SEO 2026

Пошук в еру AI

Михайло Оплаканець aka Werter

SEO 2026

Пошук в еру AI

Як працює органічний пошук у час Google AI Mode, LLM та агентів

Перше видання

2026

ПРО ВИДАННЯ

SEO 2026: Пошук в еру AI

Автор: Михайло Оплаканець aka Werter

Перше видання, 2026.

Пошукові системи, інтерфейси, правила та AI-продукти змінюються швидко. Факти, функції та документація в цій редакції перевіряються станом на 12 вересня 2026 року. Для тем, які залежать від часу, у тексті наведено дату зрізу та пряме посилання на джерело.

У книзі немає вигаданих статистик, кейсів, URL, цитат або "факторів ранжування". Якщо механізм не підтверджений, він подається як гіпотеза або практичне спостереження, а не як факт.

ПРИСВЯТА

Тим, хто не звик вірити SEO на слово.

Тим, хто відкриває Search Console, логи й SERP раніше, ніж повторює чужий кейс.

І тим, хто хоча б раз ламав сайт, втрачав трафік, знаходив причину і після цього став сильнішим спеціалістом.

ПРО КНИГУ

Це не книга про те, "що таке SEO"

SEO у 2026 році не закінчується на позиції URL у Google. Пошук тепер складається з кількох шарів: виявлення сторінки, сканування, рендеринг, вибір канонічної версії, індексація, відбір документів під запит, ранжування, генеративна відповідь, цитування і вже після цього клік або дія користувача. Якщо дивитися тільки на позиції, частина системи просто зникає з поля зору.

Ця книга пояснює весь ланцюг від URL до результату бізнесу. Без магії, без "секретних факторів" і без спроб продати новий ярлик замість SEO. AI Overviews, AI Mode, ChatGPT, Bing/Copilot, GEO, AEO та LLMO тут розглядаються не як окремий всесвіт, а як нові способи отримання інформації, які частково спираються на ті самі базові речі: доступність документа, релевантність, авторитетність, структуру сайту, посилання, сутності та якість даних.

Основна модель книги проста: зрозуміти систему, подивитися на дані, сформувати гіпотезу, перевірити її, виміряти результат і лише після цього масштабувати. Саме тому тут багато конкретних URL, методологій досліджень, запитів до API, регулярних виразів, таблиць, логів, схем, прикладів Search Console і технічних розборів. Теорія потрібна лише там, де вона допомагає прийняти рішення на реальному сайті.

Книга написана з позиції практикуючого SEO, який працює з конкурентними нішами, affiliate-проєктами, технічними обмеженнями, контентом, посиланнями, міграціями та масштабуванням. Вона не намагається зробити складну систему простою. Вона намагається зробити її зрозумілою.

Що ви отримаєте після прочитання

Не набір чеклістів "перевірте title, canonical і sitemap", а модель мислення, з якою можна діагностувати сайт навіть тоді, коли інтерфейс Google знову зміниться. Ви будете розуміти, де саме URL губиться в пошуковому ланцюгу, як відрізняти проблему індексації від проблеми відбору, як читати GSC без хибних висновків, як оцінювати AI-видимість, як будувати інформаційну архітектуру, як працювати з великими сайтами та як не плутати кореляцію з причинністю.

Писав її для людей, яким уже не треба пояснювати, що таке canonical

Основний читач цієї книги - SEO-фахівець середнього або старшого рівня, SEO Lead чи Head of SEO. Людина, яка вже працювала з Search Console, знає, навіщо потрібні robots.txt і sitemap, розуміє базову логіку зовнішніх посилань та аналізу ключових слів і хоче перейти від виконання окремих задач до розуміння системи.

Книга також підійде технічним SEO-фахівцям, менеджерам SEO-продукту, affiliate SEO, внутрішнім та агентським командам, людям, які відповідають за великі контентні ресурси, електронну комерцію, маркетплейси, SaaS або міжнародні сайти. Найбільше користі вона дасть там, де SEO вже перестало бути набором сторінок і стало системою з тисячами або мільйонами URL, кількома країнами, шаблонами, командами й залежностями.

Це не найкраща перша книга про SEO. Я свідомо не пояснюю на пів сторінки, що таке title, чому потрібен HTTPS або що зовнішнє посилання веде з іншого сайту. Базові поняття тут пояснюються лише тоді, коли без цього неможливо зрозуміти механіку. Якщо ви вже працюєте в SEO і регулярно ловите себе на питаннях "чому Google обрав не ту канонічну версію?", "чому сторінку просканували, але не проіндексували?", "чому позиція не падає, а кліків менше?", "чому конкурент цитують у AI, а нас ні?" - книга написана саме для цього рівня.

