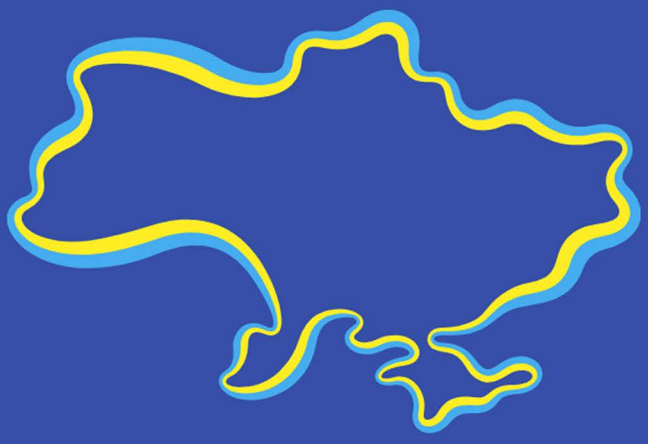
Наукові перспективи
Видавнича група

Наука і Техніка сьогодні

№ 4 (58)
2026

З Україною в серці!



Видавнича група «Наукові перспективи»

Всеукраїнська Асамблея докторів наук із державного управління

«Наука і техніка сьогодні»

Випуск № 4(58) 2026

Київ – 2026

Publishing Group «Scientific Perspectives»

Ukrainian Assembly of Doctors of Sciences in Public Administration

"Science and technology today"

Issue № 4(58) 2026

Kyiv – 2026

ISSN 2786-6025 Online

УДК 001.32:1 /3](477)(02)

R40-05553

DOI: **https://doi.org/10.52058/2786-6025-2026-4(58)**

«Наука і техніка сьогодні» (Серія «Педагогіка», Серія «Право», Серія «Економіка», Серія «Фізико-математичні науки», Серія «Техніка»)»: журнал. 2026. № 4(58) 2026. С. 5059

Згідно наказу Міністерства освіти і науки України від 07.04.2022 № 320 журналу присвоєно категорію "Б" із економіки та педагогіки (спеціальності – 015 - Педагогічні науки; 076 - Економічні науки)

Згідно наказу Міністерства освіти і науки України від 06.06.2022 № 530 журналу присвоєно категорію "Б" із права (спеціальність – 081 Юридичні науки)

Згідно наказу Міністерства освіти і науки України від 10.10.2022 № 894 журналу присвоєно категорію "Б" із техніки (спеціальність - 122 Комп'ютерні науки)

Журнал видається за підтримки Міждержавної гільдії інженерів консультантів, Інституту філософії та соціології Національної Академії Наук Азербайджану (Баку, Азербайджан), громадської організації «Християнська академія педагогічних наук України» та громадської організації «Всеукраїнська асоціація педагогів і психологів з духовно-морального виховання»

Рекомендовано до видавництва Президією Всеукраїнської Асамблеї докторів наук з державного управління (Рішення від 24.04.2026, № 8/4-26)

Журнал включено до міжнародної наукометричної бази Index Copernicus (IC), міжнародної пошукової системи Google Scholar та до міжнародної наукометричної бази даних Research Bible

Згідно Порядку формування Переліку наукових фахових видань України, затвердженого наказом МОН України від 15.01.2018 № 32, повнотекстовий доступ до наукових статей журналу представлений на платформі «Наукова періодика України» в Національній бібліотеці України імені В.І. Вернадського НАН України та в Національному репозитарії академічних текстів

Головний редактор:

Коренева Інна Миколаївна *- доктор педагогічних наук, професор, декан факультету природничої і фізико-математичної освіти Глухівського національного педагогічного університету імені Олександра Довженка; професор кафедри теорії і методики викладання природничих дисциплін Глухівського національного педагогічного університету імені Олександра Довженка (Україна)*

Редакційна колегія:

1. **Біляковська Ольга Орестівна** доктор педагогічних наук, професор, завідувачка кафедри загальної педагогіки та педагогіки вищої школи Львівського національного університету імені Івана Франка (Україна)
2. **Воровка Маргарита Іванівна** – докторка педагогічних наук, професорка, професорка кафедри освітології та педагогіки мистецтва Мелітопольського державного педагогічного університету імені Богдана Хмельницького (Україна)

