AI-ДАЙДЖЕСТ

DeepSeek виклала власний агентний харнес під MIT – і за добу він зібрав 68 тисяч зірок. Це прямий відкритий аналог того, на чому я працюю

Веду першим, бо це найближче до нашої конструкції за весь тиждень, і вперше – не «схожа ідея», а робочий код у відкритому доступі, який робить рівно те, що робить мій харнес.

п'ятниця, 14 серпня 2026

У ЦЬОМУ ВИПУСКУ

ТЕМА 1

DeepSeek виклала власний агентний харнес під MIT – і за добу він зібрав 68 тисяч зірок

Перший пункт, бо це вперше робочий код у відкритому доступі, який робить те саме, що роблять закриті агентні харнеси.

DeepSeek Harness (`dsh`), developer preview, MIT, TypeScript. Репозиторій створений **13 серпня об 11:56** - на момент збору в нього **67 988 зірок**. Тобто ~68 тисяч менш ніж за добу, один з найшвидших стартів за останні роки. На HN - 592 бали.

Гасло - «Everything is a plugin», і воно буквальне. З їхньої сторінки: «Кожна можливість - це плагін, який можна замінити або перескласти: **моделі, тули, скіли, сесії, пісочниці, сховище, цикли, планування і UI**». Побудовано на ядрі **Cordis**, яке керує монтуванням і залежностями плагінів.

Друга половина дизайну цікавіша: «Кожен прогін відстежуваний». Дослівно: «Усе, що бачить модель, записується в **append-only лог сесії**: системні промпти, міркування, виклики тулів і результати, планування субагентів і кожна ін'єкція контексту». Зверху - Trajectory view, де ці записи можна дивитись за джерелом, і «resume, fork, search і replay працюють на тому самому потоці подій».

Плюс чотири режими виконання. **Standard** - повний набір. **Code mode** - модель пише код, який оркеструє багато викликів тулів одразу. **Minimal** - лише shell і редактор файлів, для чистого бенчмаркінгу моделей. **Creator mode** - інспектувати рантайм і збирати з плагінів нові режими.

Скепсис із треду, і він по ділу. Топовий коментар холодний: «якщо воно не краще за omp - пробувати немає сенсу». Другий питає закономірне: «Чому так багато цих агентних харнесів написані на node.js?». І третій: «схоже, це рух від md-файлів до cordis-плагінів?».

Чому це важливо. Захват заслужений: append-only лог усього, що бачила модель, з розбором за джерелом - рідкість. Типовий журнал агентної сесії пише **повідомлення**, але не контекст: що саме ін'єктнулось у промпт, який шматок пам'яті підтягнувся, що повернув скрипт. Коли потім треба відновити, чому агент застряг на застиглому DOM чи вигадав ID, робиться це по пам'яті й переказу. Trajectory view - це рівно та видимість замість здогадів.

Дія з цього не випливає майже ніяка. Ставити `dsh` означає нову модель, новий рантайм і developer preview з прямим попередженням у README великими літерами: «Будуть Зміни, ЩО Ламають Сумісність». Для стабільного щоденного інструменту це мінус. Забирати варто ідею логу. Дописати запис того, що реально пішло в промпт сесії, - дрібна правка на пів години в будь-якому конвеєрі. [proven - сторінка продукту і README прочитані, цифра зірок і дата створення з GitHub API, цитати дослівні; **інсталяції й прогону не було**; 68к зірок за добу для китайського релізу - показник уваги, накрутку перевірити неможливо]

DeepSeek Harness - сторінка продукту · Репозиторій на GitHub (MIT) · HN-тред, 592 бали · @eliebakouch: ваги V4 Pro + харнес

ТЕМА 2

Gemini 3.7 Flash - через ТРИ тижні після 3.6, удвічі дешевше. І найкращий коментар доби: «а DeepMind ще фронтир-лаба?»

Топ HN за добу - 687 балів. Google випустила **Gemini 3.7 Flash** 13 серпня. Тут важать темп і ціна.

З анонсу Google: реліз «лише через три тижні після Gemini 3.6 Flash», і це прямо названо результатом фідбеку розробників. Цифри проти 3.6 Flash: **FrontierCode 1.1 Main - 43.6% проти 34.4%**, **DeepSWE v1.1 - 65.3% проти 49.0%**, WebDev Arena Elo **1588 проти 1538**, GDP.pdf **34.0% проти 22.0%**, AutomationBench **30.4% проти 17.0%**. По документах і бізнес-воркфлоу - стрибок майже вдвічі.

