

**ШТУЧНИЙ ІНТЕЛЕКТ У СУЧАСНІЙ ВЕБРОЗРОБЦІ ТА ВЕБДИЗАЙНІ:  
БАГАТОРІВНЕВА КЛАСИФІКАЦІЯ ТА СИСТЕМАТИЗАЦІЯ**

Ю.Г. Лобода, О.Г. Трофименко, С.Ю. Манаков, В.І. Гура

---

Національний університет «Одеська юридична академія»  
23, Фонтанська дорога, м. Одеса, 65009, Україна  
Emails: loboda@onua.ua, trofymenko@onua.edu.ua

---

Дослідження присвячене комплексному аналізу впливу технологій штучного інтелекту (ШІ) на процеси сучасної веброзробки та вебдизайну. Актуальність теми зумовлена стрімким розвитком технологій ШІ та необхідністю їх системної інтеграції у професійні процеси розробки програмного забезпечення. Мета дослідження полягає у формуванні багаторівневої класифікації інструментів ШІ за ступенем автономності та когнітивної взаємодії, систематизації екосистеми інструментів ШІ відповідно до етапів життєвого циклу розробки програмного забезпечення. Методологія дослідження базується на системному підході, який включає аналіз сучасної наукової літератури, функціональної класифікації інструментів ШІ та порівняльному аналізу реальних кейсів використання ШІ у веброзробці. Наукова новизна полягає у створенні комплексної систематизації інструментів ШІ для веброзробки, що враховує когнітивні аспекти взаємодії. Основні результати поєднують створення трирівневої класифікації інструментів ШІ (асистенти, генератори, автономні системи) та розробку концептуальної моделі екосистеми рішень ШІ за етапами життєвого циклу веброзробки. Розроблена концептуальна модель візуалізує характер інтеграції ШІ, підтверджуючи, що його вплив не обмежується етапом написання коду. Інструменти ШІ оптимізують кожен етап життєвого циклу веброзробки: від аналізу вимог (Notion AI) та прототипування (Uizard.io) до автоматизованого розгортання (GitLab AI) та проактивного моніторингу (Datadog AI). Особливе значення мають наскрізні процеси (SEO, а11y, A/B-тестування), які демонструють найвищий рівень зрілості інтеграції, оскільки вони функціонують неперервно та комплексно покращують продукт. Поєднання класифікації та моделі екосистеми дозволяє зробити висновок, що ефективне впровадження ШІ вимагає не точкових рішень, а побудови цілісної стратегії, яка враховує рівень автономності інструментів на кожному етапі розробки. Практична значущість результатів полягає у можливості використання запропонованої класифікації для обґрунтованого вибору інструментів, оптимізації процесів розробки, підвищення продуктивності команд та формування стратегічних рішень щодо впровадження технологій ШІ в IT-проекти. Результати можуть бути використані як у професійній діяльності IT-компаній, так і в наукових дослідженнях у сфері інженерії програмного забезпечення.

**Ключові слова:** штучний інтелект, веброзробка, вебдизайн, життєвий цикл розробки, когнітивна взаємодія, автоматизація.

**Вступ.** Сучасний етап розвитку вебтехнологій характеризується стрімкою інтеграцією технологій штучного інтелекту (ШІ) в усі процеси створення та функціонування вебресурсів. Зростання складності програмних продуктів та підвищені вимоги до швидкості їх розробки стимулюють пошук нових методологій та інструментів [1]. Веброзробка та вебдизайн, як одні з найбільш динамічних сегментів IT-галузі, зазнають суттєвих змін і переходять від традиційних практик до гнучких, автоматизованих процесів на основі інтелектуальних систем.

У Концепції розвитку штучного інтелекту в Україні наголошено, що «використання технологій штучного інтелекту сприятиме зменшенню обсягу витрат, підвищенню ефективності виробництва, якості товарів і послуг» [2]. Штучний інтелект розширює можливості розробників та дизайнерів, проте відсутність цілісного розуміння його ролі на кожному етапі життєвого циклу розробки вебзастосунків створює невизначеність для бізнесу при ухваленні стратегічних рішень.

Попри значну кількість доступних на ринку інструментів ШІ, їхнє впровадження у робочі процеси часто має фрагментарний характер. Ключовою проблемою залишається відсутність уніфікованих підходів до кількісного оцінювання впливу використання рішень на основі ШІ на економічну ефективність та на продуктивність процесів веброзробки. Такий дефіцит ускладнює формування стратегічного бачення щодо повноцінної інтеграції інтелектуальних технологій у життєвий цикл створення вебзастосунків, навіть за умов наявності відповідних технічних інструментів.

**Аналіз досліджень та публікацій** показує активне зростання інтересу до застосування ШІ у веброзробці та вебдизайні. Основні напрями досліджень охоплюють різноманітні аспекти впровадження ШІ, зокрема автоматизацію процесів веброзробки, персоналізацію користувацького досвіду, оптимізацію продуктивності та зниження порогу входу для новачків, а також оцінювання ефективності використання інтелектуальних інструментів у цифрових продуктах.

