



Fig.1. AI-model in CAD

Despite the clear advantages, challenges remain. High computational demands, the need for large and diverse training datasets, and integration with legacy CAD platforms are ongoing concerns. Moreover, successful deployment requires interdisciplinary expertise, combining engineering design, data science, and software development.

In conclusion, the integration of AI into CAD systems is a global trend reshaping computer engineering. Research and development efforts in the United States, Germany, France, and across Europe are driving innovation in intelligent design automation. Continued investment in AI infrastructure, open-source datasets, and cross-disciplinary education will be essential to fully realize the potential of AI-enhanced CAD environments.

### References

1. Müller, T., & Schneider, F. AI-driven microchip topology optimization: A Fraunhofer perspective. *Journal of Computational Engineering*, 12(4), 2022. Pp. 215–223. URL: <https://doi.org/10.1016/j.jce.2022.04.005> (accessed 25 October 2025).
2. Dupont, C., & Moreau, J. Intelligent assistants in CAD environments: Learning user behavior for adaptive design. *INRIA Research Reports*, RR-9456. 2023. URL: <https://hal.inria.fr/hal-04567890> (accessed 27 October 2025).

## ПРОМТ-ІНЖИНІРИНГ ЯК КОМПОНЕНТ МЕТОДИЧНОЇ СИСТЕМИ РОЗВИТКУ ПРОФЕСІЙНИХ КОМПЕТЕНТНОСТЕЙ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ

**Балик Анатолій Володимирович**

здобувач третього рівня вищої освіти, спеціальність Освітні, педагогічні науки  
Тернопільський національний педагогічний університет імені Володимира Гнатюка  
vodinn@tntpu.edu.ua

В умовах стрімкої інтеграції генеративного штучного інтелекту (ШІ) в усі галузі суспільного життя, що набуває особливої актуальності останнім часом, відбувається докорінна трансформація вимог до професійних компетентностей (ПК) педагогічних кадрів. Для майбутніх учителів інформатики ця трансформація є подвійною: вони мають не лише опанувати ШІ як об'єкт вивчення, але й інтегрувати його як дидактичний інструмент у власну професійну діяльність.

Ця трансформація висуває на перший план нову, фундаментальну навичку, що стає де-факто новою формою грамотності. Сучасні дослідники, зокрема Д. Федеріакін та ін. [3], концептуалізують промт-інжиніринг не просто як технічну дію, а як нову ключову компетентність 21-го століття. Ця компетентність є

багатокомпонентною і включає не лише промпт-грамотність (здатність формулювати запити), але й методи промптингу (знання технік як-от Few-shot, Chain-of-Thought тощо) та, що критично важливо, навички критичного онлайн-міркування для валідації та інтерпретації результатів, згенерованих ШІ.

Водночас, систематичні огляди, зокрема Х. Тан та ін. [5], вказують на наявний розрив між потенціалом ШІ та реальними програмами професійного розвитку вчителів. Дослідники фіксують значні бар'єри у впровадженні ШІ-технологій та наголошують на нагальній потребі розробки нових, ефективних методичних систем для підготовки та підвищення кваліфікації педагогів. Традиційні методики підготовки вже не повною мірою відповідають цим викликам.

Наукова новизна даного дослідження полягає у теоретичному обґрунтуванні та розробці методичної системи, де промпт-інжиніринг розглядається не як окрема, ізольована навичка, а як системоутворюючий компонент для цілісного розвитку всього спектру професійних компетентностей (методичної, кодувальної, проєктної, аналітичної) майбутнього вчителя інформатики.

В основі нашої методичної системи лежить гіпотеза, що цілеспрямоване використання різних за функціоналом AI-платформ через призму промпт-інжинірингу дозволяє акцентовано розвивати різні групи ПК. Ми розглядаємо AI-асистента не просто як допоміжний інструмент, а як «цифрового спаринг-партнера», взаємодія з яким вимагає від викладача, вчителя, студента постійного уточнення мислення, чіткого формулювання завдань та критичної оцінки результатів [1].

Процес навчання у запропонованій системі базується на трьох ключових типах AI-платформ, кожна з яких має унікальний дидактичний потенціал, детально проаналізований у нашому дослідженні.

1. Платформи типу IDE-асистентів (наприклад, GitHub Copilot). GitHub Copilot, інтегрований безпосередньо у середовища розробки (IDE), слугує інструментом для розвитку кодувальної та рефакторингової ПК. Дидактичний потенціал тут реалізується через промпти у вигляді коментарів до коду. Майбутній учитель вчиться не просто «просити» готовий код, а формулювати завдання (промпт) настільки чітко, щоб AI-асистент згенерував оптимальний, чистий та ефективний програмний блок. Промпт-інжиніринг у цьому випадку розвиває здатність до декомпозиції задачі, розуміння шаблонів проєктування та навичок «читання» і миттєвої оптимізації чужого коду (коду, згенерованого ШІ).

