Впровадження агентної інженерії

Наскрізна Stage-Up трансформація інженерної організації: від хаотичного використання ШІ — до верифікованої фабрики програмного забезпечення, яка нарощує віддачу.



// DOU DAY 2026 · АГЕНТНА ІНЖЕНЕРІЯ НАЖИВО

В'ячеслав Колдовський, Ph.D.

Лід трансформації ШІ-інженерії · викладач, спікер конференцій і дослідник агентної інженерії

// 01 — ЩО МИ ПРОПОНУЄМО

Зробіть агентну інженерію типовим способом роботи команд — безпечно й вимірювано.

dev-ai.dev перетворює хаотичне використання ШІ на верифікований, відтворюваний спосіб, у який ваші команди будують софт з агентами, — доведений на вашій кодовій базі, а не на слайдах.

// ЧОТИРИ СПОСОБИ ПОЧАТИ

Оцінити

години ⟶ тижні

Діагностика з фіксованим обсягом: профіль зрілості команд, метричний базлайн, послідовна дорожня карта.

Навчити

2 сесії ⟶ 12 воркшопів

Навчальні програми з оцінюванням — від двох сесій до когорти з 12 воркшопів.

Озброїти

2-6 тижнів

Артефакти агентної інженерії, закомічені у ваш репозиторій і закріплені тестами.

Трансформувати

2-3 квартали

Наскрізний Stage-Up в усіх дев'яти вимірах, доведений порівнянням із базлайном.

// ЩО ВИ ОТРИМУЄТЕ

- ✓ Чіткий базлайн і дорожню карту
- ✓ Робоче AI-середовище, яким команди користуються щодня
- ✓ Вимірюване покращення на ваших власних метриках

Почніть з Diagnostic & Baseline фіксованого обсягу ⟶

Для інженерних лідерів та їхніх команд доставки

// 02 — ЗСУВ

Від автодоповнення до автономності

За кілька років ШІ в редакторі пройшов шлях від вгадування наступного токена до самостійної роботи над багатьма файлами. Ваші розробники вже рухаються цією кривою — вибір лише в тому, чи рухається організація свідомо, чи за інерцією.

Автодоповнення
наступний токен

Підказки
цілі функції

Чат
опиши → код

Агенти
багато файлів, запускає тести

Автономність
планує та відвантажує PR-и

> **Саме тут приходить dev-ai.dev: ми свідомо піднімаємо ваші команди вище цією кривою — зі специфікаціями, верифікацією та запобіжниками, які роблять кожен крок безпечним.**

СИНТАКСИС → НАМІР

Обмеженням більше не є швидкість набору — тепер це якість специфікації на вході та довіра до верифікації на виході. Ваші люди переходять від написання коду до постановки, рев'ю та перевірки.

ШІ ПІДСИЛЮЄ ВАШУ КУЛЬТУРУ

`4× виходу` `~12% цінності` `1,7× проблем`

Прикрутіть агентів до незмінного процесу — і обсяг злетить разом із churn, дефектами та злитими без рев'ю PR-ами. Ця технологія множить ту культуру, в яку потрапляє.

// 02 — ЕКОНОМІКА

Чому ad-hoc “вайб-кодинг” виходить дорого

Виглядає безплатно — підписка і промпт. Справжня ціна захована в тому, що далі: спалені токени на спроби й помилки, переробка неструктурованого коду та пізно знайдені діри в безпеці. Агентна інженерія розвертає криву — більше вкладень на старті, значно дешевша експлуатація.

“Вайб-кодинг”

МАЛО НА СТАРТІ · ДОРОГО ПОСТІЙНО



Дешево почати — а платиш вічно: токени на повторні промпти, дебаг неструктурованого AI-коду й дорогі пізні виправлення безпеки.

Агентна інженерія

БІЛЬШЕ НА СТАРТІ · ДЕШЕВО ПОСТІЙНО



Вкладіться раз у специфікації, тести, контекст і запобіжники — і гранична вартість кожної наступної фічі падає, а маршрутизація моделей зрізає витрати на токени.