Згідно з дослідженнями ШІ активно впроваджується у процеси розробки, тестування та обслуговування вебзастосунків, сприяючи автоматизації рутинних завдань і підвищенню продуктивності [1, 3, 4]. Генерація коду, автоматичне виявлення помилок, адаптивний дизайн і машинне тестування забезпечують зниження вартості розробки та покращення якості кінцевого програмного продукту. Дослідження [5, 6] відзначають, що використання ШІ у веброзробці сприяє зменшенню часу завантаження сторінок, ефективному розподілу навантаження та адаптації інтерфейсів до потреб користувачів.

Важливим напрямом є персоналізація контенту та взаємодії з користувачем. У роботах [1, 7, 8] підкреслено роль ШІ персоналізації досвіду користувача, зокрема через системи рекомендацій, аналіз поведінки користувачів та створення адаптивних інтерфейсів. Дослідники роблять висновки, що це не лише підвищує залученість, а й позитивно впливає на UX-показники. У роботах [3, 9, 10] зазначено, що завдяки штучному інтелекту та low-code платформам навіть користувачі без глибокої технічної підготовки можуть створювати складні вебзастосунки, використовуючи drag-and-drop інтерфейси та типові шаблони. Такий підхід сприяє демократизації розробки, розширює можливості малого бізнесу й стартапів, а також стимулює інноваційну активність.

Окрема група досліджень [9, 11-14] присвячена застосуванню генеративних моделей, алгоритмів машинного зору та інтелектуальних систем для побудови візуальних елементів та стилістичної адаптації вебінтерфейсів. Автори робіт [11, 16] акцентують на етичних викликах, пов'язаних із конфіденційністю, прозорістю алгоритмів і впливом автоматизації на креативну складову дизайну.

Особливу увагу приділено оцінюванню впливу ШІ на ефективність проєктної діяльності у веброзробці та вебдизайні. У роботах [1, 4, 5, 15-18] наведено результати емпіричних досліджень і метрик, які демонструють покращення продуктивності, зменшення кількості помилок, прискорення термінів виконання проєктів та позитивний вплив на користувацький досвід. Використання метрик Usability Goals Achievement Metric та Index of Integration дозволило кількісно оцінити рівень інтеграції ШІ в проєкт і його кореляцію з якістю кінцевого продукту [17]. Такі результати підтверджують практичну цінність впровадження ШІ у веброзробку.

Проведений аналіз засвідчує наявність ґрунтовної дослідницької бази щодо застосування технологій ШІ у веброзробці та вебдизайні. Водночас, попри стрімкий розвиток інструментів ШІ, спостерігається відсутність уніфікованих методичних підходів до їхньої системної інтеграції на всіх етапах життєвого циклу вебпроєктів. Недостатньо опрацьованими залишаються питання класифікації інструментів за ступенем автономності та когнітивної взаємодії, а також методи кількісної оцінки ефективності їх впровадження.

**Метою даної роботи** є формування багаторівневої класифікації інструментів штучного інтелекту за ступенем автономності та когнітивної взаємодії у сфері веброзробки та

вебдизайну, а також систематизація відповідної екосистеми за етапами життєвого циклу розробки.

Для досягнення поставленої мети визначено такі завдання:

- проаналізувати сучасні наукові публікації та дослідження для виявлення ключових напрямів, тенденцій та проблем у застосуванні ІІІ у веброзробці та вебдизайні;
- розробити класифікацію інструментів ІІІ, яка базується на рівні їхньої когнітивної взаємодії з користувачем та ступені автономності у виконанні завдань;
- систематизувати екосистему інструментів ІІІ шляхом їх співвіднесення з основними етапами життєвого циклу розробки програмного забезпечення;
- створити комплексну концептуальну модель, яка візуально ілюструє інтеграцію інструментів ІІІ у послідовні та наскрізні процеси веброзробки.

## **1. Класифікація інструментів ІІІ за рівнем когнітивної взаємодії.**

Сучасна екосистема веброзробки переживає період кардинальних трансформацій, зумовлених стрімким впровадженням технологій ІІІ. Для системного розуміння цих змін та їхнього впливу на професійну діяльність веброзробників доцільно структурувати різноманітні інструменти ІІІ за чіткими критеріями, що дозволить оцінити як поточний стан галузі, так і перспективи її розвитку.

Запропонована класифікація структурує інструменти ІІІ на основі глибини їхньої інтеграції у творчі та технічні процеси. Вона виділяє три послідовні рівні, які відображають еволюцію ролі ІІІ: від допоміжної функції, яка автоматизує рутинні операції, до повноцінного автономного агента, здатного напрацьовувати стратегічні рішення. Такий поділ дозволяє системно оцінити ступінь заміщення людської праці та потенціал для трансформації робочих процесів.