3. **Гончарук Валентина Анатоліївна** - кандидат педагогічних наук, доцент, доцент кафедри української літератури, українознавства та методик їх навчання Уманського державного педагогічного університету імені Павла Тичини (Україна)
4. **Гончарук Віталій Володимирович** – кандидат педагогічних наук, доцент кафедри хімії та екології Уманського державного педагогічного університету імені Павла Тичини (Україна)
5. **Гуменюк Тетяна Костянтинівна** - доктор філософських наук, Заслужений працівник освіти України, професор, проректор з науково-педагогічної роботи, інноваційно-методичного забезпечення освітнього та наукового процесів Київської муніципальної академії музики ім. Р.М. Глієра (Україна)
6. **Депчинська Іветта Аттілівна** - кандидат педагогічних наук, доцент, доцент кафедри педагогіки, психології, початкової, дошкільної освіти та управління закладом освіти, Закарпатський угорський університет ім. Ференца Ракоці II (Україна)
7. **Мутмайна** - викладач Університету Аль Асярія Мандар Сулавесі Барат, Індонезія, ад'юнкт-професор Департаменту освіти, Університет Manipal GlobalNxt Малайзії (Малазія)
8. **Кожевникова Алла Власівна** - доцент кафедри освітології та педагогіки мистецтва МДПУ імені Богдана Хмельницького (Україна)
9. **Кравчук Людмила Степанівна** - кандидат педагогічних наук, доцент, професор кафедри фізичної терапії, ерготерапії, фізичної культури і спорту Хмельницького інституту соціальних технологій Університету «Україна», завідувач кафедрою фізичної терапії, ерготерапії, фізичної культури і спорту Хмельницького інститут соціальних технологій Університет «Україна» (Україна)
10. **Красницька Ольга Володимирівна** - кандидат педагогічних наук, доцент, доцент кафедри суспільних наук Національного університету оборони України (Україна)
11. **Марчук Оксана Олександрівна** - доктор педагогічних наук, професор, професор кафедри загальної педагогіки та дошкільної освіти ПВНЗ «Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука» (Україна)
12. **Небеленчук Ірина Олександрівна** - доктор педагогічних наук, старший викладач кафедри теорії і методики середньої освіти комунального закладу «Кіровоградський обласний інститут післядипломної педагогічної освіти імені Василя Сухомлинського» (Україна)
13. **Островська Маріанна Ярославівна** -доктор педагогічних наук, професор кафедри педагогіки, психології, початкової, дошкільної освіти та управління закладом освіти Закарпатського угорського університету імені Ференца Ракоці II (Україна)
14. **Р. Ахмад Закі Ель Ісламі** - доцент, професор, доктор філософії, Департамент наукової освіти, Факультет підготовки вчителів та освіти, Університет Султана Агенга Тіртаяса (Індонезія)
15. **Тавдгіріdзе Лела** - Доцент з теорії та історії педагогіки, професор кафедри педагогічних наук Батумського державного університету ім. Шота Руставелі (Грузія)
16. **Шевчук Лариса Дмитрівна** - доктор педагогічних наук, професор, завідувач кафедри математики, інформатики і методики навчання Університету Григорія Сковороди в Переяславі (Україна)

Статті розміщені в авторській редакції. Відповідальність за зміст та орфографію поданих матеріалів несуть автори

© автори статей, 2026
© Видавнича група «Наукові перспективи», 2026

№ 4(58)
2026
НАУКА і ТЕХНІКА СЬОГОДНІ
ISSN 2786-6025 Online

Денисюк В.О., Іванов Д.А. **3183**
АНАЛІЗ ПІДХОДІВ ДО ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ В МЕЖАХ ЖИТТЄВОГО ЦИКЛУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Дєгтяр О.А., Карабань О.В. **3196**
РОЛЬ ДЕРЖАВ У СУЧАСНИХ КІБЕРОПЕРАЦІЯХ

Дитко Т.В. **3209**
МОДЕЛЮВАННЯ УДОСКОНАЛЕНОГО АЛГОРИТМУ ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ КОГНІТИВНИХ ПРОЦЕСІВ

Дідківський А.А., Сілагін О.В. **3220**
ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ ДЛЯ РОЗПІЗНАВАННЯ ДЕРМАТОЛОГІЧНИХ ЗАХВОРЮВАНЬ НА ЗОБРАЖЕННІ

Домашенко Д.Г., Гоменюк С.І., Домашенко С.В., Морозов Д.М. **3233**
АРХІТЕКТУРНІ ОБМЕЖЕННЯ МОНОЛІТНИХ СИСТЕМ ПРОГНОЗНОЇ HR-АНАЛІТИКИ ТА МІКРОСЕРВІСНИЙ ПІДХІД ДО ЇХ ПОДОЛАННЯ

Домніч Д., Урняєва І., Чайковська О. **3245**
АРХІТЕКТУРНІ РІШЕННЯ ДЛЯ СТВОРЕННЯ AI-POWERED ВЕБ-ЗАСТОСУНКІВ НА ОСНОВІ МОВНИХ МОДЕЛЕЙ

Єшанов М.О., Поперешняк С.В. **3260**
ПІДХІД ДО АДАПТИВНОГО КЕРУВАННЯ ПАРАМЕТРАМИ СТИСКАННЯ В MPS-ОРІЄНТОВАНІЙ СИМУЛЯЦІЇ КВАНТОВИХ СХЕМ

Жмурко Ю.А., Юрченко Ю.Ю., Самойленко Г.Т. **3274**
ПОРІВНЯЛЬНИЙ АНАЛІЗ РІВНІВ RAID У КОНТЕКСТІ РОЗПОДІЛЕНИХ СЕРВЕРНИХ СИСТЕМ