> **dev-ai.dev виводить ваші команди на криву агентної інженерії — стартову інвестицію ми робимо разом з вами, тож витрати нижче проєктуються геть, а не сплачуються вічно.**

// ПРИХОВАНА ВАРТІСТЬ

Спалені токени — низький first-pass success означає цикли повторних промптів, які непомітно накручують рахунок.

// ПРИХОВАНА ВАРТІСТЬ

Податок на супровід — неструктурований код коштує днів реверс-інжинірингу, коли ламається за кілька місяців.

// ПРИХОВАНА ВАРТІСТЬ

Пізня безпека — діра, знайдена в продакшені, коштує набагато дорожче за ту саму діру, знайдену на етапі дизайну.

// 03 — ОПЕРАЦІЙНА МОДЕЛЬ

Драбина зрілості

Головний продукт інженера — вже не код, а система, що виробляє код: специфікації, які визначають роботу, агенти, які її виконують, тести й гейти, які її верифікують, і петлі зворотного зв'язку, які повертають збої на виправлення. Кожен агент — це Модель + Харнес: пристойна модель із чудовим харнесом перемагає чудову модель із поганим.

01
Намір і специфікація
Чи записане «чому» до запуску агента — цілі, обмеження, критерії готовності?

02
Контекст, пам'ять і харнес
Чи отримує агент правильний контекст, довготривалу пам'ять і риштування — без деградації в міру заповнення вікна?

03
Цикли, автономія й оркестрація
Ви проєктуєте системи, що промптять агентів, — чи досі промптите вручну хід за ходом?

04
Верифікація та рев'ю
Чи рев'ю ешелоноване за радіусом ураження, змагальне, підкріплене доказами — і чи рішення про merge закріплене за людиною?

05
Тести й детерміновані гейти
Чи несуть справжні тести та непорушні CI-гейти вагу, яку людське читання вже не витягує?

06
Безпека та запобіжники
Чи заблоковані небезпечні дії самою конструкцією системи — пісочниці, хуки, обмежені дозволи, захист від ін'єкцій?

07
Розуміння та судження
Чи команда досі розуміє те, що постачає, — чи мислення тихо віддано моделі?

08
Багаторазовість і «храповик»
Чи кожна помилка стає постійним правилом, скілом або хуком — щоб система накопичувала, а не повторювала?

09
Вимірювання, вартість і компаундинг
Чи можете ви вимірювати, трасувати й бюджетувати фабрику — і довести, що вона стає кращою, а не лише швидшою?

// 03 — П'ЯТЬ СТАДІЙ

Від імпровізації до агентної фабрики ПЗ

Це не узагальнена шкала “ad-hoc → дисципліновано”, а зміна постави агентної ери.
Ми піднімаємо відстаючі несучі виміри на одну сходинку за раз.

S0
Імпровізація
ШІ використовується хаотично і приватно. Копіювання з чат-вікна, без спільного налаштування, без специфікації. Результату довіряють, бо він «виглядає правильно».

S1
Асистування
Автодоповнення та чат — частина робочого дня. Реальні індивідуальні прискорення, але кожен крок керується вручну і тримається в одній голові.

S2
Делегування
Задачі агентного розміру, окреслені письмовою специфікацією; контекст добирається свідомо; кожен результат верифікується. Тести їдуть разом зі зміною.

S3
Оркестрація
Цикли та харнеси, а не разові промпти. Один агент пише, інший перевіряє; робота йде в ізольованих worktree; злиття проходять через гейти з доказами.

S4
Агентна фабрика ПЗ
Агенти виконують тривалі автономні цикли під запобіжниками; фабрика вдосконалює сама себе. Кожен збій стає правилом, скілом або хуком. Evals охороняють промпти так, як тести охороняють код. Важіль зростає без деградації якості.

