РОЗДІЛ 1

Три ролі поза технікою

Більшість AI-проектів зазнають невдачі через одне питання, яке ніхто не ставить: хто вирішує і коли.

Невдача — не в моделі. Модель здебільшого працює. Невдача в тому, що після запуску система потрапляє в прогалину, де вже ніхто не несе відповідальності. Аналітик даних пішов до наступного конвеєра. Інженер з експлуатації моделей дивиться на дашборди моніторингу, пороги яких він сам і виставив. Керівник заводу знає, що десь працює модель — і це нормально, він не зобов'язаний знати ні тренувальних даних, ні постачальника GPU. Проблема в іншому: коли модель видає хибну рекомендацію в крайовому випадку — а це трапляється щомісяця в якійсь зміні — організація шукає того, хто має мандат рішення. І не знаходить нікого.

Те, що виглядає як програмна проблема, насправді є антропологічною. Три ролі не були вимовлені до того, як систему запустили в експлуатацію. Три зони відповідальності не закріплені, три ескалаційні шляхи не побудовані. Це і є ролі цього розділу: хто тренує, хто контролює, хто відповідає.

Що антропологія означає в технічній системі

Антропологія звучить академічно — але тут йдеться про прикладне значення: про підхід і розуміння того, як AI-системи функціонують у щоденній експлуатації. Конкретно — які люди з якими мандатами на яких ділянках AI-системи є невід'ємними. Не питання, хто вміє побудувати модель (це лише частина системи і відповідальність вужчої ролі), а питання їхньої позиції в архітектурі, яка тримає модель робочою тривалий час.

AI-система складається з двох шарів. Технічний — це код, дані, програмна розробка моделі та інфраструктура. Соціальний — це люди з чіткими ролями, без яких технічний шар через короткий час починає дрейфувати, відмовляти або генерувати ризик, якого ніхто більше не контролює.

Норберт Вінер у праці *The Human Use of Human Beings* вже формулює це розрізнення: автоматична система не існує без людського контуру керування, у якому її показники інтерпретуються і її поведінка коригується. Стюарт Расселл у праці *Human Compatible* перекладає цю саму інтуїцію в сучасну рамку: без явного носія мандату контролю автономна система за визначенням оперує поза петлею керування.

Хто будує AI-систему без побудови соціального шару, той фактично купує дороге обчислення (GPU, моделі, інфраструктуру), не побудувавши шляхів ескалації — і тому при першому інциденті немає кому втрутитися. Це не питання технічного налаштування моделі; це питання організаційне.

Три ролі, яких немає в жодному технічному описі моделі

Кожен робочий AI-процес (від тренування і розгортання моделі до її щоденної експлуатації) несе три ролі. Вони відрізнювані. Вони не можуть бути зведені в одну особу. І вони не з'являються автоматично в жодній організаційній структурі.

Хто тренує. Ця особа визначає, що модель вважатиме реальністю. Вона погоджує тренувальні дані, приймає або відсіює аномальні записи (поодинокі спостереження, що сильно відхиляються від решти і можуть викривити модель), формулює схему розмітки і вирішує, які припущення кодуються явно. Хто тренує, той карбує реальність, у якій сис-

тема оперує. Термін *куратор даних* пасує цій ролі краще, ніж *аналітик даних*, бо фіксує суть роботи: не аналіз даних задля висновків, а добір, відсіювання і документування того, що увійде в навчальний набір — а отже стане «реальністю» для моделі.

У більшості індустріальних проектів ця позиція невидима. Тренувальні дані витягують зі звіту в ERP-системі (промислова платформа управління ресурсами підприємства) без перевірки, чи не спотворює день страйку середньостатистичну поведінку. Набір даних з 2022 року заходить у модель, хоча лінію перебудували в 2025-му. Ніхто не заперечує, бо ні в кого немає на це мандату.

Хто контролює. Ця особа має мандат втручатися в робочий процес. Вона визначає пороги, при яких модель ставиться під сумнів, і ескалаційні шляхи, якими користуються в надзвичайних випадках. Вона має право скасовувати рішення моделі (формальне право скасувати або змінити рекомендацію та ухвалити власне рішення з вищим пріоритетом) і протоколює їхнє застосування.

Чого вона не робить: сліпо довіряє моделі. Чого вона так само не робить: ігнорує модель і ухвалює рішення «на око», бо так звичніше і швидше. Вона є *30 %-шаром* — людською частиною підходу 70/30, яка дбає, щоб крайові випадки не ставали ескалаціями, а ескалації не ставали зупинками.

Хто відповідає. Ця особа несе юридичну відповідальність. У сенсі EU AI Act це означає: вона підписує декларацію про відповідність, веде технічне досьє, повідомляє про серйозні інциденти. Ця роль повинна бути закріплена письмово, з покриттям страхування і чітким мандатом. Вона не є автоматично частиною обов'язків технічного директора (CTO), керівника заводу чи комплаєнс-менеджера. Це окреме призначення.

Хто відповідає, мусить розуміти, що робить модель, не побудувавши її особисто. Ця позиція вимагає перекладацької роботи між технікою і організацією, яка рідко зустрічається в класичній організаційній структурі IT-департаменту.

Що відбувається, коли всі три ролі покладаються на одну особу

Найчастіший випадок в DACH-індустріальних проектах: один керівник заводу неформально несе всі три ролі. Він підписав проект, знає тренувальні дані, авторизує втручання і підписує звіти для аудитора. Поки керівник заводу на посаді, система працює. Як тільки він змінюється, проект ламається — не технічно, а антропологічно.

