РОЗДІЛ 7

70/30-Підхід — як практично будувати два шари

> *Чому 70% автоматизації і 30% людської перевірки дають економію витрат, якої не дає ні повна автоматизація, ні людський процес із доданою зверху AI-надбудовою. Як свідомо спроєктувати обидва шари, машинний і соціальний, так, щоб кожен працював у власному темпі і обидва замикались на спільну петлю відповідальності.*

Bain & Company у дослідженні 2025 року показало: компанії, що поєднують генеративний AI з реальною перебудовою процесів, скорочують витрати до 25%. Ті, що додають AI поверх існуючих процесів без перебудови, отримують одиничні відсотки або взагалі нічого. Цифра 25% в економії витрат є найсильнішим зовнішнім доказом на користь 70/30-підходу: технологія сама по собі економіки не дає, її дає конструкція людських і технічних шарів разом.

Шарова модель: технічний і соціальний шари

Більшість архітектурних діаграм AI-систем показують технічний шар: модель, конвеєр обробки даних, інфраструктуру, інтеграції. Іноді ще інтерфейс кінцевого користувача, тобто UI (user interface).

70/30-архітектура потребує діаграми, на якій технічний і соціальний шари показані разом і у взаємозв'язку: модель, конвеєр даних, інфраструктура — з одного боку; ролі, ескалаційні шляхи, мандати, пороги, що фіксується в журналі і хто це перевіряє — з іншого. Обидва шари є повноцінними, рівноправними компонентами системи. Тільки разом

вони утворюють AI-систему: окремий технічний шар залишається автоматизованим процесом без контролю, а окремий соціальний — колекцією людських рішень без автоматизованої опори. Ще Норберт Вінер у *The Human Use of Human Beings* (1950) описав автоматизовану систему як таку, що набуває сенсу лише у комбінації з людським шаром нагляду; без нього кібернетична петля замикається на саму себе і втрачає коригувальний контур.

Технічний шар. До технічного шару належать наступні компоненти. Модель або кілька моделей, що працюють разом і об'єднують свої результати в одне рішення (такий набір моделей називають ансамблем). Конвеєр попередньої обробки даних, до якого належать чистка, нормалізація і побудова ознак. Інфраструктура виведення. Логіка оцінювання й порогів, тобто перетворення результатів моделі на рішення або на позначки для перевірки людиною. Журналювання, тобто що саме система фіксує автоматично. Моніторинг, наприклад метрики продуктивності та виявлення зміщення розподілу даних у часі.

Соціальний шар. Визначення ролей: тренування, контроль, відповідальність, з конкретними іменами. Ескалаційні шляхи, тобто хто буде сповіщений у якому порядку за яких умов. Процедури ручного перехоплення, тобто як люди скасовують системні рішення і що при цьому фіксується в журналі. Мандати, до яких відноситься, хто які рішення може приймати, з чіткими межами повноважень. Процеси нагляду, наприклад як часто перевіряються пороги ескалації (числові межі, при яких рішення передається від машини до людини) і хто бере участь у перевірці. Документаційні обов'язки, тобто що, до коли і ким фіксується.

Crew Resource Management як інституційний прецедент. Авіація прийшла до цього висновку у 1970-х. До 1979 року звіти про авіаційні аварії вказували на стійку схему: технічно справні літаки розбивались через помилки координації в кабіні — старший пілот, що не приймав заперечень молодшого, ієрархічна упередженість, відмова де-

легувати. NASA-психолог Джон Лаубер ввів термін Crew Resource Management (CRM, управління ресурсами екіпажу) у 1979 році. Протягом наступного десятиліття FAA випустила Advisory Circular 120-51 — обов'язкове тренування CRM для всіх перевізників США. Поточна редакція AC 120-51E фіксує п'ять поколінь CRM з розширенням на багаточленні екіпажі та середовище одного пілота.[1] Що з цього ми можемо взяти для AI: авіація формально визнала, що навіть найдосконаліший літак — лише половина системи. Друга половина — натренована соціальна архітектура кабіни: ролі, ескалаційні шляхи, мандат молодшого пілота сказати *зупиніть це*. Це і є соціальний шар як рівноправний компонент. AI-системи у 2026 знаходяться приблизно на тому етапі, де авіація була у 1978 — технічна досконалість росте швидко, соціальний шар недорозвинений.