> **Команда зріла лише настільки, наскільки зрілий її найслабший несучий вимір. Команда на стадії 4 в інструментах, але на стадії 1 у верифікації — це ризик стадії 1: відмовляє саме найслабший гейт.**

// 04 — ФОРМАТ СПІВПРАЦІ

Наскрізна Stage-Up трансформація

Це не серія воркшопів із сертифікатом наприкінці. Ми зобов'язуємося забезпечити вимірюваний перехід на вищу стадію в усіх дев'яти вимірах для команд у скоупі — за чотири фази, і доводимо результат порівнянням із базлайном Фази 0. Кожна фаза має явні критерії виходу, прив'язані до драбини.

ФАЗА 0

Діагностика та базлайн

Встановлює стартову сходинку

Оцінюємо кожну команду та ключових людей; встановлюємо вимірювальний базлайн; аудитуємо репозиторії та SDLC на готовність до ШІ. Доступна як окрема послуга — корисна сама по собі.

Профіль зрілості кожної команди · метричний базлайн · послідовна Stage-Up дорожня карта

ФАЗА 1

Фундамент

→ Стадія 2 · Делегування

Базові політики та безпека, перший спільний харнес, грамотність за ролями — необхідний мінімум, що робить делегування безпечним.

AI Use Policy · кореневий AGENTS.md · стартова бібліотека скілів/MCP · навчені драйвери та рев'ювери

ФАЗА 2

Практика

→ Стадія 3 · Оркестрація

Перебудовуємо реальний робочий процес на живих артефактах; розгортаємо систему верифікації; переосмислюємо фази SDLC довкола наміру.

Плейбук Plan→Implement→Test→Review · ешелоноване рев'ю · CI-гейти · цикли maker/checker

ФАЗА 3

Компаундинг

→ Стадія 4 · Агентна фабрика ПЗ

Проєктуємо цикли, що вдосконалюють самі себе; встановлюємо «храповик»; будуємо eval-набори; запускаємо маховик вимірювань і внутрішню спільноту.

Захищені цикли автоматизації · eval-гейти · регулярний звіт із метрик · Центр досконалості

// 05 — РОБОЧІ ТРЕКИ

Каталог треків розвитку

У наскрізній трансформації треки розподілені за чотирма фазами; їх також можна запускати модульно — під конкретний вимір, що відстає.

Оцінювання впровадження та базлайн

Персональне та командне оцінювання зрілості плюс метричний базлайн, на який спирається програма.

Освітня програма

Воркшопи за ролями (розробник, BA, QA, архітектор, менеджер, керівництво) — теорія плюс практика.

Управління компетенціями

Матриці компетенцій та процеси співбесід/атестацій, зіставлені зі стадіями зрілості.

Впровадження в команди та проєкти

Практичне навчання на живих артефактах: плейбук plan/implement/test/review, багаторазові скіли.

Редизайн процесів SDLC

Переосмислення кожної фази довкола наміру: вимоги→специфікація, рев'ю з урахуванням ШІ, детерміновані гейти.

Харнес і платформна інженерія

Фабрика як спільна інфраструктура: AGENTS.md, бібліотеки скілів/команд, конфігурації MCP, хуки, evals.

Система верифікації та якості

Ешелоноване рев'ю, гетерогенні ШІ-рев'ювери, red/green TDD, мутаційне тестування, evals із рубриками.

Політики, безпека та захист

AI Use Policy, класифікація даних, BYOT, перевірка MCP, пісочниці, захист від ін'єкцій, аудитованість.

Інженерія циклів та оркестрації

Дизайн циклів, розділення maker/checker, масштабування флоту під пропускну здатність рев'ю, захищені нічні цикли.

Економіка / FinOps для ШІ-розробки

Економіка токенів, моделювання CapEx→OpEx / TCO, маршрутизація моделей, бюджети та квоти, доказ ROI.

Лідери, спільнота та кейси успіху

Центр досконалості, мережа чемпіонів, ШІ-ретроспективи, демо-дні, внутрішні змагання.

Продакшн-агенти як продукти

Побудова агентів як продуктів — пам'ять, evals, спостережуваність, деплоймент — а не лише агентів для написання коду.

