібридна модель генерації тестових сценаріїв на основі LLM та аналізу історії дефектів.

Метою статті є розроблення та обґрунтування гібридної моделі генерації тестових сценаріїв, що поєднує можливості великих мовних моделей та аналітику історії дефектів програмного забезпечення. Такий підхід спрямований на підвищення ефективності автоматизованого тестування шляхом формування ризик-орієнтованих сценаріїв, які забезпечують кращу адаптивність тестових наборів, зменшення витрат на їх підтримку та підвищення рівня виявлення критичних помилок у процесах CI/CD.

Виклад основного матеріалу. Запропонована в даній роботі концепція ґрунтується на створенні гібридної моделі генерації тестових сценаріїв, яка поєднує можливості великих мовних моделей та аналіз історії дефектів програмного забезпечення. Такий підхід спрямований на вирішення проблеми низької адаптивності традиційного автоматизованого тестування та обмеженої здатності існуючих систем враховувати реальні ризики, що підтверджуються історично накопиченими даними про помилки. На відміну від класичних рішень, де тестові сценарії створюються вручну або шляхом використання жорстко закодованих правил, гібридна модель передбачає автоматизовану генерацію сценаріїв з урахуванням статистики дефектів, що дозволяє орієнтуватися на найбільш уразливі ділянки системи.

Архітектурно модель складається з трьох взаємопов'язаних блоків (рис. 1). Перший блок, заснований на LLM, приймає на вхід текстові вимоги, користувацькі історії або документацію та генерує первинний набір тестових сценаріїв. Другий блок здійснює аналіз історії дефектів, журнали виконання програмного забезпечення та дані систем контролю версій. У результаті цього аналізу обчислюється рівень ризику окремих компонентів, що дозволяє визначати їх пріоритетність при тестуванні. Третій блок інтеграції об'єднує результати роботи перших двох, формуючи ризик-орієнтований набір тестів, який може бути безпосередньо інтегрований у середовище CI/CD.

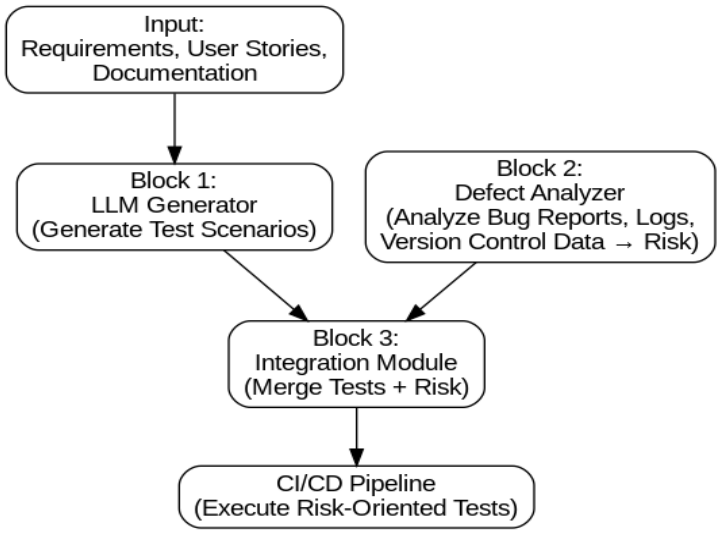

Рис. 1. Архітектурна модель гібридної системи генерації тестових сценаріїв

Для формалізації механізму пріоритизації вводиться функція ризику для компонента R(c):

$$R(c) = \sum_{i=1}^{n} \alpha f_i + \beta p_i + \gamma w_i, \quad (1)$$

де f_i – частота виникнення дефекту певного типу, p_i – ймовірність його повторного виникнення, w_i – ваговий коефіцієнт, що відображає критичність, а α, β, γ – коефіцієнти, які задають значущість кожного фактора. Обчислений рівень ризику використовується для динамічного визначення кількості тестів, що мають бути згенеровані для кожного компонента: чим вищий показник R(c), тим більший обсяг тестових сценаріїв формує система.

Інтеграція моделі у CI/CD-середовище забезпечує автоматичне виконання сценаріїв на кожному етапі життєвого циклу розробки. Процес можна описати як послідовність: розробник завантажує зміни у репозиторій; система CI виконує збірку; запускається модуль аналізу дефектів, який оновлює ризикову модель; LLM генерує нові сценарії або оновлює наявні; об'єднаний набір тестів виконується, а результати зберігаються у звіті. Даний підхід зменшує час на створення тестів та підвищує їхню релевантність у порівнянні з традиційними методами [4].

Оцінювання ефективності запропонованої моделі доцільно здійснювати за допомогою формалізованих метрик. До основних належать: Test Coverage (TC), що визначається як відношення кількості покритих вимог до їх загальної кількості; Defect Detection Rate (DDR), який відображає частку знайдених дефектів від загальної кількості; Mean Time to Test (MTTT), що вимірює середній час виконання тестів; та Flakiness Index (FI), який характеризує відсоток нестабільних тестів. Порівняння цих показників у класичній та гібридній моделі дозволяє зробити висновки про переваги застосованого підходу.

ISSN 2786-5460 (Print), ISSN 2786-5479 (Online)

Ефективність роботи гібридної моделі безпосередньо залежить від алгоритмічного забезпечення кожного її компонента. У модулі аналізу дефектів центральною задачею є обчислення метрик, що відображають ймовірність повторного виникнення помилок. Для цього використовується модель оцінки ризику, яка враховує не лише частоту появи дефектів, але й складність модулів програмного забезпечення, а також щільність змін у їхньому коді.