Восьмикроковий процес: від мапи процесу до ескалаційного дрилу

Нижче описана практична послідовність побудови 70/30-системи. Кожен крок випливає з категорії помилок, яка повторюється у різних компаніях саме тоді, коли цей крок пропускають. Скорочення восьмикрокового переліку до трьох-чотирьох пунктів виглядає привабливо на початку, проте різко знижує ймовірність того, що система працюватиме після введення в експлуатацію. Кожен з восьми пунктів закриває конкретний клас провалу, без якого жодна архітектура 70/30 не масштабується далі за пілотний проєкт — обмежене випробування на одній ділянці перед розгортанням на всю організацію.

Крок 1: Мапа процесу. Перший крок — опис поточного процесу, поки що без моделі і без AI взагалі. Команда фіксує процес таким, яким він виглядає сьогодні з людьми: кожен крок, кожне рішення, кожен перехід між учасниками. Це тривалий процес, який часто здається надмірним і породжує спокусу скоротити саме цю фазу опису поточного стану — попри це, без такого опису команда не знає, що саме зби-

рається автоматизувати, і ризикує накласти AI на викривлене уявлення про власну роботу. Результатом кроку є мапа процесу — графічне або табличне представлення з усіма точками рішень, на які спиратимуться наступні кроки.

Крок 2: Ідентифікація точок прийняття рішень. На мапі процесу, побудованій у попередньому кроці, виокремлюється кожна точка, в якій приймається рішення. Не всі ці точки підходять для автоматизації — частина з них залишається у людини незалежно від технічних можливостей, через регуляторні, етичні або контекстні причини. Для кожної точки команда відповідає на чотири питання: яка ціна помилки в цьому місці, яке регуляторне навантаження пов'язане з цим рішенням, як часто воно приймається у нормальному ритмі роботи, наскільки варіативними є вхідні дані. Відповіді на ці чотири питання визначають, чи варто взагалі розглядати конкретну точку для автоматизації, і саме вони стають критерієм відбору для наступного кроку.

Крок 3: Визначення 70%. З-поміж точок, ідентифікованих у попередньому кроці, виокремлюється підмножина тих, які система має виконувати автоматично. Критерії відбору такі: висока частота повторюваності, низька варіативність вхідних даних, прийнятна ціна помилки і відсутність регуляторної вимоги людського підпису. Сукупність точок, що задовольняють усім чотирьом критеріям одночасно, і відповідає принципу 70% автоматизації. У різних компаніях цей показник може варіюватися — у дисциплінованих процесах він іноді сягає 80%, у складніших регуляторних середовищах може бути помітно нижчим. Якщо після відбору показник можливої автоматизації виходить значно меншим за 70%, це сигнал не до зменшення амбіцій, а до перегляду самої архітектури процесів: процеси треба перебудовувати так, щоб вони витримували критерії, описані в цій секції, інакше економіка 70/30-підходу залишається недосяжною.

Крок 4: Визначення 30%. Точки, які залишаються у людини, утворюють повноцінну другу частину системи і потребують такої самої уваги при проєктуванні, як і автоматизована частина. Для кожної людської точки прийняття рішень фіксуються три параметри: хто конкретно приймає це рішення, за яких умов воно потрапляє до цієї людини і в якому часовому вікні воно має бути прийняте. Принциповою є фіксація названих осіб, а не лише функціональних ролей. Запис на кшталт *відповідальний керівник напряму* або *черговий експерт* — недостатній: за цією формулою ніхто персонально не вважає себе відповідальним у момент інциденту. Достатній запис вказує ім'я конкретної людини, її посаду та визначеного замісника на випадок відсутності, відпустки або зміни режиму роботи, щоб у момент ескалації не залишалось жодної невизначеності щодо того, до кого саме потрапляє рішення.