Зелений В.Є., Козловський А.В. **3287**
ПРОСТОРОВО-АДАПТИВНЕ ЗЛИТТЯ ПСЕВДОМІТОК З ПОПІКСЕЛЬНИМИ КАРТАМИ НЕВИЗНАЧЕНОСТІ ДЛЯ СЛАБОКОНТРОЛЬОВАНОЇ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ НА ОСНОВІ БАЗОВИХ МОДЕЛЕЙ

№ 4(58) 2026
НАУКА і ТЕХНІКА СЬОГОДНІ
ISSN 2786-6025 Online
УДК 004.8
https://doi.org/10.52058/2786-6025-2026-4(58)-3183-3195

Денисюк Валерій Олександрович кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, https://orcid.org/0000-0003-1057-3518

Іванов Данило Артемович аспірант кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, https://orcid.org/0009-0001-9431-3467

АНАЛІЗ ПІДХОДІВ ДО ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ В МЕЖАХ ЖИТТЄВОГО ЦИКЛУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Анотація. У статті розглянуто сучасні підходи до використання великих мовних моделей (Large Language Models, LLM) у межах життєвого циклу розробки програмного забезпечення (SDLC). Проаналізовано основні етапи SDLC та визначено роль LLM у кожному з них, зокрема під час аналізу вимог, проєктування, реалізації, тестування та супроводу програмних систем. Узагальнено наукові дослідження щодо застосування мовних моделей для автоматизації рутинних завдань програмної інженерії, генерації коду, створення тестової документації, аналізу технічних артефактів, а також підтримки прийняття рішень у складних інженерних процесах.

Запропоновано узагальнену класифікацію підходів до інтеграції LLM у процес SDLC, яка включає використання моделей як інструментів підтримки окремих операцій, асистентів розробників, автономних агентів та компонентів програмної інженерії.

Такий підхід дозволяє систематизувати існуючі наукові напрацювання та виділити різні рівні залучення мовних моделей у процеси розробки програмного забезпечення, що відрізняються ступенем автономності, складністю взаємодії та глибиною інтеграції в інженерні середовища.

Окрему увагу приділено перевагам і обмеженням застосування LLM, зокрема питанням якості результатів, контекстної обмеженості, інтерпретації предметної області, можливим помилкам генерації, а також аспектам інформаційної безпеки та відповідальності під час використання таких систем у реальних проєктах програмної інженерії. Додатково підкреслено необхідність контролю результатів з боку фахівців та використання моделей у допоміжній ролі.

ISSN 2786-6025 Online

Результати дослідження свідчать про значний потенціал великих мовних моделей для підвищення ефективності процесів розробки програмного забезпечення, зменшення трудомісткості рутинних операцій, пришвидшення аналізу даних та оптимізації інженерних підходів у цій галузі. Водночас підкреслюється важливість подальших досліджень, спрямованих на розробку формалізованих методологій інтеграції LLM у SDLC, удосконалення механізмів контролю якості та підвищення надійності отриманих результатів.

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

Denysiuk Valerii Olexandrovich Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, https://orcid.org/0000-0003-1057-3518

Ivanov Danylo Artemovych Graduate Student Department of Computer Science, Vinnitsa National Technical University, Vinnytsia, https://orcid.org/0009-0001-9431-3467

ANALYSIS OF APPROACHES TO THE USE OF LARGE LANGUAGE MODELS WITHIN THE SOFTWARE DEVELOPMENT LIFE CYCLE

Abstract. This article examines current approaches to the use of Large Language Models (LLMs) within the software development life cycle (SDLC). The main stages of the SDLC are analyzed, and the role of LLMs in each of them is identified, particularly during requirements analysis, design, implementation, testing, and maintenance of software systems. Scientific research on the application of language models for automating routine software engineering tasks, code generation, the creation of test documentation, the analysis of technical artifacts, as well as decision support in complex engineering processes.

A generalized classification of approaches to integrating LLMs into the SDLC is proposed, which includes the use of models as tools to support individual operations, developer assistants, autonomous agents, and software engineering components. This approach allows for the systematization of existing scientific research and the identification of different levels of language model involvement in software development processes, which differ in terms of the degree of autonomy, the complexity of interaction, and the depth of integration into engineering environments.

Particular attention is paid to the advantages and limitations of using LLMs, specifically regarding the quality of results, contextual limitations, interpretation of the subject domain, potential generation errors, as well as aspects of information

ISSN 2786-6025 Online
security and accountability when using such systems in real-world software engineering projects. Additionally, the need for experts to review the results and for the models to be used in a supporting role is emphasized.

The research results indicate the significant potential of large language models for improving the efficiency of software development processes, reducing the labor intensity of routine operations, accelerating data analysis, and optimizing engineering approaches in this field. At the same time, the importance of further research aimed at developing formalized methodologies for integrating LLMs into the SDLC is emphasized, improving quality control mechanisms, and increasing the reliability of the results obtained.