Кому вона не потрібна

Якщо потрібен список зі ста "факторів ранжування", універсальна формула кількості ключових слів у тексті або обіцянка вивести будь-який сайт у топ за 30 днів, тут цього не буде. У складних системах майже ніколи немає однієї причини, а сильний SEO відрізняється не кількістю чеклістів, а якістю діагностики.

Не обов’язково читати від першої до останньої сторінки

Перші дві частини я раджу прочитати послідовно. Вони формують модель пошуку, якою ми будемо користуватися далі: як сторінка стає відомою пошуковій системі, як її сканують, рендерять, канонікалізують, індексують і відбирають під запит. Без цієї основи розмови про AI-пошук, посилання чи масштабування швидко перетворюються на набір окремих тактик.

Далі книгу можна використовувати як робочий довідник. Перед міграцією відкрити глави про canonical, hreflang, redirects і моніторинг. Перед запуском тисяч шаблонних сторінок - блок про масштабне генерування сторінок. При падінні кліків - глави про Search Console, логи та діагностику SERP. При роботі з AI-видимістю - частину про відбір джерел, цитування і вимірювання.

У тексті я спеціально розділяю підтверджені факти, зовнішні дослідження, спостереження та робочі гіпотези. Якщо Google щось прямо документує, це так і написано. Якщо дані показують лише кореляцію, я не називаю її фактором ранжування. Якщо механізм виглядає логічно, але публічно не підтверджений, він залишається гіпотезою.

Мова та терміни

Основна мова книги - українська. Англійські слова залишаються лише там, де це усталений технічний термін, назва продукту або точний синтаксис: SEO, SERP, URL, API, noindex, PageRank, AI Overview, AI Mode, schema.org. Якщо український відповідник не погіршує точність, я використовую український.

Посилання та джерела

Усі URL подані простим текстом, щоб їх можна було перевірити незалежно від електронної версії. Для швидкозмінних тем пріоритет мають джерела 2026 року, потім 2025 і 2024. Старі матеріали використовуються тоді, коли йдеться про фундаментальний принцип, історію механізму, патент або первинне джерело, яке досі актуальне.

Спочатку докази, потім думки

Найгірша звичка в SEO - говорити впевнено там, де даних немає. Пошукова система закрита, її механізми постійно змінюються, а більшість публічних SEO-досліджень є спостережними. Тому будь-яке сильне твердження в цій книзі повинно мати основу.

1. Підтверджений факт

Офіційна документація Google, Bing, Microsoft, schema.org, Chromium, Search Console API, патенти або інші первинні технічні джерела. Навіть тут важливий контекст: патент доводить існування описаного механізму, але не доводить, що він використовується в чинній системі ранжування зараз.

2. Сильне зовнішнє дослідження

Дані Ahrefs, Semrush, SISTRIX, Similarweb, Pew Research, академічні роботи та інші джерела використовуються лише тоді, коли зрозумілі вибірка, період, методологія та обмеження. Кореляція не стає причинністю тільки тому, що вибірка велика.

3. SEO-експеримент

Для тестів важливі контроль, період, розмір вибірки, змінна, яку ми змінювали, та сторонні фактори. Експеримент без опису умов корисний як спостереження, але слабкий як доказ.

4. Дані реального сайту

Search Console, GA4, серверні логи, краулер, індексація, позиції та зовнішні SEO-сервіси. Це часто найкорисніший рівень для прийняття рішень, але й тут треба контролювати сезонність, бренд, країну, пристрій, апдейти Google, зміни шаблону та інші фактори.

5. Робоча гіпотеза

Практичне спостереження, яке варто тестувати, але не можна продавати як підтверджений фактор. У тексті такі речі позначаються прямо: "моя робоча гіпотеза", "на практиці я спостерігав" або "механізм публічно не підтверджений".

Що це змінює на практиці

Коли я бачу сильне твердження, перше питання - не "чи згоден я?", а "який це рівень доказу?" Офіційна документація відповідає на одні питання, контрольований тест - на інші, а дані одного сайту найкраще працюють для рішень саме на цьому сайті. Змішування цих рівнів і створює більшість SEO-міфів.

У кожній технічній главі далі буде той самий принцип: спочатку механіка і те, що ми точно знаємо, потім дані, потім практичне рішення і лише після цього гіпотези. Це повільніше за універсальний чекліст, але значно надійніше, коли на кону великий сайт або реальні гроші.