Ціна - і ось де дрібний шрифт. Вступна ціна **$0.75 / $3.75 за 1 млн** токенів вводу/виводу, «половина від початкової вартості 3.6 Flash». Але в примітці внизу сторінки: «Вступна ціна діє до 31 грудня 2026. З 1 січня 2027 діятиме **$1.50 / $7.50**». З нового року ціна подвоюється до тієї самої цифри, що й у 3.6, і в тілі анонсу про це не пишуть.

Скепсис із HN, найдошкульніший за добу. Топовий коментар вхопив те саме: «Вступна ціна до грудня 2026 означає, що до наступного року значних змін у Gemini Flash не буде». А другий б'є сильніше: «Вони порівнюють з 5.6 Terra, однак Terra коштує приблизно **вдвічі дешевше**... Плюс треба порівняти зі свіжим Grok 4.6, який виглядає просто краще І дешевше. Важко зрозуміти, чому за таких умов хтось обере 3.7 Flash. **DeepMind ще фронтир-лаба?**»

Чому це важливо. Висновок тут про темп релізів.

Вчора фігурували три фронтир-моделі за одну добу. Сьогодні - четверта, і вона вийшла через три тижні після попередньої версії себе самої. Це вже конвеєр. Практичний наслідок протверезний: будь-яка прив'язка до конкретної моделі старіє за тижні. Вибір моделі під агентну роботу за незалежним заміром (вчорашній пункт 2, Elo 1753) варто перечитувати раз на квартал.

І друге, дрібне, про гігієну читання: вступна ціна - це маркетинг. Half-price до грудня, з січня - повна. Рахуючи вартість чогось стороннього, дивитись треба на ціну після вступного періоду, вона в примітці дрібним шрифтом. [proven - анонс Google прочитаний повністю, усі цифри і примітка про подвоєння ціни дослівно з нього; бенчмарки - **власні заміри Google** проти власної попередньої моделі, незалежних немає; порівняння з Terra і Grok - це **коментар з HN**, самостійних замірів не було]

Google: Introducing Gemini 3.7 Flash · HN-тред, 687 балів - топ доби · @GoogleDeepMind: анонс · @GoogleDeepMind про приріст у кодингу

ТЕМА 3

OpenAI на залізі Cerebras: Sol на швидкості 750 токенів/с. Найцікавіша цифра - 11 годин проти 78

Друга половина доби, і це вже про швидкість як окремий продукт. HN: 488 балів.

Ultrafast mode - новий тариф в OpenAI API, поки для вибраної групи клієнтів. З анонсу OpenAI: GPT-5.6 Sol «до **14× швидше** за стандартну обробку», **до 750 вихідних токенів за секунду**, на залізі Cerebras.

А тепер замір, який вартий усього анонсу, і він з блогу самої Cerebras. Прогнано **Humanity's Last Exam**, 2500 питань рівня PhD:

- **GPT-5.6 Sol Ultrafast: 11 годин 11 хвилин**
- **Claude Fable 5: 78 годин 27 хвилин** - понад три доби безперервних обчислень

Формулювання Cerebras: «Ultrafast пройшов фронтир людського знання за один робочий день, досягнувши порівнянної точності майже у **7× швидше**». Плюс: Ultrafast «в **11× швидше** за Fable 5 і в **5× швидше** за Opus 4.8 у Fast-режимі» за швидкістю виводу, і на GDP-Val - «5.6× пришвидшення від початку до кінця без деградації якості».

Дві поправки, обов'язкові. ① Це заміри Cerebras, тобто сторона, що продає залізо, міряє власну перевагу. Дрібним шрифтом під графіком видно й таке: Sol ганяли з Codex на xhigh 10 липня, а Fable 5 - з Claude Code на xhigh 13-15 липня. Різні харнеси, різні дні, і «порівнянна точність» декларується без числа. ② Топовий коментар на HN вказує на дірку в анонсі OpenAI: «**Інформації про ціну немає**, що може означати або територію "якщо питаєш - не потягнеш", або що вони просто промацують інтерес, перш ніж вирішувати».

Чому це важливо. Прямої дії нуль: це закритий тариф для корпоративних клієнтів.