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

Постановка проблеми. Процес розробки програмного забезпечення є складною багаторівневою діяльністю, що охоплює сукупність взаємопов'язаних етапів, об'єднаних у життєвий цикл розробки програмного забезпечення (Software Development Life Cycle, SDLC). До основних етапів SDLC належать формування та аналіз вимог, проєктування системи, реалізація програмного коду, тестування, розгортання та подальший супровід програмного продукту. Якість виконання кожного з цих етапів безпосередньо впливає на надійність, функціональність, масштабованість та підтримуваність програмних систем.

Сучасні програмні продукти характеризуються зростанням складності архітектури, обсягів програмного коду та вимог до швидкості розробки. Це зумовлює необхідність підвищення рівня автоматизації процесів програмної інженерії та використання інтелектуальних методів підтримки прийняття рішень [1]. Одним із перспективних напрямів у цьому контексті є застосування методів штучного інтелекту, зокрема великих мовних моделей (Large Language Models, LLM).

Великі мовні моделі є класом моделей штучного інтелекту, здатних до обробки, аналізу та генерації текстової інформації на основі великих обсягів навчальних даних. Їх використання вже продемонструвало ефективність у таких завданнях, як аналіз природної мови, автоматичне створення текстів, переклад та узагальнення інформації. У межах програмної інженерії LLM привертають значну увагу завдяки можливості працювати з текстовими артефактами процесу SDLC, зокрема вимогами, програмним кодом, тестовою документацією та звітами про дефекти.

У зв'язку з цим актуальним є проведення оглядового дослідження, спрямованого на систематизацію наукових підходів до застосування великих мовних моделей у процесі SDLC. Такий огляд дозволяє узагальнити наявні

ISSN 2786-6025 Online

результати, класифікувати існуючі підходи до інтеграції LLM, виявити основні проблеми та визначити напрями подальших досліджень у галузі програмної інженерії.

Аналіз останніх досліджень і публікацій – свідчить про зростання кількості досліджень, присвячених використанню LLM на окремих етапах життєвого циклу програмного забезпечення [2]. Найбільшу увагу приділено застосуванню мовних моделей під час програмування, де вони використовуються для генерації фрагментів коду, пояснення логіки програм та автоматизації рутинних операцій.

Окремі роботи також розглядають можливості використання LLM для аналізу вимог, автоматизованої генерації тест-кейсів та підтримки супроводу програмних систем.

Водночас більшість існуючих досліджень має фрагментарний характер та зосереджується на ізольованому використанні LLM у межах окремих етапів SDLC. Відсутність узагальненого бачення інтеграції великих мовних моделей у життєвий цикл розробки програмного забезпечення ускладнює оцінку їх реального впливу на ефективність процесів програмної інженерії. Крім того, залишаються недостатньо дослідженими питання обмежень, ризиків та методологічних аспектів використання LLM у комплексних процесах розробки програмного забезпечення.

Мета статті – огляд та аналіз сучасних наукових досліджень, присвячених інтеграції великих мовних моделей штучного інтелекту до процесу життєвого циклу розробки програмного забезпечення, а також систематизація підходів до їх використання на різних етапах SDLC.

Для досягнення поставленої мети у статті передбачається розв’язання таких завдань:

- проаналізувати наукові публікації, присвячені використанню LLM у процесах програмної інженерії;
- дослідити можливості застосування великих мовних моделей на окремих етапах SDLC;
- узагальнити та класифікувати підходи до інтеграції LLM у життєвий цикл розробки програмного забезпечення;
- визначити основні проблеми та обмеження використання мовних моделей у процесі SDLC;
- окреслити перспективні напрями подальших досліджень у цій галузі.

Виклад основного матеріалу. У роботі використано підхід оглядового дослідження, спрямований на систематизацію та аналіз наукових публікацій, присвячених використанню великих мовних моделей у процесі життєвого циклу розробки програмного забезпечення. Методологічною основою є якісний

ISSN 2786-6025 Online

аналіз літературних джерел із подальшим узагальненням та класифікацією підходів до інтеграції LLM.

Під час проведення оглядового дослідження було використано наукові публікації, представлені у відкритих електронних базах даних та наукових репозитаріях, зокрема матеріали міжнародних наукових журналів і конференцій, а також препринти, розміщені у відкритому доступі. Основну увагу було зосереджено на роботах, опублікованих у період активного розвитку великих мовних моделей, що дозволило охопити сучасний стан досліджень у даній галузі.

Відбір джерел здійснювався на основі таких критеріїв: відповідність тематиці використання LLM у програмній інженерії, наявність чіткого опису застосування мовних моделей у межах окремих етапів SDLC, а також наукова обґрунтованість отриманих результатів. До аналізу включалися роботи, у яких розглядалися як теоретичні аспекти використання LLM, так і прикладні сценарії їх застосування у процесах розробки програмного забезпечення.

Метод аналізу літератури передбачав поетапне вивчення джерел із подальшим групуванням публікацій за напрямами застосування великих мовних моделей.

Зокрема, усі проаналізовані роботи було класифіковано відповідно до етапів життєвого циклу розробки програмного забезпечення, на яких пропонується використання LLM, а також за типом виконуваних завдань, таких як аналіз вимог, генерація програмного коду, автоматизація тестування та підтримка супроводу програмних систем.