### **Перший рівень. Інструменти-асистенти (Assisted AI)**

Інструменти-асистенти є найпоширенішою та найбільш інтегрованою у сучасні робочі процеси категорією рішень ІІІ у веброзробці. Такі системи функціонують як високотехнологічні «помічники», які автоматизують рутинні мікрозавдання та надають контекстні підказки, при цьому залишають всі стратегічні рішення за людиною. Характерною особливістю цього рівня є потреба постійного контролю та керування з боку фахівця, оскільки інструменти доповнюють людські можливості і не замінюють при цьому креативне мислення та професійну інтуїцію.

У сфері розробки прикладом таких рішень є GitHub Copilot (<https://github.com/copilot>), який аналізує контекст написаного коду та пропонує наступний логічний рядок або блок коду, базуючись на патернах, вивчених з мільйонів репозиторіїв. Подібним чином працює Tabnine (<https://www.tabnine.com>), який забезпечує розумне автодоповнення коду та враховує стиль програмування конкретного проєкту. Amazon Q Developer (<https://aws.amazon.com/q/developer>) є інтелектуальним помічником, який не лише автоматизує створення та пояснення програмного коду, а й виконує проактивний аналіз безпеки, виявляє потенційні вразливості ще на етапі написання.

У дизайнерській сфері інструменти-асистенти представлені різноманітними плагінами для Figma, які автоматично перевіряють контрастність кольорів відповідно до стандартів доступності, інструментами автоматичного вирівнювання об'єктів, що економлять час на технічних операціях, та генераторами кольорових палітр, які створюють гармонійні комбінації на основі завантаженого зображення або заданих параметрів.

### **Другий рівень. Інструменти-генератори (Generative AI)**

Цей рівень когнітивної взаємодії представляють інструменти-генератори, які здатні створювати цілісні артефакти на основі високорівневих текстових запитів користувача. Таки системи демонструють якісно вищий рівень автономності, оскільки можуть генерувати дизайн-макети, модулі коду, контент та інші складні об'єкти,

виходячи з абстрактного опису потреб. Однак результат їхньої роботи зазвичай потребує подальшої верифікації, доробки та адаптації людиною, що робить їх потужними інструментами для прискорення процесу створення початкових версій проєктів.

У розробці яскравим представником цієї категорії є v0.dev (<https://v0.dev>) від Vercel, здатний генерувати повноцінні React-компоненти за текстовим описом функціональності та зовнішнього вигляду. GitHub Copilot Chat розширює можливості базового Copilot, дозволяючи створювати функції, класи та навіть архітектурні рішення через діалоговий інтерфейс. Cursor AI (<https://cursor.com/>) пропонує генерацію комплексних архітектурних рішень та рефакторинг наявного коду з урахуванням кращих практик.

Дизайнерські інструменти-генератори представлені такими рішеннями, як Galileo AI (<https://galileo.ai>), який створює повноцінні UI-макети за текстовим описом бажаного інтерфейсу, враховуючи сучасні тренди та принципи UX. Uizard (<https://uizard.io>) спеціалізується на перетворенні рукописних ескізів та wireframes на цифрові прототипи з інтерактивними елементами. Midjourney (<https://www.midjourney.com>), хоча і не є спеціалізованим вебдизайн-інструментом, активно використовується для генерації унікальних графічних елементів, ілюстрацій та візуальних концепцій. Relume AI (<https://www.relume.io>) автоматизує початкові етапи проектування, створюючи карти сайтів та базові wireframes на основі опису бізнес-цілей та цільової аудиторії.

### Третій рівень 3. Автономні системи (Autonomous AI)

Найвищий рівень когнітивної взаємодії представляють автономні системи, здатні самостійно керувати цілісними процесами з мінімальним втручанням людини. Такі системи характеризуються здатністю напрацьовувати складні рішення на основі заданих KPI та бізнес-цілей, адаптуватися до змінних умов та навіть самонавчатися на основі набутого досвіду. У контексті веброзробки такі системи поки переважно є експериментальними, однак уже демонструють вражаючий потенціал для кардинальної зміни галузі.

Прикладом автономних систем є :

- комплексні рішення для А/В тестування (табл.1), які не лише генерують альтернативні варіанти дизайну на основі аналізу поведінки користувачів, а й самостійно проводять тестування, аналізують статистичну значущість результатів та автоматично впроваджують найкращий варіант без втручання людини;
- системи динамічної персоналізації UI в реальному часі адаптують інтерфейс під конкретного користувача на основі його поведінкових патернів, демографічних даних та контексту взаємодії. Прикладом такої системи є Dynamic Yield (<https://www.dynamicyield.com>);
- автоматизовані системи SEO-оптимізації із самостійним внесенням змін здатні аналізувати позиції сайту в пошукових системах, виявляти можливості для покращення та самостійно вносити необхідні зміни в контент, метатегі та структуру сайту, відстежуючи при цьому ефективність внесених змін та коригуючи стратегію відповідно до отриманих результатів. Прикладом є Alli AI (<https://www.alliai.com>).