ПРО АВТОРА

Михайло Оплаканець aka Werter

SEO-спеціаліст, який працює з органічним пошуком з 2008 року. Основний професійний фокус - конкурентні ніші, affiliate та iGaming, технічний SEO, контентні системи, посилання, багатодоменні проєкти, міжнародний пошук і автоматизація.

За роки роботи проходив ролі від операційного SEO до рівня Senior, Lead та Head: будував стратегії, запускав і масштабував сайти, працював з міграціями, індексацією, Search Console, логами, контентом, посиланнями та SEO-процесами для команд. Окремий інтерес - системи, які зменшують ручну роботу: API, Python, SQL, автоматичні перевірки, генерація та контроль якості сторінок.

Виступав на профільних SEO-заходах, зокрема SEMPRO, NaZapad, Content Marketing Day та SEMPRO Club. Працював із продуктами й проєктами в різних нішах, але найбільше професійного досвіду накопичив там, де SERP конкурентний, помилки дорогі, а результат легко перевірити цифрами.

Ця книга - спроба зібрати не "правильні відповіді на SEO", а спосіб мислення, який витримує зміни інтерфейсів, апдейти й нові назви. Якщо після прочитання ви частіше ставите правильне питання до даних, а не просто швидше виконуєте чекліст, книга зробила свою роботу.

ВІД АВТОРА

Чому я написав цю книгу

Я почав займатися SEO у 2008 році. Відтоді змінювалися SERP, алгоритми, інструменти, способи будувати посилання, формати контенту і навіть те, що ми вважали пошуком. Але одна проблема залишалася стабільною: індустрія дуже любить прості пояснення складних систем.

Колись це були "200 факторів ранжування", потім LSI, TF-IDF, E-E-A-T як формула, Core Web Vitals як універсальний ключ до топу. Тепер до цього списку додалися GEO, LLMO, контент, спеціально адаптований під AI, llms.txt та десятки нових правил, які часто з’являються раніше, ніж дані, що могли б їх підтвердити.

Я не вважаю, що AI нічого не змінив. Змінив дуже багато. Один запит може перетворюватися на серію підзапитів. Користувач може отримати відповідь до кліку. Сайт може бути використаний як джерело, не займаючи звичної позиції в класичному SERP. Частину інформаційного попиту тепер закриває генеративний інтерфейс. Вимірювання видимості стало складнішим.

Але з цього не випливає, що SEO почалося заново. Сторінку все одно треба знайти. Її треба отримати по HTTP, відрендерити, правильно канонікалізувати й проіндексувати. Система все одно повинна зрозуміти, про що документ, зіставити його з інформаційною потребою, оцінити джерело й вирішити, чи варто його показувати або використовувати у відповіді. Саме тому фундамент книги - не набір AI-хаків, а повний шлях інформації через пошукову систему.

Другий мотив простіший. Мені бракувало української книги про SEO, яку можна відкрити не для мотивації і не для базового визначення термінів, а під час роботи. Коли треба розібрати кластер канонічних URL. Коли після міграції з індексу зникає половина

шаблону. Коли GSC показує стабільні позиції та менше кліків. Коли команда хоче згенерувати 200 тисяч сторінок і запитує, чи достатньо просто відкрити їх для Googlebot.

Тому ця книга навмисно технічна. У ній будуть SQL, regex, API, серверні логи, моделі даних, формули, схеми, приклади помилок і рішення. Але технічність тут не самоціль. Кожна глава повинна відповідати на одне питання: яку практичну перевагу отримає сильний SEO після цих сторінок?

Я також не хочу робити вигляд, що ринок складається тільки з безпечних практик, що відповідають рекомендаціям пошукових систем. Affiliate, дроп-домени, платні посилання, PBN, редиректи, parasite SEO та інші речі існують. Їх треба розуміти так само добре, як офіційні рекомендації Google. Але розуміти механіку не означає приховувати ризик. Там, де практика суперечить spam policies, це буде сказано прямо.

Найважливіше правило книги: не вірте навіть цій книзі на слово. Відкривайте джерело. Дивіться на дату. Перевіряйте власні дані. Відокремлюйте факт від припущення. SEO достатньо складне, щоб не додавати до нього ще й вигадані правила.

ЗМІСТ КНИГИ

43 глави про пошук, системи, дані та AI

Нижче - структура повного видання. Нумерація сторінок буде додана після фінальної верстки всієї книги.