Для узагальнення результатів було застосовано метод порівняльного аналізу, що дозволив виявити спільні риси та відмінності між існуючими підходами, а також визначити їх переваги й обмеження. Окрему увагу приділено ідентифікації проблемних аспектів використання великих мовних моделей у процесі SDLC, зокрема питань якості результатів, надійності, валідації та безпеки.

Результати оглядового дослідження систематизовано у вигляді тематичних розділів, що відображають застосування LLM на різних етапах життєвого циклу програмного забезпечення, а також у вигляді узагальненої класифікації підходів до інтеграції мовних моделей у процес SDLC. Такий підхід дозволяє сформувати цілісне уявлення про сучасний стан досліджень у даній галузі та визначити напрями подальшого наукового пошуку.

Етап аналізу та формування вимог є одним із найбільш критичних у життєвому циклі розробки програмного забезпечення, оскільки саме на цьому етапі закладаються функціональні та нефункціональні характеристики майбутньої системи. Помилки або неоднозначності у вимогах можуть призводити до значних витрат часу й ресурсів на подальших етапах SDLC, а

ISSN 2786-6025 Online

також до зниження якості кінцевого програмного продукту. У науковій літературі LLM розглядаються як інструмент для автоматизованої обробки природномовних вимог, зокрема для їх узагальнення, структурування та класифікації. Завдяки здатності працювати з великими обсягами текстової інформації мовні моделі можуть бути використані для виділення ключових функціональних вимог, виявлення повторів та логічних суперечностей, а також для формування узгоджених описів системи у вигляді user stories або технічних специфікацій.

Окремі дослідження присвячені використанню LLM для трансформації неструктурованих вимог у напівформальні або формалізовані представлення, придатні для подальшого аналізу. У таких підходах мовні моделі застосовуються для перетворення текстових вимог у структуровані шаблони, сценарії використання або обмеження, що дозволяє підвищити трасованість вимог у межах SDLC. Це, у свою чергу, сприяє кращому зв'язку між етапами аналізу вимог, проєктування та тестування.

Разом із тим у літературі наголошується, що використання великих мовних моделей на етапі аналізу вимог має низку обмежень. Зокрема, результати автоматизованого аналізу можуть містити неточності, пов'язані з неоднозначністю природної мови або неповнотою вхідних даних. Крім того, мовні моделі не завжди здатні коректно інтерпретувати контекст предметної області без додаткових знань або залучення експертів [4].

Таким чином, аналіз наукових публікацій свідчить, що LLM на етапі аналізу вимог доцільно розглядати як інструмент підтримки аналітиків, а не як повністю автономне рішення. Їх використання може сприяти зниженню навантаження на фахівців, підвищенню узгодженості вимог та покращенню якості початкових специфікацій, однак потребує обов'язкової перевірки результатів з боку людини.

Етап проєктування системи відіграє ключову роль у забезпеченні масштабованості, надійності та підтримуваності програмного забезпечення. На цьому етапі формуються архітектурні рішення, визначається структура компонентів, їх взаємодія та основні принципи реалізації системи. Помилки, допущені під час проєктування, часто мають системний характер і складно усуваються на пізніших етапах SDLC.

У наукових дослідженнях великі мовні моделі розглядаються як інструмент інтелектуальної підтримки процесу проєктування. Зокрема, LLM можуть бути використані для аналізу архітектурних вимог, генерації описів архітектурних шаблонів та рекомендацій щодо вибору технологічних рішень. Завдяки здатності узагальнювати інформацію з великої кількості джерел мовні моделі можуть пропонувати типові архітектурні підходи, що відповідають заданим вимогам і обмеженням.

ISSN 2786-6025 Online

Окрему увагу в літературі приділено можливості використання LLM для автоматизованого створення текстових описів архітектури системи, а також допоміжних діаграм, які відображають взаємодію між компонентами. Такі підходи можуть сприяти підвищенню узгодженості проєктної документації та спрощенню комунікації між учасниками команди розробки [5].

Водночас наукові публікації підкреслюють, що мовні моделі не здатні повністю замінити архітектора програмного забезпечення, оскільки не враховують усі контекстні фактори конкретного проєкту. Тому їх застосування на етапі проєктування доцільно розглядати як допоміжний інструмент для генерації альтернативних рішень і попереднього аналізу, результати якого мають бути перевірені фахівцями.

Найбільш активно великі мовні моделі застосовуються на етапі реалізації програмного забезпечення. У сучасних дослідженнях LLM розглядаються як інструмент підтримки програмістів під час написання, аналізу та модифікації програмного коду. Основна увага приділяється можливостям автоматизованої генерації фрагментів коду на основі текстових описів задач або коментарів [3].

У літературі відзначається, що використання LLM може суттєво зменшити час виконання рутинних операцій, пов’язаних із реалізацією стандартних алгоритмів або типових конструкцій. Крім того, мовні моделі застосовуються для пояснення логіки існуючого коду, що є корисним у процесі супроводу або під час роботи з великими кодовими базами.

