

Мохналь А.В.

студентка 4-го курсу, група ІН-402,

Київський національний економічний університет імені Вадима Гетьмана

tatokhnalalina@gmail.com

Денісова О.О.,

к.е.н., доц., доцент кафедри ІСЕ,

Київський національний економічний університет імені Вадима Гетьмана

denisova@kneu.edu.ua

ДОСЛІЖЕННЯ ПРАКТИКИ ВАЙБ-КОДУВАННЯ

Вайб-кодування (VibeCoding) – це новий підхід до розробки програмного забезпечення, що був запропонований Андреем Карпати, співзасновник OpenAI і колишнім керівником ШІ в Tesla, на початку 2025 року. Його головна ідея полягає в тому, що програмісти все більше покладаються на ШІ для виконання рутинної та іноді важкої роботи з написання коду. Це дозволяє програмістам «повністю довіритися вайбу» описуючи свої думки і наміри та формулюючи завдання природною мовою, на основі чого штучний інтелект може згенерувати більшу частину фактичного коду [1]. Процес зазвичай включає цикли опису завдання, прийняття автоматично згенерованого коду з мінімальною перевіркою і запитів до ШІ про доопрацювання виявлених проблем. Важливою особливістю є те, що здебільшого розробники мають приймати код без повного його розуміння на відміну від звичайного підходу до використання ШІ для програмування, коли за нейромережа генерує програмний код, а розробник ретельно його перевіряє, тестує і повністю його розуміє [2].

Такий принципово новий підхід викликає суперечливі оцінки в ІТ-спільноті. З одного боку, зростає занепокоєння серед професійних розробників, що спрощення процесу програмування та автоматизації створення коду з використанням ШІ можуть призупинити розвиток технічних навичок фахівців, що в перспективі загрожуватиме якості програмних рішень. З іншого боку, вайб-кодування зміщує акцент з технічної реалізації на генерування цінних ідей та розробку концепції продукту. Практика демонструє, що цей підхід знаходить практичне застосування в різних контекстах. Наприклад, великі підприємства, такі як Accenture та IBM, мають команди, де певна частина консультантів є кваліфікованими програмістами, а решта "вайбують" через проєкти, працюючи з попередньо визначеними Java APIs та структурами.

Вайб-кодування відкриває ширші можливості для людей із мінімальним досвідом програмування [3]. Попри технічні недоліки та обмежений функціонал, сам факт створення життєздатних проєктів в такі стислі терміни є вражаючим. Прикладом є створення Пітером Левелсом багатокористувацької гри-симулятора польоту за один день – він лише описав бажаний функціонал, провів кілька ітерацій і опублікував гру. Продукт було успішно протестовано – в цю гру одночасно грали тисячі людей упродовж 24 годин [4]. Іншим прикладом є створення маркетологом з нульовими навичками програмування за 30 хвилин за допомогою AI Agent від Replit маркетингового інструменту під назвою «Content Genie. Цей додаток допомагає проводити мозкові штурми та генерувати ідеї контенту. Подібні проєкти раніше створювали лише програмісти [5]. Ще одним реальним кейсом є створення колекції зі 100 простих веб-інструментів на одному веб-сайті, створену автором без жодних знань з програмування – він лише описав кожен міні-додаток штучному інтелекту, і той згенерував для нього весь код. Щоразу, коли ШІ стикався з труднощами, автор підказував, що потрібно виправити або вдосконалити. Результатом став JustBuildThings.com – універсальний сайт для корисних веб-утиліт, який з нульовим маркетинговим бюджетом посів третє місце у рейтингу "Product of the Day" на Product Hunt [6].

Окремим напрямком є вайб-кодування для розробки мінімально життєздатних продуктів для тестування ідеї, виведення продукту на ринок і вивчення попиту. Прикладом є три цифрові платформи, створені за допомогою bolt та v0. Перший з них – платформа управління контрактами та

постачальниками, яка допомагає організаціям ефективно керувати бізнес-відносинами. Система включає просунуте управління життєвим циклом контрактів з автоматизованим відстеженням поновлень, складну оцінку ризиків постачальників за матрицею Кралліча, та детальний моніторинг відповідності, включно з ISO та NEN оцінками. Вона також має функції управління документами, відстеження зустрічей, моніторингу KPI та ін. Другою є платформа управління, ризиків та відповідності, яка включає сканування нормативної відповідності, нормативні фреймворки, управління політиками тощо. Третій продукт - платформа особистого розвитку, яка допомагає перетворювати наміри на досягнення у всіх сферах життя. Вона поєднує постановку цілей, управління завданнями та інструменти глибокої рефлексії для створення всебічної системи особистісного зростання. Як каже автор, перші два проекти мають гарний «локальний» відгук, що є вдалим стартом для подальшого розвитку [7].