ЧАСТИНА IV. АВТОРИТЕТ І КОНКУРЕНЦІЯ

ЧАСТИНА V. AI-ПОШУК

ЧАСТИНА VI. МАСШТАБ

ЧАСТИНА VII. ВИМІРЮВАННЯ

ЧАСТИНА VIII. РИЗИК

ЧАСТИНА IX. SEO-КОМАНДА У 2026

Глава 40. Дорожня карта, пріоритизація та бюджет: як керувати SEO як продуктом

Глава 41. Команда, KPI та звітність: що має контролювати Lead і Head

ЧАСТИНА X. АГЕНТНИЙ ВЕБ

Глава 42. AI-агенти, машиночитний бізнес і транзакції без класичного SERP

Глава 43. Роль SEO після класичного пошуку: що залишиться, коли інтерфейс знову зміниться

ВСТУП

SEO у 2026 році: що насправді змінилося

Пошук не помер. Зникла простота, з якою ми його вимірювали.

Ще кілька років тому більшість SEO-звітів можна було звести до чотирьох питань: скільки показів, яка позиція, який CTR і скільки кліків. Модель була неповною, але для великої частини задач працювала. Сайт потрапляв у видачу, користувач бачив результат, клікав і переходив на сторінку.

У 2026 році між запитом і кліком з'явилося значно більше подій. Пошукова система може розкласти складне питання на кілька підзапитів, знайти різні документи, використати окремі фрагменти як джерела, сформувати відповідь і показати посилання вже після синтезу інформації. Користувач може поставити уточнювальне питання, не відкривши жодного сайту. А може побачити бренд у відповіді, запам'ятати його і повернутися через інший канал.

Через це SEO більше не можна описувати лише позицією. Потрібно бачити щонайменше чотири рівні: чи може сторінка потрапити в індекс, чи може система відібрати її під інформаційну потребу, чи отримує вона видимість у конкретному інтерфейсі і чи перетворюється ця видимість на бізнес-результат.

Індексованість -> відбір -> видимість -> конверсія

Ця послідовність буде проходити через усю книгу. Якщо сторінка не індексується, немає сенсу сперечатися про CTR. Якщо індексується, але не потрапляє у відбір під потрібний клас запитів, проблема вже не в robots.txt. Якщо вона потрапляє у відбір і навіть цитується, але не приводить до дії, технічно SEO може працювати, а бізнесово - ні.

Саме так я пропоную дивитися на сучасний пошук: не як на один рейтинг із десяти посилань, а як на ланцюг рішень. Кожен етап має свої сигнали, метрики, помилки та способи діагностики.

AI не скасував фундамент

Навколо AI-пошуку легко впасти в одну з двох крайностей. Перша: SEO закінчилося, тепер існує GEO. Друга: нічого не змінилося, достатньо робити те саме, що й раніше. Обидві позиції слабкі.

Фундамент не зник. Пошуковій системі все ще потрібен доступний документ, зрозуміла структура, коректна канонічна версія, релевантний зміст, внутрішні та зовнішні зв’язки, достатня якість джерела. Але поверх цього фундаменту з’явився новий шар: генеративний відбір, синтез, цитування, згадки і взаємодія без обов’язкового переходу на сайт.

Тому я не відділяю AI SEO від SEO. Я розглядаю AI-пошук як розширення середовища, в якому працює оптимізація. Деякі механізми справді нові. Деякі старі механізми отримали нову роль. А частина того, що продається як нова дисципліна, є звичайним SEO з новою етикеткою.

Чого тут не буде

Не буде тверджень на кшталт "Google любить довгі тексти", "цей фактор дає +20% позицій" або "AI краще цитує абзаци по 40 слів", якщо для цього немає нормальної доказової бази. Не буде вигаданих кейсів із красивим графіком. Не буде спроб вивести універсальний рецепт із одного сайту.

SEO рідко дає лабораторні умови. Саме тому сильний спеціаліст має не лише знати інструменти, а й уміти читати невизначеність. Велика частина цієї книги присвячена не тому, "що робити", а тому, як зрозуміти, чи спрацювало саме те, що ви зробили.

Що змінилося найбільше

Найсильніша зміна не в тому, що Google додав AI-блок у SERP. Змінилася економіка інформації. Типову відповідь стало дешевше створити, дешевше узагальнити й часто дешевше показати користувачу без переходу на джерело. Через це зменшується цінність контенту, який не додає нової інформації.

Одночасно виросла цінність речей, які важко синтезувати без першоджерела: власні дані, експерименти, інструменти, калькулятори, порівняння на реальних вибірках, унікальні набори даних, досвід, методики та матеріали, які щось доводять, а не просто переказують.