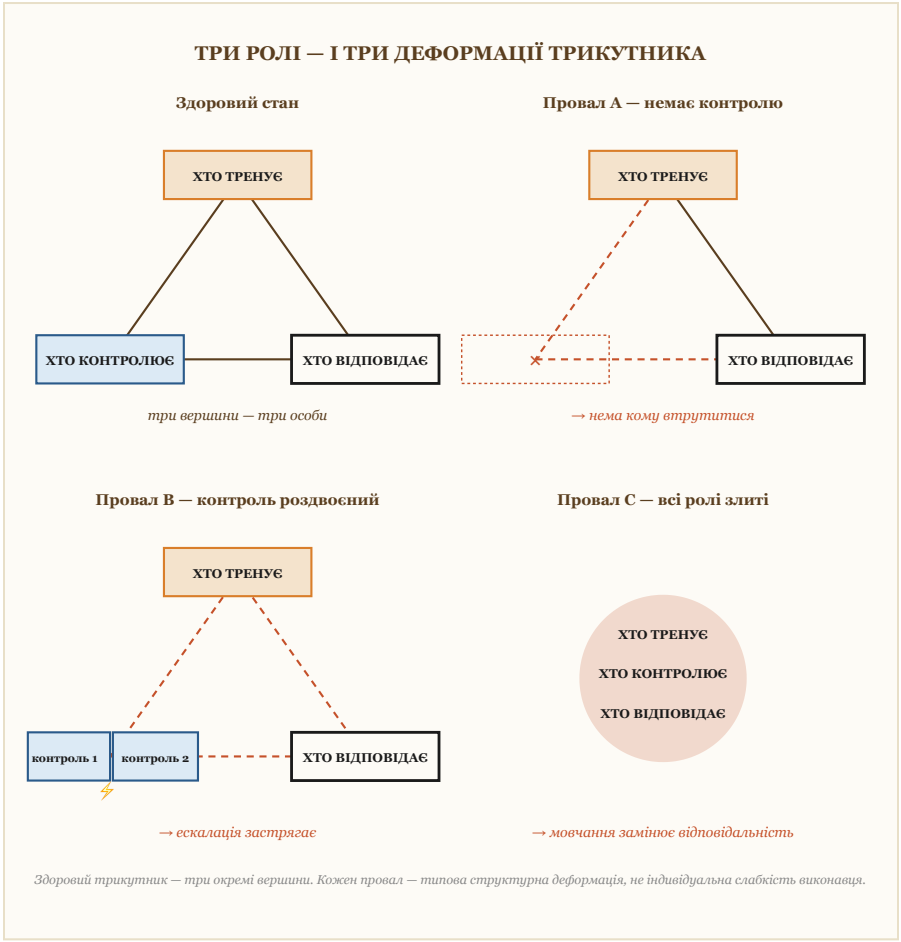
Одна особа не може водночас курувати дані, ухвалювати рішення за мандатом і нести юридичну відповідальність. Ролі стикаються при кожному серйозному запитанні. Жоден керівник заводу, який виявив помилку моделі, не прагнутиме притягнути сам себе до відповідальності — натомість здебільшого інцидент буде зам'ятий, перекваліфікований у «технічний збій» або відкладений на «розберемося пізніше». Замість ескалації виникає мовчання, і мовчання — найдорожчий варіант.

Чому ролі мусять бути виписані — аналогія

Хто знайомий з мовою програмування Rust, той знає принцип: компілятор (програма, що переводить написаний людиною код у машинні інструкції; перед цим перевіряє його коректність) вимагає, щоб кожна змінна мала вимовлену роль. Хто володіє пам'яттю? Хто позичає її тимчасово? Яке посилання може модифікувати, яке тільки читати? Без явних анотацій програма не компілюється.

Спочатку це виглядало педантично. З часом виявилося передумовою безпеки пам'яті в продуктивних системах. Те, що раніше в C++ проходило під виглядом *досвідчений програміст це й так зробить правильно*, тепер є виписаною контрактною структурою між кодом і компілятором.

Забудьте про промпти. AI-система 70/30 слідує тій самій логіці. Хто тренує, хто контролює, хто відповідає — це анотації, без яких система ламається в промисловій експлуатації. *Воно компілюється*, бо модель тренується і процес працює. Через вісімнадцять місяців приходить перша ескалація, і ні в кого немає мандату. Це AI-еквівалент звернення до вже звільненої пам'яті: формально код виконується, але працює з даними, які вже не належать нікому, і призводить до непередбачуваного збою.



Три ролі — три особи з виписаним мандатом. Провал A: відсутня вершина — немає шляху обходу рішення моделі. Провал B: розщеплена вершина — ескалація застрягає в конфлікті між двома контролерами. Провал C: трикутник згорнутий у точку — мовчання замінює відповідальність.

• • •

Три питання для власного контексту

Кожен розділ цієї книги закінчується трьома питаннями для самоперевірки — короткий оперативний орієнтир, який можна пройти за десять хвилин і записати відповіді з іменами і датою.

1. **Хто тренує?** Яка поіменно вказана особа має мандат приймати, відхиляти або курувати тренувальні дані? Коли роль востаннє підтверджена письмово?
2. **Хто контролює?** Хто має право скасовувати рішення в робочому режимі? Чи задокументовані пороги і ескалаційні шляхи, і хто протоколює їхнє застосування?
3. **Хто відповідає?** Хто підписує декларацію про відповідність за EU AI Act? Хто несе страхування? Хто повідомляє про інциденти органу нагляду ринку?

Якщо на одне з цих питань немає імені, або тричі стоїть одне й те саме ім'я, система антропологічно недобудована. У наступних розділах кожна роль розглядається окремо.