// КОРПОРАТИВНЕ AI-НАВЧАННЯ

Навчання, яке проводиться і оцінюється як інженерна робота.

Ми навчаємо команди розробки працювати з AI-агентами так, як інженери працюють із будь-якою іншою виробничою системою: із зафіксованими вимогами, верифікацією та вимірюваними результатами — у тому самому середовищі, де вони виконують реальну роботу.

Вендор-нейтральність за задумом — ми навчаємо вашого робочого процесу, а не лише інструменту, тож ви можете змінювати інструменти будь-коли. Наш основний інструмент — Claude Code, але програму можна провести на Codex, Cursor, Copilot, OpenCode чи будь-якому іншому популярному інструменті.

Подивитися, як ми це доводимо →

- програми збираємо під ваші реальні потреби
- усі практичні завдання на основі реальних робочих сценаріїв
- каталог з десятків відпрацьованих тем
- групи від кількох осіб до сотень людей
- чат підтримки, записи занять, детальні матеріали для самостійної роботи

Воркшопи проходять на вашому стеку та ваших артефактах; практичні завдання здаються як pull (merge) request-и; результати живуть у журналі прогресу, що піддається аудиту, із сертифікатами, які можна перевірити онлайн.

// МЕТОД ВЕРИФІКОВАНОГО НАВЧАННЯ

Метод верифікованого навчання

Навчання побудоване на формуванні практичних навичок у реальному середовищі і на задачах, максимально наближених до реальних. Як результат навчання зараховується реальна виконана робота належної якості. На цьому методі працює кожна програма нижче.

01 **Вхідне опитування до першої сесії**

Навчальна програма налаштовується під інструменти та запити вашої команди, різний досвід та реальні проблемні місця; ще до старту ви отримуєте портрет учасників.

02 **Воркшопи на вашому стеку**

2-годинні сесії, онлайн або на місці, з орієнтовним розподілом 50/50 між теорією та практикою.

03 **Домашні завдання як merge request-и**

Їх можна рев'ювити, порівнювати та відстежувати в часі. Різні рівні складності; можна навчати людей зовсім без досвіду програмування — або апскілити людей сеньйорного рівня.

04 **Механізм оцінювання**

Рубрика додається до кожного завдання в момент видачі; AI-рев'юер оцінює за нею; тренер додатково розбирає рев'ю, граничні випадки, апеляції щодо оцінок і особисто перевіряє кожен фінальний проєкт. За запізнілі роботи передбачено штраф за опублікованим правилом — це мотивує здавати вчасно.

05 Живий журнал прогресу

Кожна здача й кожна оцінка відстежуються протягом усього курсу. Керівництво за замовчуванням бачить агрегати когорти; повна поіменна деталізація вмикається за вашим вибором.

06 Сертифікати

Обчислюються на основі практичної успішності, а не «видаються»; кожен можна перевірити онлайн за номером і QR-кодом. Це управлінський інструмент: свідчення спроможності кожної людини, а не лише відвідуваності занять.

07 Звіт після кожної сесії

Після кожного воркшопу — письмовий огляд: що здано, що зараховано і що виправити до наступної сесії.

08 Детальні матеріали

Слайди, записи занять, детальні покрокові інструкції тощо.

Почніть з малого

Вхід із найменшим ризиком — пілот: вхідне опитування + один 2-годинний воркшоп + результат роботи групи. Ви бачите метод і базовий рівень вашої команди ще до рішення про навчання всіх працівників.

// ПРОГРАМИ

П’ять програм, один метод

Програми гнучкі: теми нижче — це каталог, і кожну групу ми збираємо з нього на старті під ваші потреби та рівень зрілості ваших спеціалістів.

Кількість сесій є гнучкою

Плани нижче — приклади форматів, які ми вже проводили; та сама програма може бути коротким інтенсивом або розширеним кількамісячним курсом — план узгоджуємо перед стартом, тривалість масштабується.

Вміємо працювати з дуже великими групами

Забезпечувати підтримку групи і перевіряти практичні завдання допомагають AI-агенти, цей підхід добре масштабується: та сама програма працює і для команди з десяти людей, і для десятків чи сотень учасників одночасно, причому кожна робота все одно оцінюється індивідуально і кожен учасник отримує персональний фідбек.