Разом із перевагами дослідники звертають увагу на обмеження застосування LLM у програмуванні. Зокрема, згенерований код може містити логічні помилки, не відповідати нефункціональним вимогам або порушувати правила безпеки. Це зумовлює необхідність ретельної перевірки та тестування результатів, отриманих за допомогою мовних моделей.

Таким чином, у наукових роботах підкреслюється, що великі мовні моделі доцільно використовувати як асистентів розробника, які підтримують процес програмування, але не замінюють повноцінного контролю з боку людини.

Етап тестування спрямований на виявлення дефектів, перевірку відповідності програмного забезпечення вимогам та оцінку його якості. У контексті використання великих мовних моделей наукові дослідження зосереджуються на можливостях автоматизації процесів створення тестової документації та аналізу результатів тестування.

У літературі розглядається застосування LLM для автоматизованої генерації тест-кейсів на основі вимог або програмного коду. Такі підходи дозволяють зменшити навантаження на фахівців з тестування та прискорити процес перевірки програмного продукту. Крім того, мовні моделі можуть бути використані для аналізу логів та звітів про помилки з метою класифікації дефектів і визначення їх можливих причин.

ISSN 2786-6025 Online

Разом із тим більшість досліджень наголошує на експериментальному характері таких рішень. Генеровані тест-кейси не завжди охоплюють усі критичні сценарії, а результати аналізу дефектів можуть потребувати додаткової інтерпретації. У зв'язку з цим використання LLM на етапі тестування розглядається як допоміжний інструмент, що підвищує ефективність роботи QA-фахівців, але не замінює традиційні методи забезпечення якості.

Етап супроводу програмного забезпечення охоплює діяльність, пов'язану з виправленням помилок, оновленням функціональності та адаптацією системи до змін у середовищі експлуатації. У цьому контексті великі мовні моделі розглядаються як засіб підтримки аналізу інцидентів і взаємодії з користувачами.

Наукові публікації відзначають можливість застосування LLM для автоматизованої обробки звернень користувачів, узагальнення інформації про проблеми та формування рекомендацій щодо їх усунення. Крім того, мовні моделі можуть бути використані для підтримки актуальності технічної документації, що є важливим аспектом довготривалого супроводу програмних систем.

Водночас дослідники звертають увагу на обмеження таких підходів, пов'язані з необхідністю забезпечення коректності та безпеки оброблюваних даних. Це особливо актуально для систем, що працюють з конфіденційною інформацією. У зв'язку з цим застосування великих мовних моделей на етапі супроводу потребує чітко визначених правил використання та контролю результатів.

Аналіз наукових публікацій, присвячених використанню великих мовних моделей у програмній інженерії, свідчить про різноманіття підходів до їх інтеграції у процес життєвого циклу розробки програмного забезпечення. Водночас відсутність єдиної узагальненої класифікації ускладнює порівняльний аналіз існуючих рішень та оцінку їх впливу на ефективність процесів SDLC. У зв'язку з цим доцільним є формування класифікації підходів до інтеграції LLM на основі аналізу функціональної ролі мовних моделей у процесі розробки програмного забезпечення.

На основі проведеного огляду літератури можна виділити кілька основних підходів до інтеграції великих мовних моделей у SDLC, які відрізняються рівнем залученості LLM у процеси програмної інженерії та ступенем їх автономності.

LLM як інструмент підтримки окремих операцій. У межах цього підходу великі мовні моделі використовуються як допоміжні інструменти для виконання окремих операцій на певних етапах SDLC. Найчастіше такі моделі застосовуються для генерації текстових фрагментів, пояснення-генерації програмного коду або створення тестової документації. Інтеграція LLM у

ISSN 2786-6025 Online

цьому випадку має локальний характер і не передбачає глибокого впливу на загальну структуру процесу розробки [7].

Основною перевагою такого підходу є простота впровадження та мінімальні вимоги до зміни існуючих процесів програмної інженерії. Водночас обмеженням є фрагментарний характер використання мовних моделей, що не дозволяє повною мірою реалізувати їх потенціал у межах усього життєвого циклу розробки програмного забезпечення.

LLM як інтелектуальний асистент учасників процесу SDLC. Другий підхід передбачає використання великих мовних моделей як асистентів для фахівців, залучених до різних етапів SDLC, зокрема аналітиків, архітекторів, розробників та тестувальників. У цьому випадку LLM взаємодіють з користувачами у діалоговому режимі, надаючи рекомендації, пропонуючи альтернативні рішення та допомагаючи в аналізі інформації.

Такий підхід забезпечує більш тісну інтеграцію мовних моделей у процеси програмної інженерії та сприяє підвищенню продуктивності роботи команд розробки. Водночас ефективність використання LLM як асистентів значною мірою залежить від кваліфікації користувачів та їх здатності критично оцінювати результати, згенеровані моделлю.

LLM як автономні або напівавтономні агенти. У межах цього підходу великі мовні моделі розглядаються як агенти, здатні виконувати послідовності дій у межах окремих етапів SDLC або навіть між ними. Такі агенти можуть самостійно аналізувати вхідні дані, генерувати артефакти розробки та передавати результати іншим компонентам системи.