**Таблиця 1.**

Приклади автономних систем III

| Система                 | Цільове призначення     | Ключові функції                                             | Особливості реалізації                                                |
|-------------------------|-------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------|
| Agentic AI (agentic.ai) | Автономне UI-тестування | Генерація, виконання і адаптація сценаріїв у реальному часі | Працює як агент, який ухвалює самостійні рішення на основі середовища |

| Система                   | Цільове призначення                          | Ключові функції                                   | Особливості реалізації                                        |
|---------------------------|----------------------------------------------|---------------------------------------------------|---------------------------------------------------------------|
| RoostGPT (Roost.ai)       | AI-генерація варіантів для А/В експериментів | Створення UI-варіантів, інтеграція у пайплайни    | Підтримка CI/CD; фокус на генеративному підході               |
| testRigor (testrigor.com) | Тестування застосунків мовними інструкціями  | Генерація тестів природною мовою, самовідновлення | Орієнтована на нетехнічних користувачів; А/В через інструкції |
| BlinqIO (blinq.io)        | SaaS для багатоваріантного тестування        | Генерація варіантів, поведінковий аналіз          | Механізм самонавчання на основі попередніх результатів        |

Розроблена трирівнева класифікація демонструє чітку траєкторію розвитку інструментів III у веброботі: від асистентів, які доповнюють можливості фахівця, через генератори, які прискорюють створення артефактів, до автономних систем, що перебирають на себе управління комплексними процесами. Такий перехід від інструментальної підтримки до стратегічного партнерства визначає майбутній ландшафт цифрової індустрії та роль людини в ньому.

## 2. Систематизація екосистеми інструментів III за етапами життєвого циклу веброботки.

На основі аналізу сучасних джерел та практичного досвіду розроблено комплексну концептуальну модель екосистеми інструментів III (рис. 1), яка охоплює всі етапи життєвого циклу веброботки та демонструє системну інтеграцію штучного інтелекту у професійні процеси. Дана систематизація враховує як традиційні послідовні етапи SDLC, так і наскрізні процеси, які проходять через весь життєвий цикл продукту.



**Рис. 1.** Концептуальна модель екосистеми інструментів III за етапами життєвого циклу веброботки

### 1. Етап планування. Формування концепції та стратегії

Початковий етап життєвого циклу веброзробки стосується аналізу вимог, дослідження ринку, формування концепції продукту та створення технічного завдання. Сучасні інструменти ШІ значно оптимізують ці процеси, автоматизують збір і оброблення даних, структурування ідей та генерацію документації.

Notion AI (<https://www.notion.com>) трансформує підходи до технічного документування, аналізує неструктуровані описи проєктів, виявляє логічні суперечності у вимогах, генерує узгоджені специфікації та формує пропозиції щодо їх вдосконалення, дозволяє швидко перейти від ідеї до структурованого технічного завдання.

Taskade (<https://www.taskade.com>) використовує можливості генеративного ШІ для мозкових штурмів та структурування ідей, автоматично формує ієрархію понять і виявляє логічні зв'язки між пропозиціями, спрощує формування початкової концепції продукту та покращує спільну роботу команди.

Userpilot (<https://userpilot.com>) спеціалізується на поведінковій аналітиці, використовує алгоритми машинного навчання для виявлення прихованих патернів взаємодії користувачів та дозволяє прогнозувати їхні потреби. Такі дані критично важливі на ранніх етапах UX-дизайну, оскільки забезпечують орієнтацію на реальні очікування цільової аудиторії.

Motion (<https://usemotion.com>) забезпечує інтелектуальне планування проєктів. Система аналізує дедлайни, пріоритети, складність завдань і рівень завантаження членів команди. На основі цих параметрів Motion формує динамічні графіки роботи, автоматизує розподіл задач і підвищує командну продуктивність.

### 2. Етап дизайну. Створення візуальної концепції та UX

Дизайнерський етап традиційно вважався найбільш креативним та найменш придатним до автоматизації. Проте сучасні інструменти ШІ демонструють значний потенціал для трансформації цього процесу.

Galileo AI пропонує новий підхід до UI-дизайну, дозволяє генерувати повноцінні інтерфейси на основі текстових описів функціональності. Система автоматично враховує принципи юзабіліті, доступності та актуальні дизайн-тренди, аналізує велику кількість наявних рішень для створення оптимальних макетів під конкретні завдання.

Uizard.io (<https://uizard.io>) спеціалізується на трансформації ескізів і wireframes у цифрові прототипи. За допомогою алгоритмів комп'ютерного зору платформа розпізнає рукописні елементи та перетворює їх на інтерактивні компоненти, що значно прискорює розробку первинних інтерфейсів.

Midjourney, хоча й не є спеціалізованим вебдизайн-інструментом, широко використовується для створення унікальних графічних елементів, ілюстрацій і концептуального візуального оформлення, дозволяє дизайнерам ефективно досліджувати альтернативні стилістичні рішення на етапі формування візуальної стратегії проєкту.

Relume AI (<https://www.relume.io>) автоматизує створення інформаційної архітектури. Система генерує карти сайтів та базові wireframes на основі аналізу бізнес-цілей та аналізу цільової аудиторії.

### 3. Етап розробки. Програмування та технічна реалізація

Етап безпосередньої розробки програмного коду зазнав найбільших трансформацій завдяки впровадженню інструментів ШІ.

GitHub Copilot не лише генерує код на основі коментарів та контексту, а й здатен до рефакторингу наявного коду, оптимізації алгоритмів та автоматичного виправлення помилок. Система навчена на мільйонах публічних репозиторіїв та розуміє контекст проєкту, пропонує рішення, які відповідають архітектурі та стилю кодування конкретного застосунку.

Tabnine забезпечує розумне автодоповнення коду, що враховує не лише синтаксис мови програмування, а й специфіку проєкту, стиль команди та найкращі практики.

Amazon CodeWhisperer додає критично важливий компонент безпеки, автоматично аналізує код на наявність потенційних вразливостей та пропонує безпечні альтернативи.

Cursor AI реалізує концепцію повністю орієнтованого середовища розробки, в якому ІІІ є активним учасником програмування, здатним підтримувати контекстний діалог із розробником і виконувати складні завдання, зокрема рефакторинг, документування та оптимізацію коду.

#### 4. Етап тестування. Забезпечення якості та стабільності

Тестування традиційно вважається одним із найбільш трудомістких і критично важливих етапів життєвого циклу розробки програмного забезпечення [19]. Проте впровадження інструментів ІІІ суттєво змінює цю сферу, автоматизує рутинні процеси та підвищує адаптивність тестових сценаріїв до змін у застосунку [20].

Testim (<https://www.testim.io>) застосовує алгоритми машинного навчання для створення стійких до змін тестів користувацького інтерфейсу. Завдяки цьому значно знижується потреба в ручному оновленні тестів при кожному редизайні інтерфейсу.

Applitools (<https://applitools.com>) трансформує візуальне тестування за допомогою механізмів ІІІ, виявляє візуальні регресії і розбіжності. Інструмент автоматично порівнює знімки екранів, забезпечуючи кросбраузерну сумісність і виявлення навіть мінімальних відхилень у відображенні контенту.

Mabl (<https://www.mabl.com/>) реалізує підхід до інтелектуального функціонального тестування. Система автономно досліджує вебзастосунок, формує гіпотези щодо потенційних помилок та створює відповідні сценарії без потреби ручного програмування.

Functionize (<https://www.functionize.com>) надає можливість описувати тестові сценарії природною мовою, що значно знижує поріг входу для нетехнічних спеціалістів. На основі цих описів система автоматично генерує програмний код для тестування.

Sentry (<https://sentry.io>) забезпечує постійний моніторинг помилок у продуктивному середовищі, використовуючи ІІІ для автоматичної класифікації, пріоритизації та виявлення першопричин проблем.

#### 5. Етап розгортання. Автоматизація впровадження

Етап розгортання забезпечує перехід програмного продукту зі стадії розробки у продуктивне середовище, а також координацію дій між різними інфраструктурними компонентами. Використання інструментів ІІІ на цьому етапі значно підвищує стабільність, швидкість та ефективність процесу.

GitHub Actions використовує елементи ІІІ для оптимізації CI/CD-пайплайнів. Система аналізує зміни у кодовій базі та автоматично формує відповідні сценарії розгортання, знижуючи ймовірність помилок конфігурації та скорочуючи час доставки оновлень.

GitLab AI доповнює традиційні можливості CI/CD розширеним інтелектуальним управлінням конфігураціями, виявляє потенційні конфлікти та забезпечує їх автоматичне вирішення до моменту розгортання, мінімізує ризики збоїв у продакшн-середовищі.

AWS Auto Scaling із вбудованими механізмами ІІІ забезпечує предиктивне масштабування. Система аналізує історичні патерни навантаження, трафік та інші показники й автоматично адаптує розподіл обчислювальних ресурсів. Такий підхід дозволяє підтримувати продуктивність застосунків на стабільному рівні при оптимальному використанні інфраструктури.

#### 6. Етап підтримки. Моніторинг та еволюція продукту

Післявпровадженська підтримка вебзастосунків поступово еволюціонує від реактивного усунення помилок до проактивного управління життєвим циклом продукту, прогнозування збоїв і постійного вдосконалення функціональності.

Intercom (<https://www.intercom.com/>) та Zendesk (<https://www.zendesk.com>) інтегрують інтелектуальні чат-боти для автоматизованої підтримки користувачів.

Системи здатні вирішувати складні технічні запити, навчатися на основі діалогів із клієнтами та адаптувати відповіді до специфіки запиту, що підвищує якість обслуговування без залучення операторів.

Mintlify (<https://www.mintlify.com/>) та DocuWriter.ai (<https://www.docuwriter.ai/>) автоматизують процес створення та оновлення технічної документації. Інструменти аналізують зміни у кодовій базі та автоматично генерують або оновлюють опис API, інструкції для користувачів і супровідні технічні документи, що суттєво скорочує витрати часу та знижує ризик застарілої документації.

Datadog AI (<https://www.datadoghq.com/>) та New Relic (<https://newrelic.com/>) реалізують концепцію інтелектуального моніторингу, використовуючи алгоритми машинного навчання для виявлення аномалій у поведінці вебзастосунків. Системи прогнозують потенційні проблеми ще до їх появи, здійснюють аналіз причин збоїв, автоматично пропонують або застосовують коригування конфігурацій та масштабування інфраструктури відповідно до навантаження.

#### 7. Наскрізні процеси. Інтеграція ІІІ через весь життєвий цикл

Окрім інструментів, орієнтованих на окремі етапи життєвого циклу веброзробки, сучасна екосистема ІІІ має низку наскрізних процесів, які функціонують неперервно від початкового аналізу вимог до підтримки продукту після впровадження.

Персоналізація користувацького досвіду (UX) реалізується як динамічна система, яка адаптує контент, структуру інтерфейсу та механізми навігації під кожного користувача в режимі реального часу. Системи аналізують поведінкові патерни, демографічні характеристики, контекст сесії та історію взаємодії для формування максимально релевантного досвіду.

A/B/n-тестування трансформувалося в автономні платформи експериментування ІІІ, здатні генерувати гіпотези, автоматично створювати кілька варіантів інтерфейсу або контенту, оцінювати статистичну значущість отриманих результатів та впроваджувати оптимальні рішення без участі людини.

Оптимізація для пошукових систем (SEO) зазнала значних змін завдяки ІІІ. Сучасні інструменти автоматично генерують метатеги, аналізують ключові слова та структуру контенту, прогнозують ефективність змін й адаптують вміст до специфіки алгоритмів ранжування пошукових систем.

Забезпечення доступності (a11y) вебресурсів також автоматизується. Рішення ІІІ сканують код та візуальний інтерфейс, виявляють невідповідності стандартам WCAG (Web Content Accessibility Guidelines), пропонують або виконують автоматичне виправлення порушень, а також генерують альтернативні описи для графічних елементів і відео, що сприяє інклюзивності цифрових сервісів.

Загалом проведений аналіз демонструє, що інтеграція ІІІ у веброзробку виходить за межі автоматизації окремих завдань і формує цілісну, взаємопов'язану екосистему. Інструменти ІІІ не лише оптимізують усі послідовні етапи життєвого циклу програмного забезпечення, від планування до підтримки, а й забезпечують реалізацію наскрізних процесів, таких як персоналізація, SEO та автоматичне A/B-тестування, які функціонують неперервно. Таким підходом окреслюється фундаментальний перехід від фрагментованого виконання завдань до створення адаптивних, самонавчальних цифрових систем, в яких ІІІ відіграє роль невід'ємного компонента розробницького процесу.

**Результати та обговорення.** У ході дослідження отримано два ключові результати: розроблено трирівневу класифікацію інструментів ІІІ за рівнем когнітивної взаємодії та створено концептуальну модель екосистеми цих інструментів у розрізі життєвого циклу веброзробки. Запропонована класифікація на інструменти-асистенти, інструменти-генератори та автономні системи дозволяє структурувати ринок технологій ІІІ та оцінити ступінь їхнього впливу на робочі процеси. Інструменти-асистенти (Tabnine, Amazon Q Developer) є базовим рівнем інтеграції, що підвищує індивідуальну

продуктивність, але зберігає повний контроль за розробником. Їх впровадження не вимагає суттєвої зміни методології роботи. Інструменти-генератори (v0.dev, Galileo AI) знаменують якісний перехід, де ШІ стає творчим партнером, здатним створювати повноцінні артефакти (код, дизайн-макети), змінює роль фахівця з «творця» на «редактора» та «верифікатора», що вимагає нових навичок для ефективної взаємодії й формулювання запитів. Автономні системи (Agentic AI, Dynamic Yield) є вищим рівнем, який делегує ШІ ухвалення рішень на основі бізнес-цілей, трансформує підходи до тестування, оптимізації та персоналізації, перетворюючи їх на самокеровані процеси. Запропонована класифікація надає компаніям чіткий фреймворк для стратегічного планування: від простих інвестицій в асистентів для підвищення ефективності до довгострокових вкладень в автономні системи для досягнення конкурентної переваги.

Розроблена концептуальна модель візуалізує характер інтеграції ШІ, підтверджуючи, що його вплив не обмежується етапом написання коду. Інструменти ШІ оптимізують кожен етап життєвого циклу веброзробки: від аналізу вимог (Notion AI) та прототипування (Uizard.io) до автоматизованого розгортання (GitLab AI) та проактивного моніторингу (Datadog AI). Особливе значення мають наскрізні процеси (SEO, а11y, A/B-тестування), які демонструють найвищий рівень зрілості інтеграції, оскільки вони функціонують неперервно та комплексно покращують продукт.

Поєднання класифікації та моделі екосистеми дозволяє зробити висновок, що ефективне впровадження ШІ вимагає не точкових рішень, а побудови цілісної стратегії, яка враховує рівень автономності інструментів на кожному етапі розробки.

**Висновки.** У статті проведено комплексний аналіз інтеграції ШІ-технологій у сучасну веброзробку та вебдизайн. Запропонована класифікація поділяє інструменти на три рівні за ступенем когнітивної взаємодії та автономності: асистенти, генератори та автономні системи. Така структура дає змогу оцінювати зрілість технологій та планувати їх впровадження. Створена концептуальна модель екосистеми інструментів ШІ наочно демонструє інтеграцію ШІ в усі етапи життєвого циклу веброзробки планування, дизайну, розробки, тестування, розгортання та підтримки, а також у наскрізні процеси, підтверджує перехід від локальної автоматизації до побудови комплексних інтелектуальних систем. Запропоновані класифікація та модель екосистеми слугуватимуть методичною основою для ІТ-компаній при ухваленні стратегічних рішень щодо впровадження ШІ, оптимізації робочих процесів та обґрунтування інвестицій у технології. Перспективи подальших досліджень лежать у площині розробки стандартизованої методики кількісної оцінки економічної ефективності від впровадження ШІ-інструментів, а також у дослідженні етичних аспектів використання автономних систем у творчих та інженерних процесах.

#### Список літератури

1. Upadhyaya N. Artificial Intelligence in Web Development: Enhancing Automation, Personalization, and Decision-Making. *International Journal of Advanced Research in Science, Communication and Technology*. 2024. Vol. 4(1). DOI: <https://doi.org/10.48175/ijarsct-19367>.
2. Концепція розвитку штучного інтелекту в Україні: розпорядження КМУ від 2 грудня 2020 р. № 1556-р. URL: <https://zakon.rada.gov.ua/laws/show/1556-2020-p>
3. Vangavolu S. The Future of AI-Powered Web Development and Low-Code Platforms. *International Journal of Innovative Research in Science, Engineering and Technology*. 2025. Vol. 14. Issue 1. DOI : <https://doi.org/10.15680/ijirset.2025.1401111>.
4. Sonali S. J., Sonali S. G. The Impact of AI on Web Development. *International Journal of Scientific Research in Modern Science and Technology*. 2024. Vol. 3(8). P. 7-12. DOI: <https://doi.org/10.59828/ijrmst.v3i8.240>
5. Ayyagiri A., Goel P., Renuka A. Leveraging AI and Machine Learning for Performance Optimization in Web Applications. *Darpan International Research Analysis*. 2024. Vol. 12(2). P. 199-218. DOI: <https://doi.org/10.36676/dira.v12.i2.85>

6. Cherukuri B. Enhancing Web Application Performance with AI - Driven Optimization Techniques. *International Journal of Science and Research (IJSR)*. 2021. Vol. 100, no. 2. P. 1779-1788. DOI: <https://doi.org/10.21275/sr21021103246>.
7. Kosuru V. (2024). Integration of Artificial Intelligence with Web Development. *International Journal of Innovative Science and Research Technology (IJISRT)*. 2024. Vol. 9. Issue 8. P. 208-210. DOI: <https://doi.org/10.38124/ijisrt/ijisrt24aug061>
8. Manikantam S., Akhil P., Reddy K., Reddy G., Hariharan S., Kekreja V. Enhanced automated web scraping tool with proliferation of AI techniques. *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*, Nagpur, India, 2024. P. 1-5. DOI: <https://doi.org/10.1109/ICICET59348.2024.10616333>.
9. Zhang K., Kumar V., Zhang M. Interactive Web: Leveraging AI-Driven Code Generation to Simplify Web Development Algorithms for Novice Programmers. *Artificial Intelligence and Big Data Trends*. 2025. DOI: <https://doi.org/10.5121/csit.2024.150107>.
10. Meshri A. Design to Code. *International Journal of Scientific Research in Engineering and Management (IJSREM)*. 2025. Vol. 9. Issue 4. P. 1-4. DOI: <https://doi.org/10.55041/ijisrem44881>.
11. Yauheni M. How Artificial Intelligence will change Web Design: opportunities and challenges. *International Journal of Latest Engineering and Management Research (IJLEMR)*. 2023. Vol. 8. Issue 6. P. 51-54. DOI: <https://doi.org/10.56581/ijlemr.8.6.51-54>.
12. Wang P. The Influence of Artificial Intelligence on Visual Elements of Web Page Design under Machine Vision. *Computational Intelligence and Neuroscience*. 2022. DOI: <https://doi.org/10.1155/2022/4328400>
13. Iqbal H. Sarker. AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems. *SN Computer Science. SCI*. Vol. 3, Issue 158, 2022. DOI: <https://doi.org/10.1007/s42979-022-01043-x>
14. Supolia M. AI-Driven Personalization and Recommendations for Web Design Resources. *International Journal of Scientific Research in Engineering and Management (IJSREM)*. Vol. 09, Issue 04, 2025. DOI: <https://doi.org/10.55041/ijisrem44812>
15. Kozhakhmetova A., Mamyrbayev A., Zhidebekkyzy A. Assessing the impact of artificial intelligence on project efficiency enhancement. *Knowledge and Performance Management*. 2024. Vol. 8. Issue 2. P. 109-126. DOI: [http://dx.doi.org/10.21511/kpm.08\(2\).2024.09](http://dx.doi.org/10.21511/kpm.08(2).2024.09)
16. Cherukuri B. Enhancing Web Application Performance with AI - Driven Optimization Techniques. *International Journal of Science and Research (IJSR)*. 2021. Vol. 10. Issue 2. P. 1779-1788. <https://www.ijsr.net/getabstract.php?paperid=SR21021103246>. DOI: <https://doi.org/10.21275/sr21021103246>.
17. Anitha C., Gupta N., Chintala B., Pilli D., Kesavan E., Ali M. Impact of Artificial Intelligence on Software Development Processes. *Journal of Information Systems Engineering and Management*. 2025. Vol. 10, No. 25. P. 431-437. DOI: <https://doi.org/10.52783/jisem.v10i25s.4039>.
18. Kapoor E. The Impact of Artificial Intelligence on Modern Website Design. *International Journal for Multidisciplinary Research*. 2024. Vol. 6. Issue 3. DOI: <https://doi.org/10.36948/ijfmr.2024.v06i03.23234>.
19. Трофименко О.Г., Дика А.І., Лобода Ю.Г. Аналіз уразливостей та проблем безпеки вебзастосунків. *Системні технології*. 2023. № 3(146). С. 25-37. DOI: <https://doi.org/10.34185/1562-9945-3-146-2023-03>.
20. Трофименко О.Г., Дика А.І., Лобода Ю.Г. Аналіз інструментів тестування вебзастосунків. *Кібербезпека: освіта, наука, техніка*. 2023. № 4(20). С. 62-71. DOI: <https://doi.org/10.28925/2663-4023.2023.20.6271>.

Ю.Г. Лобода, О.Г. Трофименко, С.Ю. Манаков, В.І. Гура

## ARTIFICIAL INTELLIGENCE IN MODERN WEB DEVELOPMENT AND WEB DESIGN: MULTILEVEL CLASSIFICATION AND SYSTEMATIZATION

Yu.G. Loboda, O.G. Trofymenko, S.Yu. Manakov, V.I. Hura

National University “Odesa Law Academy”  
23, Fontans'ka doroga st., Odesa, 65009, Ukraine  
Emails: loboda@onua.ua, trofymenko@onua.edu.ua

The research is devoted to a comprehensive analysis of the impact of artificial intelligence (AI) technologies on the processes of modern web development and web design. The relevance of the topic is driven by the rapid development of AI technologies and the need for their systematic integration into professional software development processes. The purpose of the research is to develop a multi-level classification of AI tools based on their degree of autonomy and cognitive interaction, as well as to systematize the ecosystem of AI tools according to the stages of the software development life cycle. The research methodology is grounded in a systematic approach, which includes an analysis of current scientific literature, a functional classification of AI tools, and a comparative analysis of real-world cases of AI usage in web development. The scientific novelty of the research lies in creating a comprehensive systematization of AI tools for web development, considering the cognitive aspects of interaction. The main results of the research include the creation of a three-level classification of AI tools (assistants, generators, and autonomous systems), as well as the development of a conceptual model of the AI solutions ecosystem, structured according to the stages of the web development life cycle. This developed conceptual model visualizes the nature of AI integration, confirming that its impact extends beyond the stage of code writing. AI tools optimize each stage of the web development life cycle: from requirements analysis (Notion AI) and prototyping (Uizard.io) to automated deployment (GitLab AI) and proactive monitoring (Datadog AI). End-to-end processes (SEO, accessibility, A/B testing) are of particular importance, as they demonstrate the highest level of maturity of integration, functioning continuously and improving the product in a comprehensive manner. The combination of classification and ecosystem models allows for the conclusion that the effective implementation of AI requires not just point solutions, but the development of a holistic strategy that considers the level of autonomy of tools at each stage of development. The practical significance of the results lies in the ability to use the proposed classification for informed decision-making when selecting tools, optimizing development processes, improving team productivity, and forming strategic decisions for the introduction of AI technologies in IT projects. These results can be applied both in the professional activities of IT companies and in scientific research in the field of software engineering.

**Keywords:** artificial intelligence, web development, web design, development life cycle, cognitive interaction, automation.