УДК 004.41:004.8:378.091.3

Олексієнко А.О., здобувач,
Вовк Р.Б., к.т.н., доцент
Івано-Франківський національний технічний університет нафти і газу

МЕТОДОЛОГІЯ ПІДГОТОВКИ ІНЖЕНЕРІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УМОВАХ ІМПЛЕМЕНТАЦІЇ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ

Сучасна парадигма професійної підготовки інженерів програмного забезпечення перебуває у стані фундаментальної трансформації, детермінованої масовою дифузією великих мовних моделей (Large Language Models, LLM). Інтеграція інтелектуальних агентів, таких як GitHub Copilot та Claude, у дидактичний процес створює прецедент нівелювання чіткої межі між автономним інтелектуальним пошуком суб'єкта навчання та автоматизованою генерацією програмних рішень. Ключовою проблемою даного переходу постає феномен «когнітивного розвантаження», що полягає у делегуванні складних логічних операцій алгоритмічним системам. Це призводить до ерозії етапу побудови внутрішніх ментальних моделей коду та формування ілюзії компетентності: при високій операційній швидкості здобувач втрачає здатність до самостійної дедукції та верифікації рішень у нетривіальних сценаріях [1].

Аналіз когнітивної взаємодії людини з системами штучного інтелекту свідчить про якісну зміну характеру інтелектуальної праці інженера. Традиційний акцент на синтаксичній коректності та алгоритмічній грамотності зміщується у площину «архітектурного нагляду» [2]. Рольова модель студента еволюціонує від лінійної імплементації коду до експертної валідації та рецензування. Зазначена трансформація вимагає не лише опанування методів промпт-інжинірингу, а й розвитку здатності до критичного аудиту згенерованого контенту на предмет безпеки та логічних аберацій («галюцинацій»). Мінімізація «продуктивних труднощів» (productive failure) у процесі написання коду з нуля загрожує зниженням когнітивної пластичності та аналітичної глибини фахівця.

Реалізація оновленої освітньої моделі в умовах домінування генеративного ШІ потребує системної реконфігурації структури підготовки. На зміну ізольованим технічним навичкам приходить комплексна цифрова грамотність, що базується на синергії з інтелектуальними системами. У межах науково-технічного дискурсу виокремлюються такі стратегічні компетенції:

- Метакогнітивний архітектурний аналіз - здатність до концептуальної декомпозиції складних систем за умови делегування рутинних операцій ШІ при збереженні контролю над цілісністю системних зв'язків.

- Технічний аудит та етико-правова валідація - верифікація згенерованих артефактів на відповідність стандартам безпеки (зокрема CWE) та аналіз ліцензійної чистоти отриманого коду [3].

- Адаптивне забезпечення якості - перехід від детермінованого написання тест-кейсів до проектування стратегій автоматизованого тестування для виявлення граничних умов та вразливостей.

- Гібридна технічна комунікація - формалізація вимог у специфікації, що є релевантними як для людського сприйняття, так і для обробки інтелектуальними агентами.

Запропонований структурний підхід дозволяє конвертувати інструменти ШІ з фактора когнітивної деградації у засіб інтелектуального масштабування, що корелює із запитами індустрії на підготовку адаптивних фахівців [4].

Перспективи подальших досліджень полягають у розробці адаптивних навчальних планів, що інтегрують етичні та безпекові виклики використання LLM. Виховання професійної відповідальності за автоматизований результат має бути імплементовано як базисну компетенцію нарівні з фундаментальними знаннями структур даних та алгоритмів. Пріоритет теоретичного базису над швидкістю імплементації є необхідною умовою підготовки інженерів, здатних до генерації інновацій у пост-алгоритмічну епоху.

Список використаних джерел:

1. Lau S., Guo P. J. The Impact of AI on Computer Science Education: A Study of Cognitive Offloading [Електронний ресурс]. Stanford University. 2024. URL: https://hci.stanford.edu/publications/2024/ai-educognitive-offloading.pdf (дата звернення: 22.03.2026).

2. White J., Fu Q., Hays S. Prompt Engineering for Software Engineering Education [Електронний ресурс]. IEEE Software. 2023. URL: https://arxiv.org/abs/2302.11382 (дата звернення: 22.03.2026).

3. Pearce H., Ahmad B., Tan B. Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions [Електронний ресурс]. 2023. URL: https://arxiv.org/abs/2108.09293 (дата звернення: 22.03.2026).

4. Designing Transparent Rubrics for AI-Based Evaluation: A Practical Guide [Електронний ресурс]. The Case HQ. 2026. URL: https://thecasehq.com/designing-transparent-rubrics-for-ai-basedevaluation/ (дата звернення: 22.03.2026).