Незважаючи на високий потенціал автоматизації, застосування LLM як автономних агентів супроводжується значними ризиками, пов’язаними з контролем якості, передбачуваністю поведінки моделей та відповідальністю за прийняті рішення. У зв’язку з цим більшість наукових досліджень розглядає такі підходи як експериментальні та такі, що потребують подальшої валідації.

LLM як компонент процесу SDLC. Найбільш комплексний підхід передбачає інтеграцію великих мовних моделей безпосередньо у структуру процесу SDLC як одного з його функціональних компонентів. У цьому випадку LLM взаємодіють з усіма етапами життєвого циклу, забезпечуючи безперервний обмін інформацією, накопичення знань та підтримку прийняття рішень.

Такий підхід дозволяє розглядати мовні моделі не як ізольований інструмент, а як системний елемент програмної інженерії. Водночас його реалізація потребує розробки чітких методологічних підходів, стандартів інтеграції та механізмів контролю результатів роботи LLM.

Незважаючи на значний потенціал великих мовних моделей для автоматизації та оптимізації процесів програмної інженерії, їх інтеграція у життєвий цикл розробки програмного забезпечення супроводжується низкою

ISSN 2786-6025 Online

проблем і обмежень. Аналіз наукових джерел свідчить про те, що більшість існуючих рішень перебуває на стадії експериментального впровадження, а питання їх практичного використання потребують додаткового дослідження [8].

Проблеми якості та достовірності результатів. Однією з ключових проблем використання великих мовних моделей є відсутність гарантій щодо коректності згенерованих результатів. LLM можуть створювати тексти або програмний код, що виглядають логічно обґрунтованими, проте містять концептуальні або логічні помилки. У контексті SDLC це може призводити до неправильного формулювання вимог, некоректних архітектурних рішень або помилок у програмному коді.

У наукових публікаціях наголошується, що мовні моделі не мають власного розуміння предметної області, а лише відтворюють статистичні закономірності на основі навчальних даних. Це зумовлює необхідність обов'язкової перевірки результатів, отриманих за допомогою LLM, з боку фахівців.

Обмеження контекстної обізнаності. Ще одним суттєвим обмеженням є складність повного врахування контексту конкретного програмного проєкту. Великі мовні моделі зазвичай працюють із обмеженим обсягом вхідної інформації та не завжди мають доступ до всіх артефактів SDLC, таких як історія змін, внутрішні стандарти або специфічні нефункціональні вимоги.

У результаті рекомендації або рішення, згенеровані LLM, можуть не відповідати реальним умовам розробки. Це особливо критично для великих та складних програмних систем, де контекст відіграє вирішальну роль у прийнятті інженерних рішень.

Питання безпеки та конфіденційності даних. Інтеграція великих мовних моделей у процеси SDLC пов'язана з обробкою значних обсягів даних, серед яких можуть бути конфіденційні відомості, вихідний код або комерційна інформація. Передача таких даних до зовнішніх сервісів або моделей може створювати ризики витоку інформації та порушення вимог інформаційної безпеки.

У науковій літературі підкреслюється необхідність розробки механізмів захисту даних, зокрема використання локальних моделей, анонімізації інформації та чітких політик доступу. Без вирішення цих питань широке впровадження LLM у корпоративні SDLC-процеси є обмеженим.

Проблеми відповідальності та контролю рішень. Застосування LLM як автономних або напівавтономних агентів у процесі розробки програмного забезпечення породжує питання відповідальності за прийняті рішення. У разі виникнення помилок або дефектів складно визначити, чи є вони результатом дій розробника, чи наслідком використання мовної моделі.

ISSN 2786-6025 Online

Це ускладнює процеси управління якістю та відповідальності у команді розробки. У зв'язку з цим більшість дослідників сходяться на думці, що LLM мають використовуватися в рамках підходу «людина в контурі», де остаточні рішення залишаються за фахівцем.

Проведений аналіз свідчить, що проблеми та обмеження використання великих мовних моделей у SDLC мають як технічний, так і організаційний характер. Їх наявність не зменшує потенціалу LLM, проте вимагає обережного та обґрунтованого підходу до інтеграції мовних моделей у процеси програмної інженерії.

Одним із перспективних напрямів подальших досліджень є розробка формалізованих методологій використання великих мовних моделей у межах усього життєвого циклу програмного забезпечення. Такі методології мають враховувати специфіку окремих етапів SDLC, роль учасників процесу та вимоги до якості результатів. Особливу увагу доцільно приділяти визначенню меж автономності LLM та механізмів контролю їх роботи.

Іншим важливим напрямом є дослідження підходів до інтеграції мовних моделей з існуючими інструментами програмної інженерії, зокрема системами управління вимогами, середовищами розробки, системами контролю версій та платформами автоматизованого тестування. Така інтеграція може сприяти створенню єдиного інтелектуального середовища підтримки SDLC.