Супровід і підтримка

Квартальне переоцінювання, підтримка харнесу та evals, звіти з метрик, fractional-лід ШІ-інженерії.

// 06 — ВИМІРЮВАННЯ

Кожне твердження вимірюється, а не декларується

П'ять інструментованих категорій — впровадження, пропускна здатність, якість і ризики, економіка, компаундинг — з регулярним звітуванням, щоб швидкість ніколи не ховала якість. Метрики DORA розширено виміром ШІ-впровадження; модель ROI цілиться в сукупну вартість володіння — число, яке впізнає CFO.

2,5×
зниження вартості розробки на одному з продуктів

93%
ШІ-згенерованого коду в масштабі команди, підкріплено ~1 400 інтеграційними тестами

7 міс
для 48 інженерів у 5 командах — від хаосу до системної практики

// РЕПРЕЗЕНТАТИВНІ РЕЗУЛЬТАТИ ПОРІВНЯННОЇ КОРПОРАТИВНОЇ ПРОГРАМИ

МЕТРИКА	ДО	ПІСЛЯ
Участь у корпоративному ШІ-навчанні	10,6%	**66,7%**
Глибока обізнаність та активне використання ШІ-інструментів	42,6%	**56,7%**
Системний підхід до промпт- і контекст-інженерії	53%	**70%**
Щоденне використання кількох ШІ-інструментів	59,6%	**70%**

Додатково: частка інженерів, що заощаджують 9+ годин на тиждень, зросла на 22 пункти, а найбільші прирости активності — у написанні тестів та автоматизації, документації і код-рев'ю (близько +40 пунктів за кожним напрямом).

// 07 — ҐАВЕРНАНС, БЕЗПЕКА Й ЗАХИСТ

Впровадження тримається на довірі

Ґавернанс-фундамент, який лишає ШІ придатним для щоденної роботи й водночас захищає конфіденційність клієнтів, інтелектуальну власність, персональні дані та безпеку систем — за зразком політики, вже розгорнутої в enterprise-організації доставки.

- **Політика інструментів під вас.** Допомагаємо визначити, які AI-інструменти схвалені — і за яких умов — з ваших вимог безпеки, клієнтів і комплаєнсу: bring-your-own-tool там, де доречно, жорсткі обмеження там, де ні, і вимкнене навчання провайдерів на ваших даних.
- **Класифікація даних як основний механізм.** Конкретні категорії того, що може і що не може потрапляти в інструмент; припущення, що все введене може витекти, якщо не авторизовано інше; мінімізація даних — підсумовуй, редагуй, ізолюй.
- **Класифікація на рівні кожного проєкту.** Кожен проєкт позначений: AI дозволено, AI дозволено з обмеженнями або AI заборонено. Суворіші правила клієнта завжди виграють.
- **Технічні запобіжники, а не лише політика.** Секрети не потрапляють у контекст; обмежені права інструментів і MCP; перевірені за походженням, зафіксовані за версіями MCP; пісочниці й захист від ін'єкцій; системні блокування деструктивних дій.
- **Відповідальність лишається за людьми.** AI-код проходить те саме рев'ю, тестування й сканування безпеки, що й будь-який код, з посиленим контролем на шляхах auth, crypto, IaC, доступу до даних і комплаєнсу; авторство атрибутується, використання аудитується.

// 08 — ДОКАЗИ

Вибрані кейси

Анонімізовано з реальних проєктів; назви клієнтів і продуктів замінено кодовими іменами.

Project Atlas

Скептично налаштована delivery-команда, тиск через вартість ШІ-асистованої роботи.

Зниження вартості розробки у 2,5 раза; валідовані стандарти якості коду перетворили скепсис на прийняття.

Project Beacon

Велика регульована платформа у США, 60 000+ проіндексованих файлів, суміш легасі та сучасних фронтендів.

Фулстек-баги виправляються менш ніж за хвилину; кастомний звіт, що займав місяць, тепер готовий за години; ~$200/міс на весь ШІ-тулінг команди.