ПРОГРАМА	АУДИТОРІЯ	ПРИКЛАД ПЛАНУ	ЧАС КОМАНДИ (ДЛЯ ПРИКЛАДУ

			ПЛАНУ)
1. Агентна інженерія для розробників	розробники, техліди, архітектори	12 × 2 год, зібрано з каталогу на 29 тем	~40-50 год на людину за ~3 місяці
2. Агентний QA	QA-інженери	9 × 2 год + Сесія 0	~30 год на людину
3. Агентна бізнес-аналітика	BA, продуктові аналітики	9 × 2 год + Сесія 0	~30 год на людину
4. Агентні QA + BA (об'єднана програма)	QA та BA однією когортою	12 × 2 год + спільний капстоун; кожен відвідує 10	~32 год на людину
5. AI-інженерія для лідерів	CTO, VP, керівники, архітектори	2–6 сесій × 2 год із переліку на 7 тем	~4-12 год на людину

Ціноутворення

За когорту: базова ставка за сесію плюс плата за кожного учасника за верифікацію (оцінювання, журнал, сертифікати). Точні ставки — у комерційній пропозиції після вхідного опитування.

// ПРОГРАМА 1 — НАЙГЛИБШИЙ ТРЕК

Агентна інженерія для розробників

AI-агенти в щоденній роботі розробника.

ПІСЛЯ ЦІЄЇ ПРОГРАМИ

Ваша команда випускає зміни, виконані за допомогою агентів, через верифікований процес: записаний намір, курований контекст, рев'ю за рівнем ризику.

Інтенсивність навчання

Рекомендований темп — один воркшоп на тиждень, що дозволяє тримати баланс між інтенсивністю і прогресом. Інтенсивність

Сертифікати: на вимогу клієнта багаторівневі (з відзнакою, звичайні, лише теорія). Усі обчислюються з журналу й перевіряються за QR-кодом.

можна змінювати, проводити змагання проєктів тощо.

Типові конфігурації груп

`Core-12 (типовий вибір)`

F1–F8, одна з F9/F10, F11, плюс деякі просунуті сесії, обрані на старті (найчастіше A1 та A6).

`Advanced-12`

A1–A12, для команд, які проходять діагностику рівня (домашнє завдання фундаментального рівня на складності L2) або завершили Core-12.

`Brownfield-інтенсив (6)`

Територія F11 + A1 + A2: археологія legacy-коду, безпечні зміни, оцінювання згенерованого коду, рев'ю PR і командні правила, робочі процеси з MCP.

`Навчання, що переходить у консалтинг`

Навчання плюс супровідний консалтинг на вашій реальній системі — ваші розробники за кермом, тренер скеровує і допомагає зі складними моментами.

`Навчання навколо міграції`

Навчальна програма, викладена через реальний міграційний проєкт у процесі AI-трансформації команди (наприклад, no-code-процеси, перебудовані агентами в код).

Рівень 1 — Фундаментальний (12 тем)

Мета: кожен розробник безпечно делегує рутинну роботу агенту — із записаним наміром, курованим контекстом і верифікацією кожного результату. Кожна сесія має спостережувану ціль, і критерії домашнього завдання виводяться з неї. Матеріали завжди оновлюються перед заняттям відповідно до змін у нових інструментах та підходах.

WS	ТЕМА	ЩО ОХОПЛЮЄ
F1	**Стан AI-розробки**	Де насправді перебуває агентна розробка: докази, чому одні проєкти з AI злітають, а інші провалюються, що змінюється в ролі розробника.