Визначений рівень ризику використовується для формування так званої матриці пріоритизації тестів, де кожному компоненту ставиться у відповідність кількість і тип сценаріїв, які мають бути згенеровані. Таким чином, якщо історично у модулі авторизації спостерігалася висока кількість критичних дефектів, система автоматично створюватиме додаткові негативні сценарії, орієнтовані на перевірку виняткових випадків.

У модулі генерації сценаріїв на основі LLM ключовим завданням є трансформація текстових специфікацій у формалізовані кроки тестування. Наприклад, для вимоги «користувач має можливість відновити пароль за допомогою електронної пошти» мовна модель здатна побудувати набір сценаріїв: від стандартного введення коректної адреси до некоректних даних, SQL-ін'єкцій та перевірки відсутності витоку інформації. На рис. 2 подано блок-схему, що відображає логіку виконання сценарію.

Рис. 2. Блок-схема сценарію скидання пароля з некоректною електронною адресою

Інтеграція моделі в CI/CD-середовище є важливою умовою її практичної реалізації. Типовий процес включає етапи: отримання останніх змін у коді, запуск модулю аналізу дефектів, формування ризикового профілю, генерацію сценаріїв за допомогою LLM та виконання тестів. Результати у вигляді звітів автоматично зберігаються для подальшого аналізу. Додатково система може адаптувати набір сценаріїв залежно від кількості ресурсів: наприклад, у випадку обмеженого часу виконуються лише тести з найвищим рівнем ризику.

Оцінювання ефективності гібридної моделі можливе через порівняння з класичними методами за низкою формальних метрик. Так, Test Coverage (TC) визначається за формулою:

$$TC = \frac{T_c}{T_t} \times 100\%, \tag{2}$$

де T_c – кількість покритих функціональних вимог, T_t – загальна кількість функціональних вимог.

Інша важлива метрика – Defect Detection Rate (DDR), яка обчислюється як

$$DDR = \frac{D_f}{D_t} \times 100\%, \tag{3}$$

де D_f – кількість знайдених дефектів, а D_t – їхня загальна кількість.

Для оцінки стабільності тестів використовується Flakiness Index, що визначається як відсоток нестабільних тестів серед усіх сценаріїв, які виконуються у середовищі CI/CD.

Висновки. У статті представлено концепцію гібридної моделі генерації тестових сценаріїв, що поєднує можливості великих мовних моделей та аналіз історії дефектів. Запропонований підхід забезпечує перехід від статичного тестування до динамічного та ризик-орієнтованого, де формування сценаріїв відбувається з урахуванням як поточних вимог, так і накопиченого досвіду попередніх збоїв.

Наукова новизна дослідження полягає у створенні моделі, здатної інтегрувати генеративні можливості LLM з механізмами обробки журналів виконання програмного забезпечення. Це дозволяє формувати більш релевантні й адаптивні сценарії, що орієнтуються на зони підвищеного ризику. Практична цінність полягає у можливості зниження витрат на підтримку автотестів, підвищення точності виявлення критичних помилок та інтеграції моделі у сучасні CI/CD-процеси.

Отримані результати свідчать про перспективність застосування гібридних підходів до автоматизованого тестування. Подальші дослідження доцільно спрямувати на розробку методів самоадаптації тестових сценаріїв при суттєвих змінах у програмних системах, а також на впровадження механізмів автоматичного відновлення тестів, що підвищить рівень автономності запропонованої моделі.

Список використаних джерел:

1. Alencar P., Cowan D., Lucena C., Lucena M. Log-based anomaly detection in software systems: A natural language processing approach. *Journal of Systems and Software,* 2020. 165, 110570. https://doi.org/10.1016/j.jss.2020.110570

2. Hellendoorn V. J. Large language models for software engineering: The next generation of tools. *Communications of the ACM,* 2023. 66(7), 34–36. https://doi.org/10.1145/3589300

3. Kiciński P., Dylong T. Applying ChatGPT in Software QA: A comparative study of manual and AI-generated test cases. *Proceedings of the 2024 IEEE International Conference on Software Quality, Reliability and Security (QRS).* 2024.

4. Li W., Zhao Y., Sun Q. Defect data-driven testing strategy optimization. *Software Quality Journal,* 2023. 31(2), 487–509.

5. Liu C., Wu J., Lu P. A hybrid framework for AI-assisted test case generation using bug reports. *IEEE Access,* 2023. 11, 90345–90358.

6. Panichella A., Kifetew F. M., Tonella P. Automated test case generation as a learning task. In *Proceedings of the 40th International Conference on Software Engineering (ICSE'18)* 2018. (pp. 1070–1081). New York, NY: ACM. https://doi.org/10.1145/3180155.3180204

7. Schäfer M., Nadi S., Eghbali A., Tip F. An empirical evaluation of using large language models for automated unit test generation. *IEEE Transactions on Software Engineering,* 2023. 50(1), 85–105. https://doi.org/10.1109/TSE.2023.3334955

8. Wang S., Chen T. Y., Harman M. Test case prioritization using machine learning: A systematic mapping study. *Information and Software Technology,* 2019. 93, 41–57. https://doi.org/10.1016/j.infsof.2017.09.002

9. Wang Y., Guo S., Tan C. W. From code generation to software testing: AI Copilot with context-based RAG. *IEEE Software.* 2025. https://doi.org/10.1109/MS.2025.3549628

10. Zhang J., Harman M., Ma L. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering,* 2020. 47(11), 2205–2231. https://doi.org/10.1109/TSE.2019.2962027

Дата надходження статті: 16.11.2025
Дата прийняття статті: 10.12.2025
Опубліковано: 30.12.2025