Але думка стосується того, як влаштована робота асистента. Їхній власний приклад використання - розбір інциденту: «коли спрацьовує алерт, інженерам треба швидко зібрати картину». І цитата дослідника OpenAI: «Раніше доводилось чекати пару хвилин, поки задача добіжить, - тепер вона **завершується до того, як встигаєш переключити контекст**».

Ось це і є справжня зміна, і вона не про 750 токенів. Швидкість перетворює агента з «поставив задачу і пішов» на «думаєш разом». У щоденній роботі це видно в дрібному: 40 секунд очікування відповіді відправляють думку робити щось інше, і розмова рветься. Це пояснює, чому швидкі відповіді відчуваються інакше, ніж довгі прогони, і чому дрібні питання варто ставити асистенту замість запуску повного скрипта. [proven - обидва анонси, OpenAI і Cerebras, прочитані у браузері; всі цифри і цитати дослівні з них; **усі заміри зроблені Cerebras**, продавцем заліза; різні харнеси і різні дати тестів - у них же в примітці; **ціни немає ніде**, тому оцінити вигідність неможливо в принципі]

OpenAI: Previewing Ultrafast mode · Cerebras: технічний розбір і заміри HLE · HN-тред, 488 балів · @OpenAI: анонс · @gdb: «дико бачити Sol на 14×»

ТЕМА 4

«Розуміння - це нове вузьке місце». Джефрі Літт з конкретними техніками, і це пряма відповідь на вчорашню тезу Черіті

Мейджорс про «рев'ю переоцінене»

Цей пункт закриває вчорашню суперечку з протилежного краю, і в ньому єдине за тиждень, що можна застосувати завтра.

Вчора наводилась теза Мейджорс: код-рев'ю «переоцінене, і це найменш цінна частина того, що людина додає до інженерії». Сьогодні на HN вилазить (246 балів) есей Джефрі Літта, письмова версія його доповіді на AI Engineer у липні, і він відповідає рівно на це.

Його теза починається з того, що він відкидає стандартний аргумент. Дослівно: «Одна можлива відповідь: ми розуміємо, щоб верифікувати… Але річ у тім, що **агенти дедалі краще верифікують власну роботу**. І це добре! То де це лишає людей?»

Його власна відповідь здається сильнішою за обидві вчорашні позиції: «**Можна розуміти, щоб брати участь**». Розгортає так: «Це ніколи не один цикл! Проєкт - це багато-багато циклів з агентом. І розуміння системи - це **частина здатності придумати наступну ідею** для її розвитку. Потрібен багатий набір понять у голові, щоб мислити креативно… Якщо цієї побіжності бракує, здатність брати участь у проєкті суттєво обмежена».

Він прив'язує це до поняття «**когнітивний борг**» (Маргарет Сторі і Саймон Вілісон): «Це як техборг: можна якийсь час обходитись без розуміння того, що відбувається, але колись воно вкусить».

Три техніки, і вони не абстрактні:

- **Пояснення.** Скіл `/explain-diff`, яким він користується щодня: агент після роботи видає структурований пояснювач - спершу бекграунд («навчи, що там уже було»), потім інтуїція перед деталями, і лише тоді «літературний дифф»: зміни, викладені прозою в осмисленому порядку, замість купи файлів за абеткою.
- **Мікросвіти** (за Сеймуром Пейпертом): попросити зробити інструмент, щоб зрозуміти самому. Приклади: дебагер для власного Prolog-інтерпретатора, де можна крутити час і бачити стек; і, найкраще, при міграції сайту на новий фреймворк агента попросили зробити «командний центр», де покроково клікалось портування і бачився ефект. Пояснення навіщо: «Це лишило з таким самим розумінням, як робити руками, - **але значно швидше**».
- **Спільні простори** - коли розуміння треба тримати командою, не наодинці.

А тепер скепсис із треду, бо він розумний. Найгостріший коментар: «ШІ має обмеження і галюцинує. **Складний код буде пояснено галюцинованим способом**… Стаття, яку хотілось би прочитати, підказала б, як змусити LLM будувати архітектуру як міцну вежу, а не купу нестабільної грязі». І другий, з іншого краю: «"Код читають" - Мітчелл Гашимото… Важко уявити, як робити продакшн-коміт, який не прочитаний до розуміння. **Наслідки свого коду належать людині; цієї відповідальності агент узяти не може**».

Чому це важливо. Це найпрактичніший пункт тижня.