F2	**Промпт-інжиніринг і безпека**	Надійний промптинг; що можна й чого не можна передавати в інструмент; гігієна секретів; захист від prompt injection; робота з синтетичними замінниками даних.
F3	**Контекст-інжиніринг**	Керування тим, що бачить агент: контекст репозиторію, вибіркове підключення, керування деградацією контексту, надійні результати з першої спроби.
F4	**Оцінювання та вибір агентних інструментів**	Ландшафт сучасних інструментів, їх порівняння за різними критеріями.
F5	**Правила та команди агентної IDE**	AGENTS.md як єдине джерело правди, файли правил, власні slash-команди: агент за замовчуванням дотримується конвенцій вашої команди.
F6	**Agent Skills**	Пакування повторюваних робочих процесів у багаторазові skills, якими можна ділитися, — важливий елемент командної експертизи.
F7	**MCP**	Підключення готових та розробка власних MCP для доступу до зовнішніх ресурсів (трекер задач, документація, бази даних, внутрішні API; керування дозволами).
F8	**Spec-Driven Development: записуємо намір**	Цілі, обмеження, definition of done — специфікації, за якими агент може діяти, а рев'юер — оцінювати. Один фреймворк глибоко (зазвичай OpenSpec), інші у порівнянні (Spec Kit, BMAD).
F9	**Фронтенд-розробка з AI**	Повсякденні агентні процеси на фронтенді: каркаси компонентів, рефакторинг, стилізація та дизайн-системи, design-to-code (від макета до робочого компонента), UI-тести, що їдуть разом із кожною зміною.
F10	**Бекенд-розробка з AI**	Повсякденні агентні процеси на бекенді: API, шари даних, міграції, дебагінг, інтеграційні тести, що їдуть разом із кожною зміною.

F11	**Код-рев'ю з AI**	Читати AI-дифи як senior-рев'юер; рев'ю за масштабом наслідків (blast radius); рев'ю PR і командні правила, автоматизація рев'ю.
F12	**Верифікована документація**	README, ADR та онбординг-гайди, які агенти генерують і підтримують разом із кодом — і які проходять той самий рев'ю-пайплайн, що й сам код. Структура і підтримка оптимальної документації для роботи AI-агентів.

Рівень 2 — Просунутий

Мета: команда перестає промптити вручну й починає інженерити систему, яка виробляє код, — оркестровану, верифіковану, здатну вдосконалювати саму себе.

WS	ТЕМА	ЩО ОХОПЛЮЄ
A1	**Великі проєкти та legacy**	Археологія коду, безпечні зміни в системах, яких ніхто не розуміє повністю, компактні індекси контексту.
A2	**Оцінювання AI-згенерованого коду**	Борг розуміння (comprehension debt) у brownfield-роботі: чому довіряти, що перевіряти заново, як зберегти розуміння, коли більшість коду згенерована.
A3	**Спільна агентна інфраструктура — harness**	Skills, хуки, конфігурація MCP, субагенти, довготривала пам'ять — спільна платформа, завдяки якій команда працює в синергії.
A4	**Мультиагентна оркестрація**	Розділення maker/checker — перевіряльник працює на іншій моделі з незалежним контекстом, з ескалацією до людини за необхідності; ізольовані worktree для роботи агентів; запуск паралельних агентів.
A5	**Цикли та інженерія автономності**	Довготривалі агенти з явними умовами зупинки, нічні цикли під наглядом ґардрейлів, мерджі лише за доказами.

A6	**Верифікація в масштабі**	Увесь верифікаційний стек в одному місці: гетерогенні AI-рев'юери, red/green TDD, механіка мутаційного тестування, детерміновані CI-ґейти, які несуть навантаження, що його вже не витягує людське читання.
A7	**Evals — захист агентів від регресій**	Набори evals, які охороняють промпти так, як тести охороняють код; кожна невдача стає постійним правилом, skill-ом або хуком (ефект храповика).
A8	**Архітектура та DevOps з AI**	Архітектурні рішення, infrastructure-as-code, проєктування CI/CD з агентами в контурі.
A9	**Тестування та QA з AI**	Погляд розробника: генерація тестових наборів, цінність яких доведена на навмисно закладених дефектах (механіка верифікації живе в A6), плюс економіка генерації тестів.
A10	**Економіка AI та FinOps**	Економіка токенів, маршрутизація моделей, бюджети та квоти, частка успіху з першої спроби, моделювання TCO і ROI, який визнає CFO.
A11	**Software Factories — стан справ**	Що таке фабрики програмного забезпечення і хто вже їх запускає: типи фабрик, світовий досвід, чому фабрики провалюються, практики, безпека та ґавернанс, економіка автономного виробництва коду.
A12	**Software Factories — як збудувати фабрику**	Принципи фабрики на прикладі реальної робочої фабрики.
A13	**Greenfield за задумом — архітектура, дружня до агентів**	Старт нових систем так, щоб агенти залишалися ефективними в міру зростання: структура репозиторію, технічний стек, конвенції, документація й тестові каркаси як повноцінна частина проєктування.
A14	**Створення продакшн-агентів**	Агенти як функції продукту: агентні SDK, проєктування інструментів, ґардрейли, спостережність та evals для агентів, які ви відвантажуєте користувачам.