Project Orbit

Greenfield-продукт (~1 рік), архітектурно спроєктований дружнім до ШІ.

~93% коду згенеровано ШІ у масштабі команди (зростання з ~75%); ~1 400 інтеграційних тестів; дисциплінований процес plan/implement/test/review, продемонстрований наживо всій організації.

Harbor & Summit

Навчання бізнес-аналітиків, не лише розробників.

BA тепер генерують вимоги, що спираються на реальну кодову базу, — специфікації з урахуванням імплементації, які роблять подальшу роботу ШІ ефективною.

Project Cobalt

Незалежна сольна greenfield-доставка для окремого клієнта.

Повний MVP (36 вимог, ~395 файлів) побудовано за ~28–32 інженерні години у spec-driven, повністю ШІ-асистованому процесі з QA-пакетом доказів.

Project Quartz

Когорта із 70 QA та BA у продуктовій компанії, оцінювання через merge request-и.

100% / 100% / 98% виконання за трьома рівнями складності до шостого воркшопу — із відстеженням кожної здачі в журналі, який можна проаудитувати.

Project Keystone

Поставка агентних артефактів для команди з монорепозиторієм .NET + Angular.

146 файлів і ~19 500 рядків закомічено в репозиторій клієнта: 41 правило, 5 скілів, захисні хуки та pre-commit ШІ-гейт, покритий 84 тестами.

> **Переносний урок: коли обсяг ШІ-згенерованого коду масштабується, обмеження зсувається від написання коду до його рев'ю. Вигідно рано інвестувати в ШІ-асистоване рев'ю, сильніші автоматичні гейти та якісні специфікації — саме це й будують ці співпраці.**

// 09 — ПРО НАС І ЯК СТАРТУЄМО

Побудовано й ведеться практиком

В'ячеслав Колдовський, Ph.D.

Лід трансформації ШІ-інженерії · викладач, спікер конференцій і дослідник агентної інженерії

Автор і єдиний драйвер корпоративної програми впровадження агентної інженерії: 17 практичних воркшопів по всьому SDLC, фреймворки компетенцій зі 100+ пунктами оцінювання та ~300 питаннями, AI Use Policy, чеклісти зрілості із супровідними гайдами та практичне навчання п'яти delivery-команд.

Підхід ґрунтується на порівнянній корпоративній програмі, яка за сім місяців перевела 48 інженерів у п'яти командах від хаотичних експериментів до системної практики: зростання активного використання ШІ-інструментів, зниження вартості розробки у 2,5 раза на одному продукті та 93% ШІ-згенерованого коду на іншому — усе в межах явної політики безпеки.

// ДОСВІД

Корпоративні тренінги для: SoftServe, InventorSoft, Intelliarts, Langate, MR Lab, KeenEthics

Виступи на конференціях: DOU Day 2025 і 2026 · fwdays AI Summit 2026

Академічна діяльність: IT Step University — доцент, керівник Центру досліджень агентного ШІ · American University — автор курсу з агентної інженерії для магістрів

// ЯК МИ СТАРТУЄМО

1. Фаза 0 — Діагностика та базлайн (фіксований обсяг, ~2–3 тижні)
2. Узгоджуємо цільову стадію, скоуп і запобіжники політик
3. Запускаємо пофазну трансформацію з регулярним звітом із метрик

Нам потрібні: спонсор в інженерному керівництві, доступ до репрезентативних репозиторіїв і команд та погодження запобіжників використання інструментів.

Старт — з першого дня.

dev-ai.dev

Dev AI Consulting — трансформація агентної інженерії: оцінювання, навчання та програмна фабрика, побудована наскрізно.

Живий сайт, завжди актуальний: dev-ai.dev/uk

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

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

// джерела — зовнішні цифри 2026 року з публічних досліджень агентної інженерії та AI-SDLC (Osmani; whitepaper "The New SDLC"; GitClear, Faros AI, CodeRabbit, METR). Результати програми — з анонімізованої enterprise-ініціативи. Детальні джерела — на запит.