Крок 5: Встановлення порогів. Для кожної автоматизованої точки команда визначає пороги, на яких справа переходить до людини. Приклади формулювань: *якщо впевненість моделі нижче 85%, рішення ескалюється до AI-оператора*, або *якщо рішення впливає на партію понад 500 одиниць, потрібне людське підтвердження*. Пороги мають бути специфічними, вимірюваними і зафіксованими в експлуатаційній документації, а не лише прихованими у коді — інакше вони залишаються невидимими для тих, хто за ними працює щодня, і ревізії порогів стають неможливими без занурення у код.

Крок 6: Дизайн журналювання. Журналювання охоплює кожне системне рішення, кожну ескалацію і кожне ручне перехоплення — із зазначенням імені, мітки часу та причини дії. Окремо фіксується, де саме зберігаються журнали, хто має до них доступ і протягом якого терміну вони зберігаються. EU AI Act для систем високого ризику встановлює мінімальний термін зберігання у п'ять років, однак внутрішня політика компанії може передбачати довші терміни, якщо цього вимагають галузеві стандарти або характер ризиків.

Крок 7: Експлуатаційна документація. До введення в експлуатацію готуються три підписані документи. *Системний опис* фіксує, для кого створено систему, що вона робить, які пороги ескалації встановлено і хто відповідає за тренування, експлуатаційний контроль та підсумкове прийняття рішень. *Експлуатаційна процедура* описує, що робить оператор у нормальному режимі, при ескалації та при відмові. *План відновлення після відмови* визначає, що відбувається при повній недоступності системи і як процес продовжується вручну до моменту повернення системи у штатний режим.

Крок 8: Ескалаційний дрил. До введення в експлуатацію команда симулює інцидент — обирає ескалаційний сценарій і проходить його у реальному часі: хто отримує сповіщення, хто реагує, що ця людина робить, що фіксується в журналі. Час проходження сценарію замірюється відразу, ще на етапі дрилу. Допустима тривалість ескалаційного шляху залежить від типу рішення: для критичних інцидентів вона зазвичай вимірюється кількома хвилинами, для рутинних — десятками хвилин, і конкретна межа має бути зафіксована в експлуатаційній процедурі. У реальному інциденті ця тривалість ще зросте через стрес, нічний час або відсутність ключової людини, тож дрил у спокійних умовах слугує лише нижньою межею оцінки.

NASA «Lock the Doors» як прецедент дрилу. У NASA Mission Control існує задокументована процедура «Lock the Doors» — Flight Director оголошує її у момент критичного інциденту. Двері до контрольного приміщення фізично замикаються, доступ ззовні припиняється, кожна станція переходить у режим, який в галузі описують як FREEZE-ISOLATE-PROTECT — заморозити, ізолювати, захистити: зафіксувати поточний стан системи, ізолювати від нових змін, захистити документацію і журнали. Процедура не для самого інциденту — вона для того, що буде після. Вона гарантує, що коли почнеться розслідування, дані будуть у первісному вигляді, а не перезаписаними під час хаотичної реакції. Для AI-операцій еквівалент очевидний: коли ескалація приходить, перший крок не *виправити* — перший крок зафіксувати, який

саме стан системи призвів до ескалації, з ким, у який час, на яких вхідних даних. Без цього розслідування інциденту через тиждень стає спекуляцією, не аналізом. Дрил, який не включає фазу заморожування, тренує реакцію, але не вчить діагностиці.[2]

Стандарт документації: що має існувати письмово до того, як модель іде в продакшн

Це мінімальний обов'язковий перелік без гарантії повноти. Для систем високого ризику за EU AI Act — суттєво більше.

Системна карта. Назва і версія системи. Компоненти, до яких належать модель, версія, інфраструктура, інтеграції. Класифікація за EU AI Act. Дата останнього оновлення і підпис.

Ролі і контакти. Тренування: ім'я, посада, контакт. Контроль (операційний): ім'я, посада, контакт, контакт замісника. Відповідальність: ім'я, посада, контакт. Дата затвердження, підпис кожного причетного.