Роль людини в конструкції «людина + агент» зазвичай описують як верифікацію: погодження на незворотні дії. Літт каже, що верифікація - не головна причина розуміти код. Головна - щоб можна було придумати, що робити далі. І це прямо стосується практики: найкращі зміни в будь-якому власному інструменті приходять з фрази «а зроби, щоб…», і це працює рівно доти, доки в голові тримається модель того, як він влаштований.

Звідси дешева практика: коли наступного разу правиться щось нетривіальне, замість диффу віддавати пояснювач у стилі його `/explain-diff` - спершу як воно було влаштовано, потім навіщо міняється, і тільки тоді що саме змінилось. Одне повідомлення замість «закомітив, ось хеш». Це нічого не коштує і не потребує нового тула. І це точніша версія того, що вчора пропонувалось як «півгодини на тести»: тести ловлять поломку, пояснювач тримає модель системи живою. Потрібне і те, і те, але друге дешевше. [proven - есей прочитаний повністю, цитати дослівні, коментарі з HN item-API; це **доповідь одного інженера з його особистого досвіду**, жодних замірів чи даних під тезою нема; Літт **працює в Notion** і сам це в тексті дисклеймить - частина прикладів рекламує їхні фічі]

Джефрі Літт: Understanding is the new bottleneck · HN-тред, 246 балів · вчорашня позиція Мейджорс у Pragmatic Engineer

---

ТЕМА 5

Як реально влаштовані водяні знаки в тексті - і чому їх завжди можна зняти

Це пункт-борг. **11 серпня** давалась теза «Anthropic вшила невидимий підпис» з застереженням, що механізму не перевірено. **13 серпня** повторено рядком і знову зазначено, що незалежного підтвердження нема. Сьогодні у вікно потрапили два технічні розбори одразу, і борг закривається.

① **Візуальний гайд «How AI text watermarking works»** (declaude.org, 96 балів). Пояснює механізм найкраще з усього, що траплялось. Ключове речення: знаки «працюють, бо живуть не в символах узагалі. Вони живуть у **виборі між словами**».

Як саме: модель на кожному кроці має шортліст слів, кілька з яких однаково доречні. Секретний ключ **фарбує кандидатів у «зелені» і «червоні»** і м'яко підштовхує кубик до зелених. Дві деталі роблять це непомітним: «підштовхування слабке - червоне слово все ще може виграти», і забарвлення не є фіксованою властивістю слова: ключ рахує його з кількох попередніх слів, тому «той самий кандидат зелений після одного префікса і червоний після іншого». Google-івський **SynthID** робить те саме через «крихітний секретний турнір» так, що «в середньому шанси кожного слова лишаються рівно такими, як задумала модель».

І там же - фактаж, якого бракувало три дні: «Google маркує текст з застосунку і вебу Gemini **з 2024** (його API - задокументований виняток), і станом на **серпень 2026** нові

моделі Claude маркують текст на рівні моделі, старіші - згодом».

② **Есей Шона Ґьодеке «Водяні знаки в тексті завжди буде тривіально зняти»** (105 балів) - той самий механізм, але з протилежним висновком і з причиною, чому все це взагалі відбувається.

Причина: **EU AI Act, стаття 50**, яка стає застосовною з **серпня 2026** і вимагає, щоб усі виводи ШІ «піддавались виявленню як штучно згенеровані».

Головний аргумент проти: «Якщо є доступ навіть до відносно слабкої немаркованої LLM, можна зняти SynthID, **попросивши її перефразувати текст**. Оскільки знак притаманний тонкому вибору слів, переформулювання знищує його». І юридична вилка: AI Act вимагає, щоб методи були «інтероперабельні», тобто щоб лаби публікували свій процес маркування. «Важко побачити, як це сумісне з **безпекою через невідомість**, на якій водяні знаки в тексті і тримаються».

Плюс спостереження про **гомогліфи** - заміну звичайного пробілу (U+0020) на схожі (U+2004, U+3000): «Claude Code точно робив це, щоб мітити підозрілі запити від китайських користувачів… відтоді це відкотили». Чесний висновок автора: «Чи використовують OpenAI і Anthropic гомогліфи як водяний знак? **Не впевнений. Але вони точно використовують гомогліфи.**»

Скепсис із треду - і він на боці знаків. Топовий коментар: «Це наче казати "замок Masterlock завжди легко збити молотком". Звісно, але роблячи це, ти **активно вчиняєш шахрайство**, і тягар лягає на тебе». І другий: «Це краще, ніж нічого. Люди недооцінюють цінність правил, які вимагають лише злого наміру і трохи знань, щоб їх порушити».

Чому це важливо. Практичний висновок стосується будь-яких текстів, що проходять через модель: у них може лишатись і статистичний слід у виборі слів, і гомогліфи в пробілах. Друге вичищається механічно перед публікацією (заміна нестандартних пробілів на звичайні - три рядки коду). Перше не знімається нічим, крім переписування руками, і турбувати воно має лише того, хто видає чужий текст за свій.

Методологічний висновок: тема подавалась тричі, і тричі писалось «механізму не перевіряв», хоча матеріал існував уже **11 серпня**. Якщо тема повертається втретє з тією самою поміткою - це сигнал іти шукати першоджерело самому. [proven - обидва матеріали прочитані повністю, механізм (green/red list, SynthID-турнір, гомогліфи) і цитати дослівні; дата застосовності EU AI Act - з есею Ґьодеке; обидва автори - **незалежні дослідники, не лаби**; заява про маркування Claude «на рівні моделі» досі береться з **тексту declaude.org, не з офіційного джерела Anthropic**; сам детектор не запускався і жодного тексту не перевірялось]

How AI text watermarking works - візуальний гайд · Шон Ґьодеке: знаки завжди буде тривіально зняти · HN-тред Ґьодеке, 105 балів · HN-тред візуального гайду, 96 балів

Звіт OpenAI: розрив між «фронтир-фірмами» і рештою виріс з 2.6× до 8.3× за пів року

Компанія публікує дослідження про користь власного продукту, тому це подається разом зі скепсисом. Але цифри варті уваги, бо вони про форму роботи.

Два звіти одразу: **Enterprise Signals** (по клієнтах OpenAI) і робоча стаття **How Organizations Use AI: Evidence from ChatGPT** (84 бали на HN, 69 сторінок).

Що виділяється:

- **«Фронтир-фірми» (топ-10% за використанням) генерують у 8.3× більше вихідних токенів на активного користувача, ніж типові, проти 2.6× у січні.** Розрив потроївся за пів року.
- **Станом на червень Codex дає 64% сукупних вихідних токенів Codex+ChatGPT** у корпоративних клієнтів. Більшість роботи вже агентна.
- **Розрив саме у складних можливостях:** плагінами щотижня користуються **21% активних юзерів у фронтир-фірм проти 9% у типових**, скілами - **19% проти 3%**. У самому OpenAI - 95% співробітників щотижня.
- **Агенти розповзлись за межі інженерії:** з лютого тижневі активні користувачі Codex зросли **×108 у юристів, ×41 у продажах, ×41 у рекрутингу, ×26 у маркетингу, проти ×5 в інженерії**.
- Молодші співробітники користуються ШІ більше за старших: використання найвище на початку кар'єри і падає з ростом сеньйорності.
- Приклад із клієнта (Virgin Atlantic): рефакторинг легасі-коду «за 30 хвилин замість двох тижнів».

Протиотрута, обов'язкова. ① Метрика «вихідні токени на користувача» - це проксі обсягу, не користі. Сама OpenAI це визнає («proxy for depth of use»), але саме на ній тримається вся конструкція «фронтир-розриву». Більше токенів ≠ більше зробленої роботи, а зростання ×108 у юристів - це зростання **з мізерної бази**. ② На HN звіт зустріли холодно: «Чи вбило б цих хлопців з OpenAI навчитись писати білу книгу?... чи це справжній прогрес, чи просто **маркетингова метрика для стейкхолдерів**?». ③ Найтверезіше - топовий відповідач під самим твітом OpenAI: «Черговий проривний розділ у **тонкому мистецтві скорочення витрат**, поки продукт продовжує сповзати».

І окремо Моллік, який дав це у стрічку, пише обережніше за OpenAI: «**Наскільки** використання ШІ підсилює результати фірми - є ранні ознаки, що фірми-ранні-адоптери, які вже й так були успішними, можуть почати відриватись від решти». Причинність не встановлена: можливо, кращі фірми просто більше користуються ШІ. Він формулює це як «early signs».

Чому це важливо. Одна цифра тут пряма ілюстрація: **скіли - 19% проти 3%**. Найбільший розрив у всьому звіті - в тому, чи оформлені повторювані задачі в багаторазові інструкції.

Практична дія з цього одна і дешева: коли ловиться момент, що втретє пояснюється один і той самий процес, це кандидат у скіл. Оформити його раз коштує менше, ніж переказувати те саме вчетверте. [proven - сторінка звіту OpenAI прочитана у браузері, всі цифри дослівні звідти; коментар з HN item-API; це **дані OpenAI про власних клієнтів**, незалежної перевірки нема і бути не може; 69-сторінкова робоча стаття не розбиралась, узято лише те, що на веб-сторінці; причинність не доведена і сама OpenAI цього не стверджує]

OpenAI: From assistance to execution · Робоча стаття, 69 сторінок (pdf) · HN-тред, 84 бали · @OpenAI: про плагіни і скіли · @emollick з обережним формулюванням

ТЕМА 7

ChatGPT почав пам'ятати все, що робиться на комп'ютері - не скріншотами, а подіями

Computer History у десктопному застосунку ChatGPT. З анонсу: «ChatGPT тепер може пам'ятати активність через застосунки і сайти на комп'ютері». 1.1 млн переглядів на твіті, і реакція очікувано нервова.

Тому варто піти у документацію, і там усе цікавіше за твіт. Головне, чого в анонсі нема:

- «Computer History замінює ранній research preview Chronicle, але це перебудована система, не перейменування. **Chronicle використовував скріншоти. Computer History записує події взаємодії і НЕ захоплює екран чи звук**». Саме цієї різниці не видно з твіту.
- «**Вимкнено за замовчуванням**» для Pro, Business і Enterprise. Для Business/Enterprise адміністратор мусить явно надати доступ, перш ніж співробітник зможе увімкнути.
- «Історія починається тільки після того, як вирішено її увімкнути». Контроль: які застосунки і сайти дають внесок, пауза з меню-бару, перегляд і видалення в будь-який момент.
- **Недоступно в ЕЕА, Швейцарії і Британії** (в анонсі - обтічне «доступ згодом»).
- Вимагає **Memories**, не працює з API-ключем чи через Amazon Bedrock. Поки тільки macOS.

Заявлена користь: «підхопити, де зупинилися», знайти нещодавню роботу і, найцікавіше, «коли Computer History помічає **повторювану роботу**, запис у таймлайні може запропонувати скіл або автоматизацію».

Чому це важливо. Порівняння з пам'яттю, яку агентні конвеєри будують самі, напрошується. Типова пам'ять асистента - це те, що йому сказали, і те, що він зробив: файли в git, які читаються очима як markdown, і які не бачать нічого, чого їм не показали. Тут же - потік подій з усіх застосунків у хмару. Перше вужче, друге ширше, і вибір між ними - це вибір, як багато віддавати назовні.

Фіча «помітив повторювану роботу → запропонував скіл» - це рівно висновок попереднього пункту, тільки виведений з телеметрії кліків замість того, що людина втретє пише той самий шматок.

Одне варто сказати прямо: фіча за замовчуванням вимкнена і в ЄС недоступна, тож сама собою вона не ввімкнеться. [proven - офіційна документація прочитана в markdown-версії, всі обмеження і цитати дослівні; **фічу не бачено і не вмикано**; «не захоплює екран» - це **заява OpenAI в документації**, незалежної перевірки нема]

Документація: Computer History · @OpenAI: анонс · @OpenAI: як увімкнути і де недоступно

ТЕМА 8

«Обирай нудні технології» вилізла на HN з 2015 року і зібрала 291 бал

Стаття Дена Мак-Кінлі одинадцятирічної давнини раптом набирає 291 бал, і топовий коментар складається з двох слів: «добре зістарілась».

Суть - концепція **токенів інновацій**: «Скажімо, кожна компанія отримує приблизно **три токени інновацій**. Витрачати їх можна як завгодно, але запас фіксований надовго… загальна тенденція - **переоцінювати вміст свого гаманця**».

Другий за рейтингом коментар пояснює, чому це живе: «Це одна з найулюбленіших статей… "токени інновацій" - **одна з найкорисніших концепцій за кар'єру** як PM / інженерного лідера. Вона допомагає робити правильні компроміси і, ще більше, **пояснювати ці компроміси колегам** усіх рівнів».

Чому це важливо. Цей пункт стоїть восьмим свідомо, як противага до пунктів 1, 2 і 3 цього ж випуску.

Що було в стрічці за одну добу: новий відкритий харнес з 68к зірок, нова модель через три тижні після попередньої, новий тариф на екзотичному залізі. Спокуса «а давай спробуємо» після такого дня природна. І рівно на це стаття відповідає: токенів три, і в будь-якій сталій конструкції вони вже витрачені. Все інше в ній свідомо нудне: bash, python, cron, sqlite, markdown. Це і є стратегія.

Висновок з усього дня в один рядок: дивитись і читати варто, переносити до себе - майже ніколи, крім дрібних запозичень на кшталт логу контексту з пункту 1, які нічого не ламають. [proven - стаття і коментарі прочитані, цитати дослівні; це **есей 2015 року**, не дослідження - цінність у формулюванні, не в даних]

Choose Boring Technology (2015) · HN-тред, 291 бал

ТЕМА 9

Практики про те, що змінилось: «люди перестали казати, що інженерія скінчилась»

Зріз настрою, і він вартий окремого пункту, бо змінює тон вчорашнього випуску.

@levie (Box), сьогодні вранці: «**Ліквідація інженерів була однією з найдикіших гіпотез. Просто абсурдно хибна.** Інженерам просто дали потужний інструмент, що пришвидшує розробку будь-чого потрібного. Звісно, їхня цінність **насправді зростає**». Це реакція на пост Сема Ламберта: «Хтось помітив, що люди перестали казати, що інженерія програмного забезпечення скінчилась?»

@shl (Сахіл Лавінгія), вчора: «**Читати код після ШІ - це як читати книжки після соцмереж**». Одним рядком, але точно, і в тему пункту 4. Він же, пізніше, з самоіронією про рекурсію агентів: «ШІ керує бізнесом. Потім ШІ дивиться на ШІ, що керує бізнесом, і дає фідбек. Потім ШІ дивиться на ШІ, що дає фідбек...».

@amasad (Replit), з іншого краю спектра і найрадикальніше: «Наступного року користування комп'ютером стане опційним. Робота радикально зміниться.»

А от протиотрута - з того ж дня і з іншого краю. @agupta, репостнутий Гаррі Таном: «Економіка знань **розшаровується на дві підекономіки: ① те, що потребує фронтир-моделей, і ② те, де різниця між DeepSeek V4 Pro / Grok 4.6 і GPT-5.6 Sol / Claude Fable непомітна**». Це найтверезіший спосіб думати про вчорашній «день трьох моделей»: для більшості задач розрив у 1-2 бали індексу не має значення, і питання звучить як «**задача взагалі з першої категорії?**».

Чому це важливо. Вчора було чотири джерела поспіль з тезою «дефіцит перемістився з написання коду в судження». Сьогодні три практики кажуть м'якше: інженер став дорожчим. Разом з Літтом (пункт 4) це складається в картину, де цінність - у тому, щоб розуміти систему настільки, аби знати, що робити з нею далі. Пункт 4 дає конкретний інструмент цю цінність підтримувати. [fuzzy - це **думки практиків у твітах**, не дослідження і не заміри; жодних даних під жодним із цих тверджень нема; подається як зріз настрою стрічки, не як факт про ринок]

@levie: «абсурдно хибна гіпотеза» · @shl: читати код після ШІ · @shl про рекурсію агентів · @amasad: «комп'ютер стане опційним» · @agupta про дві підекономіки

ТЕМА 10

Anthropic пояснила, що можна і не можна робити з виводом Claude. 88 балів на HN і тред, який складається зі злості

Короткий пункт, але прямо про правила користування інструментом, на якому тримається багато агентної роботи.

Довідкова стаття Anthropic: «Коли користувач користується Claude, **він володіє виводом**, згенерованим з його вводів. Однак є важливі обмеження на використання цих виводів **для тренування моделей ШІ**».

Що дозволено: тренувати моделі, які не конкурують з моделями Anthropic - класифікатори і тули: аналіз тональності, категоризація контенту, підсумовування, витягування інформації, семантичний пошук, виявлення аномалій. Плюс вбудовувати вивід у свої застосунки, генерувати контент для клієнтів, структурувати дані, покращувати внутрішні процеси.

Що заборонено: «загальні чат-боти», «моделі для відкритої генерації тексту», «використання виводу як **тренувальних цілей**», «зворотна розробка методів тренування».

Обґрунтування Anthropic: моделі, натреновані на виводі Claude, «не матимуть їхніх запобіжників», плюс прямо про бізнес: клієнти «використовують їхню інфраструктуру та інвестиції, щоб побудувати **прямих конкурентів** їхньому сервісу».