Окремої уваги потребують питання оцінювання ефективності використання великих мовних моделей у процесі розробки програмного забезпечення. Подальші дослідження можуть бути спрямовані на розробку метрик і критеріїв, які дозволять кількісно оцінювати вплив LLM на продуктивність, якість та надійність програмних систем [6].

Крім того, перспективним є напрям дослідження етичних та правових аспектів використання великих мовних моделей у програмній інженерії. Зокрема, це стосується питань відповідальності за результати роботи LLM, захисту інтелектуальної власності та дотримання вимог інформаційної безпеки.

Висновоки. Проведено аналіз сучасних підходів до інтеграції великих мовних моделей штучного інтелекту у процес життєвого циклу розробки програмного забезпечення.

Розглянуто основні напрями використання LLM на різних етапах SDLC, зокрема під час аналізу вимог, проєктування, реалізації, тестування та супроводу програмних систем.

У результаті огляду літератури встановлено, що великі мовні моделі мають значний потенціал для підвищення ефективності процесів програмної інженерії, зменшення навантаження на фахівців та автоматизації рутинних операцій. Водночас більшість існуючих рішень має фрагментарний характер і зосереджується на окремих етапах SDLC.

ISSN 2786-6025 Online

Запропоновано класифікацію підходів до інтеграції LLM у процес SDLC, що дозволяє систематизувати наявні дослідження та визначити рівні залученості мовних моделей у процеси розробки програмного забезпечення. Проведений аналіз проблем і обмежень показав, що ефективне використання LLM потребує обов'язкового контролю з боку людини, а також врахування питань якості, безпеки та відповідальності.

Отримані результати підтверджують актуальність подальших досліджень у напрямі розробки комплексних методологій інтеграції великих мовних моделей у SDLC, що може стати основою для формування нових підходів до програмної інженерії в умовах активного розвитку технологій штучного інтелекту.

Література:

1. Zhang Q., Fang C., Xie Y. та ін. A survey on large language models for software engineering. 2023. URL: https://arxiv.org/abs/2312.15223

2. Hou X., Zhao Y., Liu Y. та ін. Large language models for software engineering: a systematic literature review. 2023. URL: https://arxiv.org/abs/2308.10620

3. Dong Y., Jiang X., Qian J. та ін. A survey on code generation with LLM-based agents. 2025. URL: https://arxiv.org/abs/2508.00083

4. Zheng Z., Ning K., Wang Y. та ін. A survey of large language models for code: evolution, benchmarking, and future trends. 2023.

5. Dhanuka D. Impact of LLMs on team collaboration in software development. 2025. URL: https://arxiv.org/abs/2510.08612

6. Wang K., Li T., Zhang X. та ін. Software development life cycle perspective: a survey of benchmarks for CodeLLMs and agents. 2025. URL: https://arxiv.org/abs/2505.05283

7. Joel S., Wu J. W., Fard F. H. A survey on LLM-based code generation for low-resource and domain-specific programming languages. 2024. URL: https://arxiv.org/abs/2410.03981

8. Mateus de Brito V., Farias K. Understanding the role of large language models in software engineering: evidence from an industry survey. 2025. URL: https://arxiv.org/abs/2512.21347

References:

1. Zhang Q., Fang C., Xie Y. та ін. A survey on large language models for software engineering. 2023. URL: https://arxiv.org/abs/2312.15223

2. Hou X., Zhao Y., Liu Y. та ін. Large language models for software engineering: a systematic literature review. 2023. URL: https://arxiv.org/abs/2308.10620

3. Dong Y., Jiang X., Qian J. та ін. A survey on code generation with LLM-based agents. 2025. URL: https://arxiv.org/abs/2508.00083

4. Zheng Z., Ning K., Wang Y. та ін. A survey of large language models for code: evolution, benchmarking, and future trends. 2023.

5. Dhanuka D. Impact of LLMs on team collaboration in software development. 2025. URL: https://arxiv.org/abs/2510.08612

6. Wang K., Li T., Zhang X. та ін. Software development life cycle perspective: a survey of benchmarks for CodeLLMs and agents. 2025. URL: https://arxiv.org/abs/2505.05283

ISSN 2786-6025 Online
7. Joel S., Wu J. W., Fard F. H. A survey on LLM-based code generation for low-resource and domain-specific programming languages. 2024. URL: https://arxiv.org/abs/2410.03981
8. Mateus de Brito V., Farias K. Understanding the role of large language models in software engineering: evidence from an industry survey. 2025. URL: https://arxiv.org/abs/2512.21347

Дата першого надходження статті до видання: 07.04.2026
Дата прийняття статті до друку після рецензування: 22.04.2026

Журнал

«Наука і техніка сьогодні»

Випуск № 4(58) 2026

Формат 60х90/8. Папір офсетний.
Гарнітура Times New Roman.
Ум. друк. арк. 8,2. Наклад 100 прим.

Видавець:
Громадська наукова організація «Всеукраїнська асамблея докторів наук з державного управління»
Свідоцтво серія ДК №4957 від 18.08.2015 р., Андріївський узвіз, буд.11, оф 68, м. Київ, 04070.