A15	**RAG та інженерія баз знань**	Пайплайни пошуку по вашій документації та коду: чанкування, ембедінги, гігієна документів, обґрунтовані відповіді з цитатами — і коли RAG — неправильний інструмент.
A16	**Реагування на інциденти та дебагінг з AI**	Археологія логів і телеметрії з агентами, дебагінг через гіпотези, чорнові постмортеми, що скорочують час до діагнозу на реальних інцидентах.
A17	**Політика використання AI, безпека та ґавернанс**	Робоча політика того, що може потрапляти в який інструмент: класифікація даних, погодження інструментів, аудиторські сліди, безпечні налаштування за замовчуванням. Також доступно як окремий модуль для команд безпеки.

// ПРОГРАМА 2

Агентний QA

AI-агенти в роботі QA.

ПІСЛЯ ЦІЄЇ ПРОГРАМИ

Як AI-агенти допомагають вашій QA-команді.

ДЛЯ КОГО

Мануальні та automation QA-інженери.

ФОРМАТ

Сесія 0 — технічний онбординг (основи git, середовище, чекліст доступів).

WS	ТЕМА	ЩО ОХОПЛЮЄ
Q1	**Принципи AI-агентів: делегування, контекст і вартість**	Робоча ментальна модель агентів, практичний промптинг.
Q2	**Від acceptance criteria до тестових артефактів**	Acceptance criteria → чеклісти, тест-кейси та трасованість — згенеровані, а потім верифіковані.

Q3	**Екосистема, правила та ґардрейли**	Підключений, керований агент: трекер, тест-менеджмент і база знань через MCP з безпечними дозволами; командна бібліотека правил і skills.
Q4	**Специфікація як контракт (OpenSpec)**	Тестування за записаним наміром, а не за фольклором: специфікація як артефакт, дотримання якого QA вимагає від усіх.
Q5	**Технічний контекст із коду: мапа систем та інтеграцій**	Мапи систем та інтеграцій, побудовані агентом, — розуміння, що з чим взаємодіє, без читання коду.
Q6	**Тест-дизайн, дослідницьке тестування і тестові дані з AI**	AI-асистований тест-дизайн і рев'ю до будь-якої автоматизації; агент як партнер у дослідницькому тестуванні; генерація тестових даних.
Q7	**Автотести природною мовою та їх валідація**	Тести, написані зі сценаріїв простою мовою — Playwright як еталонний стек, метод переноситься на ваш фреймворк — плюс генерація page-object-ів для наявних застосунків. Валідація: пайплайн ламає застосунок, і ваш набір мусить це помітити.
Q8	**API-тести, регресійні набори та CI/CD**	Тестування на рівні API, регресійні тести.
Q9	**Діагностика дефектів і чорновий фікс**	Трикрокова дисципліна діагностики, що завершується чорновим фіксом у вигляді pull (merge) request-а, — QA пропонує, розробники вирішують. Включно з баг-репортами, за якими може діяти агент: відтворення, контекст.

ФІНАЛЬНИЙ ПРОЄКТ

Ви отримуєте підготовлений застосунок із навмисно закладеними дефектами (класи: логіка / межі / інтеграції; ін'єктовані пайплайном оцінювання, категорії не розголошуються). Потрібно здати тестовий набір і пакети діагностики, які ловлять щонайменше 60% закладених дефектів. Побудова власного невеликого продукту з AI пропонується як розширений варіант на рівні L3.

// ПРОГРАМА 3

Агентна бізнес-аналітика

AI-агенти в роботі бізнес-аналітика.