Операційні пороги. Кожен автоматичний поріг ескалації (числовий). Кожен контекстний тригер ескалації. Дата останньої перевірки, особа, що затвердила.

Ескалаційний шлях. Сценарій → хто сповіщений → час відповіді → що ця людина робить. На сценарій: стандартна ескалація, нічна/вихідна ескалація, критична ескалація.

Процедура ручного перехоплення. Як технічно виконується перехоплення? Які поля мають бути в журналі — формат запису? Хто перевіряє журнали перехоплень і коли?

План відновлення після відмови. Що відбувається при повній відмові системи? Ручна заміщувальна процедура. Критерій повернення до системи після відновлення.

Типові помилки

Помилка 1: Концентрація мандатів. Досвідчений технік отримує всі ролі — тренування, контроль, відповідальність плюс *загальна AI-відповідальність*. На папері зрозуміло. У експлуатаційній реальності — людина перевантажена, не може бути всюди одночасно, і в момент інциденту вона єдина точка відмови. Розподіл ролей має відображати реальну доступність, а не організаційну зручність.

Помилка 2: Технічні пороги без людського контексту. *Ескалація при впевненості моделі нижче 85%* звучить точно. Але якщо оператор, до якого приходить ескалація, не знає, що означає *впевненість 84% для цього типу рішення* практично, він не може прийняти осмисленого рішення. Технічні пороги потребують людського пояснення: *Це означає, що система не впевнена у своєму рішенні — наприклад через незнайомий формат документа або нового постачальника. У такому випадку оператор сам перевіряє тип документа і ключові поля, після чого підтверджує рішення або повертає документ на повторну обробку.*

Помилка 3: Документація, яку ніхто не читає. Чудова документація у спільній теці на корпоративному порталі, яку з моменту введення в експлуатацію ніхто не відкривав, — це не документація. Експлуатаційна документація має бути доступна там, де її потребують люди: на робочому місці оператора, у системному інтерфейсі, у першому вікні після надходження ескалації.

Помилка 4: Нагляд без дати. *Пороги перевіряються за потреби* означає, що вони не перевіряються. Нагляд належить у календар: раз на квартал — або після кожного значимого інциденту — конкретна зустріч з конкретними людьми, що перевіряє конкретні метрики і вирішує, чи треба адаптувати пороги або розподіл ролей.

Помилка 5: Немає плану переходу при зміні персоналу. AI-оператор бере відпустку. Або звільняється. Хто переймає його функцію? Чи введена ця людина в курс справи? Чи знає, де лежить експлуатаційна документація? Передача знань не повинна бути ситуативною — вона має бути запланованою компонентою системи.

> *Bain & Company (2025): компанії, що комбінують генеративний AI з перебудовою процесів, скорочують витрати на 25%. Економіка з'являється лише разом із перебудовою, і 70/30-підхід саме є тим способом такої перебудови.*

Опорна діаграма Р7/1. *Дві закриті петлі: машинна (норма, секунди) й людська (виняток, хвилини). Кожна стабілізує себе у власному темпі.*

Три питання для власного контексту

1. **Шарова діаграма.** Намалюйте свою AI-систему у двох шарах — технічному і соціальному. Якщо соціальний шар намалювати неможливо, конструктивно його не існує.

2. **Перевірка порогів.** Коли востаннє перевірялись операційні пороги вашої системи? Хто був присутній? Що було змінено? Якщо відповідь *не пам'ятаю* або *ніколи* — дійте.

3. **Ескалаційний дрил зараз.** Виберіть сценарій, замірте час. Хто отримав сповіщення? Хто відреагував? Що тривало найдовше? Результат дрилу — порядок денний наступної зустрічі з нагляду над системою.

Посилання

1. John K. Lauber, «Resource Management on the Flightdeck», NASA Conference Publication 2120, червень 1979; FAA Advisory Circular 120-51E «Crew Resource Management Training», 22 січня 2004 (поточна редакція).

2. NASA Johnson Space Center, Flight Director Operations Handbook (FDOH); опис процедури «Lock the Doors» у Gene Kranz, *Failure Is Not an Option*, Simon & Schuster, 2000.