Не зважаючи на численні успішні кейси, вайб-кодування має суттєві обмеження. Результати аналізу надають підстави стверджувати, що з одного боку, фахівці без технічної освіти отримують можливість самостійно створювати функціональні додатки, проводити швидку валідацію бізнес-гіпотез та мінімізувати витрати на початкових етапах розробки. Підприємці отримують небачену раніше швидкість трансформації ідей у робочі прототипи, що відбувається протягом декількох днів замість тижнів або місяців, як при традиційному програмуванні. Водночас, ШІ-генерований код часто не відповідає стандартам оптимізації, що негативно впливає на продуктивність системи при високих навантаженнях. Аналіз безпеки таких додатків показує підвищений ризик вразливостей через недостатню увагу ШІ до специфічних питань захисту даних, а масштабування рішень ускладнене архітектурними недоліками, що проявляються при розширенні функціональності. Для складних рішень на рівні підприємства чи продуктів з критичними навантаженнями такий підхід матиме серйозні ризики. Додатковою проблемою є складність підтримки та модифікації коду, бо не кожен зможе одразу зрозуміти логіку штучного інтелекту, а відсутність чітких пояснень логіки та документації значно ускладнює процеси налагодження та оновлення для технічних спеціалістів. На основі цього можна зробити висновок про оптимальність використання гібридного підходу – генерування коду за допомогою ШІ та подальше його вдосконалення професійними розробниками.

Результати дослідження надають підстави стверджувати, що вайб-кодинг є ефективним підходом для прототипування, валідації ідей та швидкої розробки мінімально життєздатних продуктів, але не є придатним інструментом для рішень, що потребують масштабування.

Список використаних джерел

1. Пост Andrej Karpathy в X (Twitter). URL: https://x.com/karpathy/status/1886192184808149383?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1886192184808149383%7Ctwgr%5E81bb25c6a8999f5b3c1308987ba4f4a49a51a8cd%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fdev.ua%2Fnews%2Fvaib-kodynh-intensifies
2. Not all AI-assisted programming is vibe coding (but vibe coding rocks). URL: <https://simonwillison.net/2025/Mar/19/vibe-coding/>
3. What is vibe coding? How creators are building software with no coding knowledge. URL: <https://news.ycombinator.com/item?id=43218410#:~:text=Accenture%2C%20IBM%20and%20other%20big,just%20vomits%20something%20inside%20it>
4. Пост Pieter Levels в X (Twitter). URL: <https://x.com/levelsio/status/1894429987006288259>
5. No-Code App in 30 Minutes? My Replit AI Experiment (Step-by-Step). URL: <https://www.dotmartin.io/p/build-app-without-coding-replit-ai-guide?ref=creator.alitu.com>
6. Zero coding skills, my web app ranked #3 on producthunt. URL: https://www.reddit.com/r/cursor/comments/1gexwmq/zero_coding_skills_my_web_app_ranked_3_on/?ref=creator.alitu.com#:~:text=zero%20coding%20skills%2C%20my%20web,3%20on%20producthunt

7. What vibe coding projects have you made? URL: https://www.reddit.com/r/vibecoding/comments/1j1ykeg/what_vibe_coding_projects_have_you_made/?rdt=59749

8. Why I stopped using AI code editors. URL: <https://lucianonooijen.com/blog/why-i-stopped-using-ai-code-editors/>

9. What's the vibe around vibe coding? URL: <https://www.ibm.com/think/news/vibe-coding-devs>

*Неділько І.М., аспірант
Запорізький національний університет
i.nedilkko@gmail.com*

ЗАГАЛЬНИЙ ОГЛЯД ПЛАТФОРМИ ДЛЯ АГЕНТНОГО МОДЕЛЮВАННЯ ЯК АЛЬТЕРНАТИВА NETLOGO

Агентно-орієнтоване моделювання (Agent-Based Modeling, ABM) є одним із ключових підходів до аналізу складних динамічних систем. Воно використовується в різних сферах, таких як економіка, соціальні науки, екологія, моделювання трафіку та епідеміологія[2]. Головна перевага ABM полягає в можливості врахування індивідуальної поведінки агентів і їх взаємодії, що дозволяє отримувати глибші аналітичні висновки порівняно з традиційними диференційними або статистичними моделями[3].

Однією з найпопулярніших платформ для ABM є NetLogo, що забезпечує простоту використання завдяки вбудованому скриптовому середовищу та візуальному інтерфейсу. Вона широко використовується для освітніх і наукових цілей. Однак ця платформа має значні обмеження:

- Низька продуктивність при великій кількості агентів – NetLogo використовує однопотокову обробку, що обмежує швидкість розрахунків[1].
- Обмежена масштабованість – при моделюванні великомасштабних систем продуктивність суттєво падає.
- Закритість екосистеми – хоча платформа має багатий набір функцій, вона менш гнучка для інтеграції з іншими інструментами, наприклад, з мовами програмування високої продуктивності.

Однією з основних проблем NetLogo є те, що всі обчислення виконуються в одному потоці[1]. Нова платформа підтримує багатопотоковість і паралельні обчислення, що дозволяє суттєво збільшити продуктивність при моделюванні великої кількості агентів.

Отже, розглянута платформа для агентного моделювання демонструє значні переваги перед NetLogo, особливо в аспектах продуктивності, масштабованості та гнучкості інтеграції. Подальший розвиток передбачає:

- Оптимізацію обчислень за допомогою GPU для ще більшої продуктивності.
- Додавання підтримки хмарних технологій для можливості виконання складних моделей на розподілених системах.
- Розширення бібліотеки шаблонних агентних моделей, що дозволить дослідникам швидше адаптувати платформу до власних задач.

Таким чином, запропонована альтернатива NetLogo має потенціал стати потужним інструментом для наукових досліджень та прикладного моделювання в різних сферах.

Список використаних джерел

1. Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University.