> ПІСЛЯ ЦІЄЇ ПРОГРАМИ
>
> Ваші аналітики створюють вимоги, за якими агенти, тестувальники та розробники можуть діяти напряму: вони спираються на реальну систему, а їхня якість оцінюється за рубрикою рев'юера.

ДЛЯ КОГО

Бізнес-аналітики, продуктові аналітики, product owner-и — на початковому рівні складності програмувати не потрібно.

ФОРМАТ

Сесія 0 — технічний онбординг; оцінювані домашні завдання; фінальний проєкт.

WS	ТЕМА	ЩО ОХОПЛЮЄ
B1	**Принципи AI-агентів: делегування, контекст і вартість**	Робоча ментальна модель агентів, практичний промптинг і вартість.
B2	**Від вхідних даних стейкголдерів до acceptance criteria**	Від сирих розмов, документів і записів зустрічей до acceptance criteria, за якими можуть діяти і агенти, і тестувальники, і розробники — включно з процесною документацією, складеною безпосередньо із записів.
B3	**Екосистема, правила та ґардрейли**	Трекер і база знань як робоча пам'ять агента через MCP; бібліотека команди аналітиків: шаблони сторі, рубрики рев'ю, ґардрейли генерації — і чернетки комунікацій зі стейкголдерами, узяті з трекера, а не написані з пам'яті.
B4	**Специфікація як контракт (OpenSpec)**	Специфікація як артефакт, на якому тримається все нижче за течією, — написана так, щоб агент міг за нею діяти.
B5	**Технічний контекст із коду: мапа систем та інтеграцій**	Писати вимоги, що враховують імплементацію.

B6	**Discovery та декомпозиція**	Від формулювання проблеми до епіків і сторі з агентом — швидші чернетки, людське судження щодо обсягу. Сесія завершується генератором епіків/сторі, запакованим у багаторазовий командний skill.
B7	**Прототипування зі сторі**	Клікабельні прототипи, згенеровані зі сторі й валідовані зі стейкголдерами до першого рядка продакшн-коду, плюс пакети UAT-сценаріїв із тих самих сторі.
B8	**Аналіз даних через генерацію скриптів**	Відповіді на питання про дані — скриптами, які пише агент, а валідує аналітик; повноцінна сесія, а не доважок до програми.
B9	**AI як рев'юер: вимірювана якість вимог**	Агент рев'ює специфікації за рубрикою до того, як їх побачать розробники, — якість вимог стає числом.

ФІНАЛЬНИЙ ПРОЄКТ

Повний пакет від формулювання проблеми до сторі, прототипу, UAT-сценаріїв і задач у трекері. Випробування: наприкінці тренер ін'єктує дві зміни вимог; усі артефакти нижче за течією мають бути оновлені так, щоб AI-рев'юер не знайшов жодної суперечності.

// ПРОГРАМА 4

Агентні QA + BA (об'єднана програма)

QA та BA навчаються як одна когорта.

ПІСЛЯ ЦІЄЇ ПРОГРАМИ

Одна спільна мова й одна командна бібліотека для QA та BA — і handoff «вимоги → тести», відпрацьований на спільних артефактах у капстоуні.

ДЛЯ КОГО

QA-інженери та бізнес-аналітики однієї організації, які навчаються як одна когорта.

ФОРМАТ

Спільні сесії BA + QA.

WS	СЕСІЯ	ХТО ВІДВІДУЄ
C1	Принципи AI-агентів: делегування, контекст і вартість	усі
C2	Аналіз і уточнення вимог: від acceptance criteria до тестових артефактів	усі
C3	Єдина екосистема: агент, трекер і база знань через MCP	усі
C4	Специфікація як контракт: Spec-Driven Development з OpenSpec	усі
C5	Правила, skills і ґардрейли. Командна бібліотека	усі
C6	Технічний контекст із коду: мапа систем та інтеграцій	усі
QA1	Автотести природною мовою та їх валідація	QA
QA2	API-тести, регресійні набори та CI/CD	QA
QA3	Діагностика дефектів і чорновий фікс	QA
BA1	Discovery та декомпозиція	BA
BA2	Прототипування й аналіз даних через генерацію скриптів	BA
BA3	AI як рев'юер: вимірювана якість вимог	BA
C7	Капстоун-handoff: QA тестує те, що специфікувала BA — спільне фінальне рев'ю на спільних артефактах	усі

Чому об’єднана програма краща за два окремі тренінги

- Handoff справжній: у капстоуні фінали QA проганяються на сторі та UAT-сценаріях, які створив BA-трек.
- Одна командна бібліотека правил, skills і промптів — збудована разом, у спільній власності, переживає курс.
- Фундамент ідентичний, тому об'єднана програма — це 13 сесій замість 18.

