

ДОСЛІДЖЕННЯ ІНТЕГРАЦІЇ LLM МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ДО SDLC ПРОЦЕСУ

Вінницький національний технічний університет

Анотація

Розглянуто сучасний стан наукових досліджень у галузі використання великих мовних моделей (Large Language Models, LLM) у процесі розробки програмного забезпечення (Software Development Life Cycle, SDLC). Основну увагу приділено аналізу наукових публікацій, присвячених застосуванню LLM на окремих етапах життєвого циклу програмного забезпечення, зокрема під час аналізу вимог, програмування, тестування та супроводу. Виявлено основні переваги, обмеження та проблемні аспекти інтеграції мовних моделей у SDLC. Обґрунтовано доцільність подальших досліджень, спрямованих на розробку методологічних підходів до використання LLM у процесах програмної інженерії.

Ключові слова: великі мовні моделі, штучний інтелект, життєвий цикл розробки програмного забезпечення, програмна інженерія, автоматизація.

Abstract

The current state of scientific research in the field of using large language models (LLM) in the software development life cycle (SDLC) is considered. The main focus is on the analysis of scientific publications devoted to the application of LLM at certain stages of the software life cycle, in particular during requirements analysis, programming, testing, and maintenance. The main advantages, limitations and problematic aspects of integrating language models into SDLC are identified. The feasibility of further research aimed at developing methodological approaches to the use of LLM in software engineering processes is substantiated.

Keywords: large language models, artificial intelligence, software development life cycle, software engineering, automation.

Вступ

Процес розробки програмного забезпечення є формалізованою послідовністю етапів, об'єднаних у життєвий цикл розробки програмного забезпечення, що визначає порядок виконання робіт від формування вимог до супроводу програмного продукту. Ефективність реалізації кожного з етапів безпосередньо впливає на якість, надійність та підтримуваність програмних систем.

Застосування методів штучного інтелекту в програмній інженерії розглядається як один із напрямів підвищення рівня автоматизації процесів розробки програмного забезпечення. Великі мовні моделі є інструментом обробки та генерації текстової інформації, що потенційно може бути використано під час роботи з програмним кодом, технічною документацією та формалізованими описами вимог [1].

У зв'язку з цим виникає необхідність системного аналізу можливостей використання великих мовних моделей у процесі життєвого циклу розробки програмного забезпечення. Такий аналіз є підґрунтям для формування подальших наукових досліджень, спрямованих на розробку підходів до інтеграції мовних моделей у процеси програмної інженерії [2].

Постановка задачі дослідження

Метою дослідження є аналіз сучасних підходів до використання LLM-моделей у процесі SDLC, визначення перспектив їх подальшого застосування, розробка концептуальної моделі інтеграції та рекомендацій щодо практичного впровадження.

Завдання дослідження:

- провести аналіз існуючих методів та інструментів інтеграції штучного інтелекту у процеси розробки програмного забезпечення;
- дослідити можливості використання LLM для автоматизації різних етапів SDLC: планування, проектування, розробка, тестування, підтримка;
- розробити концептуальну модель інтеграції LLM моделей до SDLC, що забезпечує продуктивність та якість програмного забезпечення;
- провести оцінку ефективності використання LLM у розробці програмного забезпечення порівняно з традиційними методами;
- розробити рекомендації щодо практичного впровадження інтеграції LLM до SDLC.

Виклад основного матеріалу

Основна увага дослідження присвячена аналізу джерел інформації. Аналіз наукових джерел показує, що використання великих мовних моделей штучного інтелекту розглядається в контексті окремих етапів життєвого циклу розробки програмного забезпечення. Основна увага в літературі приділяється можливостям застосування мовних моделей для роботи з текстовими артефактами процесу SDLC, зокрема вимогами, програмним кодом та тестовою документацією.

У ряді робіт великі мовні моделі описуються як інструмент підтримки етапу програмування, де вони використовуються для генерації фрагментів коду, пояснення логіки програм та автоматизації рутинних операцій. Водночас наголошується, що результати роботи мовних моделей потребують додаткової перевірки, оскільки можливі логічні помилки та невідповідність вимогам програмного забезпечення [3-4].

Окрему групу досліджень присвячено застосуванню великих мовних моделей на етапі аналізу вимог. У таких роботах розглядаються можливості обробки природномовних специфікацій, узагальнення вимог та їх формалізації.

У межах етапу тестування програмного забезпечення великі мовні моделі розглядаються як засіб автоматизованої генерації тест-кейсів та сценаріїв перевірки. Аналіз літератури показує, що більшість таких підходів перебуває на стадії експериментального дослідження та потребує подальшої валідації в контексті реальних програмних систем.

Загалом огляд літератури свідчить про фрагментарний характер застосування великих мовних моделей у процесі життєвого циклу розробки програмного забезпечення. Відсутність системного підходу до їх використання в межах SDLC зумовлює необхідність подальших досліджень, спрямованих на аналіз можливостей та обмежень інтеграції мовних моделей у процеси програмної інженерії.

Висновки

Проведений аналіз джерел інформації свідчить про значний потенціал великих мовних моделей у процесі розробки програмного забезпечення. Використання LLM дозволяє автоматизувати низку рутинних операцій на різних етапах SDLC, зокрема генерування коду, аналіз вимог, автоматизацію тестування та підтримку документації. Водночас відсутність комплексних методологій та стандартизованих підходів до інтеграції LLM у життєвий цикл програмного забезпечення створює ризики помилок, невідповідності вимогам та потребу у додатковій перевірці результатів роботи моделей.

Отримані дані підкреслюють доцільність подальших досліджень, спрямованих на розробку методологічних підходів та концептуальних моделей інтеграції LLM у SDLC, що забезпечують підвищення продуктивності розробки, якості програмного продукту та ефективності командної роботи.

Висновки роботи можуть слугувати теоретичною та практичною основою для формування рекомендацій щодо впровадження штучного інтелекту у різні типи процесів розробки програмного забезпечення, а також для оцінки ефективності використання LLM порівняно з традиційними методами програмної інженерії.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Zhang Q., Fang C., Xie Y., Zhang Y., Yang Y., Sun W., Yu S., Chen Z. «A Survey on Large Language Models for Software Engineering». 2023. URL: <https://arxiv.org/abs/2312.15223>.
2. Fan A., Gokkaya B., Harman M., Lyubarskiy M., Sengupta S., Yoo S., Zhang J. M. «Large Language Models for Software Engineering: Survey and Open Problems». 2023. URL: <https://arxiv.org/abs/2310.03533>.
3. Uzay C., Tosun M. S., Karakuş O. «A Survey of Large Language Models: Evolution, Architectures, Adaptation, Benchmarking, Applications, Challenges, and Societal Implications». 2025. URL: <https://www.mdpi.com/2079-9292/14/18/3580>.
4. Dong Y., Jiang X., Qian J., Wang T., Zhang K., Jin Z., Li G. «A Survey on Code Generation with LLM based Agents». 2025. URL: <https://arxiv.org/abs/2508.00083>.

Іванов Данило Артемович – студент кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: rock2472467@gmail.com;

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Ivanov Danylo Artemovich – student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: rock2472467@gmail.com;

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua.