Ткаченко Олексій Миколайович, к.т.н, доцент

ДОСВІД ВИКОРИСТАННЯ ШІ ДЛЯ ФОРМАЛЬНОЇ СПЕЦИФІКАЦІЇ В КУРСІ "ФОРМАЛЬНІ МЕТОДИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ"

У статті проаналізовано досвід використання ШІ для формальної специфікації програмних систем (Dafny, TLA+) магістрантами у 2025 та 2026 роках. Досліджено еволюцію використання ШІ-моделей та перехід від інтерактивного генерування коду до розробки автономних агентів. Виявлено зростання рівня успішності верифікації, а також зміну характеру помилок моделей від простих синтаксичних до глибоко семантичних. Обґрунтовано зміну парадигми розробки: ключовою навичкою фахівця стає інженерія формальних вимог, тоді як ШІ виконує автоматизоване генерування та верифікацію коду.

Ключові слова: формальна специфікація, ШІ

The work analyzes the use of AI for formal software specification (Dafny, TLA+) by M.S. students in 2025 and 2026. The evolution of AI models and the transition from interactive code generation to autonomous agents are investigated. An increase in verification success rate and a shift in model errors from simple syntactic to deeply semantic are revealed. A development paradigm shift is substantiated: the specialist's key skill becomes formal requirements engineering, while AI performs automated code generation and verification.

Keywords: formal specification, AI

Вступ

Впровадження формальних методів розробки програмного забезпечення історично розглядалося як ресурсоємний процес, виправданий лише у сферах з критичними вимогами до безпеки.

Згідно з даними NASA і програмами досліджень DARPA, верифікація ПЗ може складати до 88% загальних витрат на створення систем високої надійності [1]. Тривалий час фундаментальні підходи до проєктування, розроблення і тестування залишалися незмінними, що призводило до стрімкого зростання собівартості розробки у поєднанні з ускладненням самих систем. Ще відносно недавно частка програмних систем у світі, при розробці яких застосовувалися формальні методи та інструменти, становила менше

відсотка, і ці системи були переважно в аерокосмічній, оборонній та ядерній галузях.

Однак динаміка за останні 10 років суттєво змінилася. Сучасні апаратні та програмні архітектури досягли рівня складності, який не дозволяє забезпечити ефективне покриття тестами. У 2025 році глобальний ринок інструментів формальної верифікації оцінювався у 1,8 мільярда доларів США, а до 2034 року очікується його зростання до 6,2 мільярда доларів при середньорічному темпі зростання 14,7%. Сьогодні середньостатистична команда розробників, наприклад, напівпровідників витрачає від 28% до 34% загального бюджету верифікації саме на формальні методи, що свідчить про перехід цих технологій з академічної ніші до індустріального стандарту. Окрім галузі напівпровідників (31,5% ринку), драйверами є аерокосмічна (22,4%) та автомобільна галузь (19,8%) [2].

Очевидним паралельним трендом індустрії ПЗ є розвиток і використання великих мовних моделей (LLM). У 2023 році близько 70% розробників повідомляли про використання інструментів ШІ, у 2024 році цей показник зріс до 76%, а згідно з глобальним опитуванням Stack Overflow, 84% розробників почали використовувати ШІ у своєму робочому процесі, при цьому 51% професійних інженерів застосовують інструменти штучного інтелекту щодня [3]. У той же час, 46% розробників зазначають, що не довіряють точності згенерованого коду, що створює потребу у формальних методах його перевірки [4]. Це посилює потребу у вивченні відповідних методів та інструментів на ІТ-спеціальностях.

Основна частина

У 2025 та 2026 роках в межах магістерського курсу "Формальні методи розробки програмних систем", який викладається для студентів освітньо-наукової програми "Інформатика" спеціальності 122/F3 "Комп'ютерні науки" на факультеті комп'ютерних наук та кібернетики в Київському національному університеті імені Тараса Шевченка, у практичне завдання формальної специфікації було включено дослідницький компонент, відповідно до якого студенти повинні були виконати завдання самостійно та з використанням обраної системи ШІ, зробити порівняння обох підходів та виконати рефлексію з обгрунтуванням проблем, що виникли та описом досвіду їх усунення. Формальні мови специфікацій та ШІ для виконання завдання студенти могли обирати на свій розсуд.

Аналіз студентських робіт продемонстрував, що найбільш популярною мовою формальних специфікацій була Dafny [5], дещо меншою

мірою – TLA+ [6]. Еволюція використання ними моделей ШІ подано в табл. 1, а розподіл популярності відображено на рис. 1.

Таблиця 1. Використання ШІ-моделей для формальної специфікації

Сімейство моделей	**Версії 2025 року**	**Частка 2025**	**Версії 2026 року**	**Частка 2026**
Anthropic Claude	3.5 Sonnet / Opus	26%	3.7 Sonnet / Opus	38%
OpenAI GPT/o-series	GPT-4o / GPT-5.2	42%	OpenAI o1 / GPT-5.5	32%
DeepSeek	DeepSeek V3.2 / R1	21%	DeepSeek-V4 / R2-Pro	22%
Відкриті моделі	Llama 3.1 / OLMo	11%	Llama 3.3 / Mistral Large 3	8%

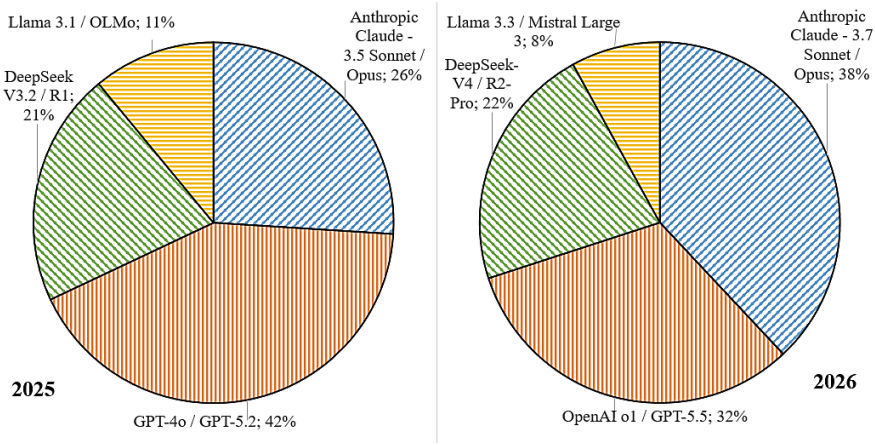


Рис.1. Частка використання студентами ШІ-моделей для формальної специфікації у 2025 та 2026 році

У 2026 році моделі сімейства Claude 3.7 витіснили продукти OpenAI з першого місця. У студентських звітах це пояснюється розширеним вікном контексту Claude та кращою здатністю дотримуватися строгих синтаксичних правил математичних мов. Також 2026 рік відзначився переходом на моделі з

внутрішнім циклом міркування (наприклад, OpenAI o1 та нові ітерації DeepSeek-R2).

Перехід на "reasoning" моделі у 2026 році обумовив покращення здатності ШІ писати специфікації.

Успішність (здатність згенерувати код, який проходить верифікацію) оцінювалася у двох режимах: без втручання людини ("one-shot") та з ітеративним зворотним зв'язком (REPL):

- середній показник "One-shot" успішності: зріс із 28-35% (у 2025) до 48-52% (у 2026). Хоча моделі стали якіснішими, вони все ще не здатні надійно видавати коректний результат з першої спроби;
- середній показник успішності з REPL (ітеративний): зріс із 65-72% (у 2025) до 88-92% (у 2026).

Процент успішності моделей для обраних мов:

- Dafny (82% успіху у 2026 проти 65% у 2025): стрибок успішності зумовлений тим, що нові моделі (Claude 3.7 та o1) були спеціально дотреновані розробниками на синтаксисі Dafny, що відзначили студенти у своїх рефлексіях;
- TLA+ (40% успіху у 2026 проти 29% у 2025): Темпоральна логіка все ще алгоритмічно "незручна" для текстових моделей.

Порівняння виявлених помилок за два роки ілюструє перехід проблем ШІ від синтаксичних до глибоко семантичних.

У 2025 році студенти відзначали:

- грубі синтаксичні помилки (змішування мов);
- плутанина з базовими кванторами (загальності та існування);
- нездатність написати навіть базовий інваріант циклу.

Що змінилося у 2026 році? Кількість синтаксичних помилок суттєво зменшилася. Однак у 2026 році студенти зіткнулися з новими типами помилок:

- "читерство" при доведенні (Proof Cheating): студенти виявили, що стикаючись зі складною теоремою, нові моделі ШІ іноді додають суперечливі аксіоми (наприклад, конструкції типу assume false; у Dafny), щоб обдурити солвер і змусити його прийняти доведення;
- деградація інваріантів у складних структурах даних: ШІ у 2026 році навчився писати інваріанти для масивів та простих циклів, але при роботі з графами чи складними деревами знову скочується до генерування сильних умов, які обумовлюють для солвера труднощі з індуктивним доведенням;

- переускладнення специфікацій (Over-specification): нові "reasoning" моделі схильні описувати кожен дрібний стан системи, навіть коли це не вимагається контрактом, що призводить до зависання SMT-солверів.

Висновки

На основі аналізу студентських звітів про виконання практичної частини формальної специфікації самостійно та за допомогою ШІ можна зробити кілька висновків.

1. Бачення студентів щодо готовності ШІ генерувати формальну специфікацію.

У 2025 році студенти робили висновок, що ШІ – просто "розумний автокомпліт", який вимагає інтерактиву з розробником та ручне втручання. У 2026 році понад 60% студентів не використовували веб-чати, а вже почали розробляли агентні пайплайни (Agentic Workflows) через API. Деякі здобувачі писали скрипти, де ШІ-агент самостійно генерував специфікацію, викликав компілятор (наприклад, Dafny), читав помилки, аналізував їх і переписував код, доки не отримував успішну верифікацію.

Загальний висновок 2026 року: ШІ частково готовий до написання формальних специфікацій. Роль людини зводиться до формулювання математично точної початкової вимоги, а рутинну роботу виконує агент.

2. Було проаналізовано рефлексії студентів щодо майбутнього індустрії розробки ПЗ та ролі розробника і ШІ.

У 2025 році студенти зазначали зниження вартості написання коду і вважали формальну специфікацію просто "корисною навичкою" для тестування того, що нагенерував ШІ. Інженерія вимог розглядалася як убезпечення від ШІ-галюцинацій. У звітах 2026 року з'явилося усвідомлення зміни парадигми. Оскільки ШІ-агенти тепер здатні самостійно писати складний прикладний код і виправляти його, популярний раніше "Prompt Engineering" визнано менш перспективним. Деякі студенти 2026 року зазначають, що в найближчі 5 років професія "програміст" значною мірою трансформується у професію "Інженер-специфікатор" (Specification Engineer). Головним завданням людини буде написання контрактів в математично-орієнтованій нотації. Коли контракт написаний людиною і є формально доведеним, наступне генерування коду, йогох оптимізація і верифікація стають автоматизованим процесом, який виконують ШІ-агенти.

3. Загальні висновки у порівнянні 2025 та 2026 року.

Порівняльний аналіз студентських звітів курсу "ормальні методи розробки програмних систем" за 2025 та 2026 роки показує не просто

еволюцію інструментів, а якісний перехід у методології розробки. За один рік студенти перейшли від виправлення згенерованих ШІ синтаксичних помилок до побудови автономних довідних агентів. Мови типу Dafny вийшли на перший план, вони краще пристосовані для машинної обробки. Головний висновок студентів 2026 року полягає в тому, що формальна специфікація більше не є "нішовим академічним предметом", а еволюціонує у напрямі промислового стандарту у поєднанні з використанням ШІ.

Список використаних джерел

1. Reducing Verification Costs through Practical Formal Methods: A Survey - KnowledgeBytes, *http://www.knowledgebytes.net/downloads/Practical_Formal_Methods_A_Survey_2011_v1.pdf* (дата звернення: 24.08.2026).

2. Formal Verification Copilot Market Research Report 2034 - Dataintelo, *https://dataintelo.com/report/formal-verification-copilot-market* (дата звернення: 24.08.2026).

3. AI | 2025 Stack Overflow Developer Survey, *https://survey.stackoverflow.co/2025/ai* (дата звернення: 24.08.2026).

4. 84% of developers use AI, yet most don't trust it! - ShiftMag, *https://shiftmag.dev/stack-overflow-survey-2025-ai-5653/* (дата звернення: 24.08.2026).

5. The Dafny Programming and Verification Language. *https://dafny.org*/ (дата звернення: 24.08.2026).

6. Learn TLA+. *https://learntla.com*/ (дата звернення: 24.08.2026).