ФІНАЛЬНІ ПРОЄКТИ

Обидва рольові фінали (QA: впіймано ≥60% закладених дефектів; BA: тест консистентності з ін'єкцією змін), з'єднані в капстоуні, плюс спільна командна бібліотека як результат когорти.

// ПРОГРАМА 5

AI-інженерія для лідерів

Стратегічні сесії для інженерного керівництва.

ПІСЛЯ ЦІЄЇ ПРОГРАМИ

Обґрунтована стратегія розробки з AI: що автоматизувати, що верифікувати, скільки це коштує і яка політика робить це безпечним. Побудована на доказах, а не на вендорських презентаціях.

ДЛЯ КОГО

CTO, VP of Engineering, керівники делівері, провідні архітектори. Часто це перша співпраця: брифінг, який визначає обсяг майбутніх когорт.

ФОРМАТ

2–6 сесій × 2 год, зібраних перед стартом із тем нижче; senior-аудиторія, вхідне опитування до першої сесії. Без домашніх завдань; натомість — executive summary та мапа готовності.

WS	ТЕМА	ЩО ОХОПЛЮЄ
L1	**Software Factories — стан справ**	Як сьогодні виглядає автономне виробництво ПЗ: докази, хто вже запускає фабрики, чому більшість спроб провалюється, безпека та ґавернанс.
L2	**Як збудовані справжні фабрики**	Принципи фабрики на робочій еталонній фабриці, holdout-верифікація наживо, розділення обов'язків — включно з живим запуском фабрики з реальними витратами на обчислення.
L3	**Економіка розробки з AI**	Економіка токенів, маршрутизація моделей, бюджети, частка успіху з першої спроби, TCO і ROI — крива вартості та як її перевернути.

L4	**Оцінка трудомісткості в агентну еру**	Чому сторі-поїнти й моделі velocity ламаються, коли агенти стискають час імплементації; оцінювання за вартістю верифікації та рев'ю, а не за часом написання коду; бюджети токенів поруч із людино-годинами; прогнозування від частки успіху з першої спроби — і планування навколо нового вузького місця: спроможності рев'ю.
L5	**Новий SDLC: процес і ролі**	Як виглядає життєвий цикл, коли більшість коду пишуть агенти: spec-first потік із верифікаційними ґейтами; як зміщуються ролі розробника, QA, BA та архітектора; топологія команд, найм і сеньйорність у новому пайплайні.
L6	**Ефективна командна робота з агентними артефактами**	Команда на спільній агентній інфраструктурі: командні бібліотеки правил, skills і промптів; власність і обов'язок рев'ю, коли більшість змін створюють агенти; онбординг і потік знань. Артефакти, які накопичують цінність, а не гниють у вікі.
L7	**Політика, ґавернанс і план впровадження**	Політика використання AI, класифікація даних, верифікаційні ґейти, готовність команд — і в якій послідовності запускати рольові програми вище.

// КОРПОРАТИВНЕ AI-НАВЧАННЯ

Оберіть програму — або нехай її підбере вхідне опитування

Розкажіть нам про ролі, стек і обмеження. Ми опитуємо команду до першої сесії та формуємо програму за тим, що кажуть дані.

Записатися на ознайомчий дзвінок →

Подивитися докази



Цей каталог надруковано з живої сторінки — актуальна версія завжди за адресою
dev-ai.dev/uk/training



Зразки сертифікатів і живе демо журналу прогресу:
dev-ai.dev/uk/proof



Записатися на ознайомчий дзвінок:
dev-ai.dev/uk#contact