Тред - це переважно злість, і вона зрозуміла. Топовий коментар: «**Дозволу не питали, коли тренували свою модель на моєму виводі**». Другий, з чорним гумором: «Додав "© 2024 - Ніяких прав для лицемірів" у футер свого сайту. Claude сканує всі сторінки щонайменше раз на тиждень відтоді, як з'явився. І що тепер».

Чому це важливо. Типова особиста автоматизація під це не підпадає з запасом: записувати вивід моделі у файли пам'яті, щоб завтрашня сесія їх прочитала, - це «покращення внутрішніх процесів», прямо в дозволеному переліку.

Межа проходить в іншому місці. Ідея «а давай натренуємо маленьку локальну модель на архіві сесій» - це рівно те, що заборонено: «використання виводу як тренувальних цілей». Таку ідею варто відкидати одразу. Інакше тиждень проєктування впреться в ліцензію. [proven - довідкова стаття Anthropic прочитана в браузері, переліки дозволеного і забороненого дослівні; коментарі з HN item-API; це **політика, не закон**, і як вона застосовується на практиці - з тексту не видно; юридична оцінка тут не дається]

Anthropic: Can I use my Outputs to train an AI model? · HN-тред, 88 балів

misc - коротко, що ще варте ока

• **MiniMax-Music3 - відкриті ваги для генерації музики** - @multimodalart: «SoTA генерація пісень, тепер відкриті ваги, **8B LLM + 2.7B DiT**, що перетворюють промпт + текст пісні на повні пісні», і головне - «влазить у скромні споживчі GPU». П'ята цеглина в лінію «локальне і своє». Ваги на Hugging Face

• **Codex у десктопному ChatGPT для Linux - прев'ю** (450 балів). Показово: 450 балів за портування застосунку на лінукс - це про те, наскільки розробники хочуть цих агентів локально

• **Spaghettifying DRAM** (535 балів) - атака на пам'ять, досяжна з софту. Коментар для контексту: «це наче **програмно-досяжна версія апаратної атаки** з batteringram.eu». Жанр той самий, що вчорашній баг SQLite, а найдотепніший коментар треду: «навіщо вони пишуть свої розбори за допомогою ШІ?!»

• **@lennysan про візит до стоматолога**: «Візит до стоматолога - напрочуд цікавий спосіб відстежувати впровадження ШІ. **Пів року тому - нуль ШІ. Сьогодні ШІ аналізує рентгени і транскрибує розмову.**» Найкоротший опис того, як швидко це просочується в звичайні професії

• **@awilkinson: Grok Bot = Openclaw для нормальних людей** - і поруч його ж технічне спостереження: «дивовижно, яку різницю робить **правило Spotify - усе вантажиться менш ніж за 60 мілісекунд**... ті самі результати, але клікати відчувається миттєво». Те саме, що в пункті 3, тільки з боку UI

• **@anna_y_zhang про трасування агентів**: «трейси агентів - **найважливіший новий вхідний матеріал** для компанії, що самовдосконалюється. Але купа трейсів сама по собі нічого не покращує», і далі три компоненти циклу, перший з яких «**мозок компанії** - спільна пам'ять того, що команда знає: рішення, контекст, чому саме так». Дуже близько до пункту 1

• **Exa перетнула 100 мільярдів сторінок** - @WilliamBryk: «щойно перевалили за 100 мільярдів». За їхньою ж оцінкою, Google ~1 трлн, Bing ~500 млрд. Маркер того, що пошуковий індекс перестав бути недосяжною фортецею

• **@typesfast про дві лінзи** (Райан Петерсен, Flexport), 942 лайки: «Є **дві лінзи**, крізь які можна дивитись на світ: ① робити більше роботи - погано; ② **розв'язувати проблеми - це те, як заробляєш**. Завжди і назавжди користуйся лінзою 2. **Більше роботи - це добре, якщо вона дає розв'язувати важливі проблеми**»

• **Куди подівся старий веб - простежили 657 607 посилань** (145 балів) - дослідження гниття посилань. Прямо в тему щоденної перевірки лінків курлом

• **Дата-центри в містечку у Вашингтоні** (6.1 тис. лайків): «Містечко... **побудувало нову старшу школу, лікарню, бібліотеку, каналізацію, поліцію і пожежну частину. Бідність упала з 29% до 6%.**» Один кейс, не дослідження, але рідкісний контраргумент до стандартної історії «дата-центри висмоктують громаду»