2. Платформи типу інтегрованих онлайн-IDE (наприклад, Replit AI). Replit AI поєднує середовище розробки з хмарною інфраструктурою та потужним ШІ-асистентом. Цей інструмент є ідеальним для розвитку проєктної та колаборативної ПК. Промпт-інжиніринг тут використовується для швидкого прототипування, спільного рефакторингу коду в реальному часі та отримання миттєвого зворотного зв'язку у хмарному середовищі. Студенти вчать використовувати ШІ для управління навчальними проєктами, автоматизації рутинних завдань та організації спільної роботи. Такий підхід узгоджується з сучасними українськими дослідженнями щодо практичного застосування генеративного ШІ для розробки прикладних елементів навчальних програм [4].

3. Платформи типу AI-аналітиків (наприклад, Claude). Моделі від Anthropic, зокрема Claude, вирізняються здатністю опрацювати надзвичайно великі обсяги контексту (до сотень сторінок коду чи документації) та надавати глибокі, пояснювальні відповіді. У нашій методичній системі Claude використовується для розвитку аналітичної, пояснювальної та методичної ПК. Майбутні вчителі використовують промпт-інжиніринг для: а) аналізу складних алгоритмів (промпт: «Поясни цей код покроково, ніби я 10-класник, акцентуючи на рекурсивному виклику»); б) генерації дидактичних матеріалів (промпт: «Створи три задачі різного

рівня складності на тему «Словники у Python» з поясненнями типових помилок»); в) аналізу та документування цілих проєктів. На відміну від Copilot, орієнтованого на генерацію коду, Claude орієнтований на генерацію розуміння та методології.

Важливим аспектом нашої методичної системи є не лише що вивчати (тобто, диференціація платформ), але й як це впроваджувати. Замість тривалих теоретичних курсів, ми пропонуємо апробацію у форматі короткострокових освітніх інтенсивів (наприклад, «воркшопів з промпт-інжинірингу») та мікро-турнірів з розв'язання методичних завдань. Наш підхід спирається на новітні емпіричні дані, зокрема пілотне дослідження Р. Давіла-Моран та ін. [2]. Це дослідження довело, що навіть короткі, трисесійні практикуми здатні суттєво підвищити ІІІ-грамотність та, що важливо, значно знизити рівень «технологічної тривожності» (technology anxiety) у майбутніх педагогів. Це доводить високу ефективність саме таких коротких навчальних практикумів для швидкої адаптації студентів до нових інструментів.

Проведене дослідження дає підстави стверджувати, що промпт-інжиніринг трансформувався з допоміжної технічної навички в одну з ключових професійних компетентностей вчителя інформатики 2025 року. Він є фундаментальним компонентом запропонованої методичної системи підготовки, оскільки слугує наскрізним інструментом для розвитку всього спектру фахових компетентностей.

Запропонований у роботі підхід, що базується на диференційованому використанні AI-платформ (GitHub Copilot – для кодування, Replit AI – для проєктів, Claude – для аналітики та методики), дозволяє систематично та цілеспрямовано розвивати професійні компетентності майбутнього вчителя. Обґрунтовано доцільність впровадження таких методик у форматі короткострокових освітніх інтенсивів, оскільки такий формат сприяє зниженню технологічної тривожності студентів та швидкому підвищенню їхньої ІІІ-грамотності. Це, у свою чергу, створює міцне підґрунтя для ефективного впровадження програм професійного розвитку вчителів, нагальна потреба в яких фіксується у сучасних наукових оглядах.

Перспективи подальших досліджень полягають у експериментальній перевірці розробленої методичної системи, розробці валідного інструментарію для оцінювання рівня сформованості «промпт-компетентності» та її інтеграції у зміст освітніх програм підготовки вчителів інформатики.

### Список використаних джерел

1. Олексюк В., Спирін О., Балик Н., Іванова С. Розвиток цифрової компетентності наукових та науково-педагогічних працівників засобами генеративного штучного інтелекту. *Освіта. Інноватика. Практика*, 2025. № 13(8). С. 110–121.
2. Davila-Moran R. C., Sanchez Soto J. M., Lopez Gomez H. E., Silva Infantes M., Arias Lizares A., Huanca Rojas L. M., Cama Flores S. J. Brief Prompt-Engineering Clinic Substantially Improves AI Literacy and Reduces Technology Anxiety in First-Year Teacher-Education Students: A Pre-Post Pilot Study. *Education Sciences*, 2025. Vol.15(8). 1010 p.
3. Federiakin D., Molerov D., Zlatkin-Troitschanskaia O., Maur A. Prompt engineering as a new 21st century skill. *Frontiers in Education*, 2024. Vol. 9. 1366434 p.
4. Kakun A., Tytenko S. Generative AI and Prompt Engineering in Education. *Modern Engineering and Innovative Technologies*, 2023. № 1(29-01). P. 117–121.
5. Tan X., Cheng G., Ling M. H. Artificial intelligence in teaching and teacher professional development : A systematic review. *Computers & Education: Artificial Intelligence*, 2025. Vol. 8. 00355 p.