Для SEO-команд це означає неприємну, але корисну зміну: виробляти більше сторінок стало легше, а виробляти сторінки, які справді заслуговують існувати, стало важче.

Навіщо ця книга

Я хочу, щоб після неї ви могли відкрити незнайомий сайт і не починати з чекліста. Спочатку зрозуміти модель: як сайт побудований, як URL потрапляють у систему, де формується дублікатність, як розподіляються внутрішні посилання, що індексується, які класи запитів дають попит, що змінилося в SERP і де саме втрачається результат.

Якщо ця модель побудована правильно, конкретні інструменти можна замінити. Ahrefs можна замінити Semrush. Screaming Frog - іншим краулером. SQL - Python. Інтерфейс Search Console теж зміниться. Але спосіб думати про систему залишиться.

З цього й почнемо. Не з "факторів ранжування". З того, що сьогодні відбувається між запитом користувача і результатом, який ми бачимо в даних.

ЧАСТИНА I / ПОШУК У 2026

ГЛАВА 1

Пошук більше не дорівнює SERP

Що реально змінилося до 2026 року, як AI вплинув на кліки і чому ранжування, відбір документів, цитування та трафік тепер треба вимірювати окремо

КОРОТКО

Класичний SEO-ланцюжок не зник: сторінку все ще треба знайти, відрендерити, визначити канонічну версію, проіндексувати й отримати з індексу. Змінилася поверхня видачі та економіка кліку. Одна й та сама сторінка тепер може одночасно бути органічним результатом, джерелом для AI Overview, посиланням у AI Mode або взагалі не отримати клік, хоча її інформація була використана у відповіді. Тому позиція більше не є достатньою одиницею вимірювання.

До появи AI-пошуку SEO часто зводили до лінійної моделі: запит -> SERP -> позиція -> CTR -> клік -> конверсія. Вона й раніше була спрощенням через блоки швидких відповідей, локальну видачу, товарні блоки, зображення та інші елементи SERP, але для більшості задач працювала достатньо добре. У 2026 році ця модель уже системно недооцінює те, що відбувається між запитом і переходом на сайт.

Причина не в тому, що Google перестав ранжувати документи. Навпаки: Google прямо описує AI Overviews та AI Mode як функції, що спираються на індекс пошуку, основні системи ранжування й оцінки якості, RAG та розгортання запиту. Базова інфраструктура пошуку нікуди не зникла. Новий шар з’явився після відбору документів: система може витягнути кілька джерел, використати їх як фактичну

основу відповіді, синтезувати текст і показати користувачеві лише частину цих джерел. Для SEO це принципова різниця: ранжування і цитування пов’язані, але це не одна й та сама подія.[1][2]

1.1. Що саме зламалося в старій моделі SEO

Проблема старої моделі не в позиціях як таких. Позиція все ще має сенс там, де користувач бачить класичну органічну видачу і вибирає між результатами. Проблема в тому, що позиція більше не описує всю видимість сайту в пошуку.

Один запит користувача може породити кілька внутрішніх підзапитів. Google називає цей механізм query fan-out; далі я називатиму його розгортанням запиту. Офіційний приклад: для запиту про газон, зарослий бур’янами, система може паралельно шукати найкращі гербіциди, способи видалення бур’янів без хімії та методи профілактики. Кожен підзапит формує власний набір сторінок-кандидатів. Відповідь для користувача одна, але фактичний простір пошуку ширший за буквальний текст початкового запиту.[2]

Для SEO це означає, що логіка точного входження стає ще менш корисною як основний спосіб моделювання попиту. Сторінка може бути релевантною не до вихідної фрази, а до одного з підзапитів, окремого фрагмента тексту або зв’язку між сутностями, який система використала під час розгортання запиту. Це не доказ того, що «ключові слова померли». Це доказ того, що окреме ключове слово давно не дорівнює всьому простору можливих запитів.

Рис. 1.1. Від запиту користувача до посилання на джерело: ранжування, відбір документів, опора на джерела і цитування є різними етапами.

Авторська схема. Механіка узагальнена на основі Google Search Central: https://developers.google.com/search/docs/appearance/ai-features

Поверхня	Що бачить користувач	Що важливо для SEO
Класична органічна видача	Список ранжованих URL та інші елементи SERP	Ранжування, можливість показу фрагмента опису, CTR, відповідність наміру
AI Overviews	Згенерована відповідь і посилання на джерела поруч із класичною видачею	Відбір документів, опора на джерела, можливість цитування, органічна видимість
AI Mode	Генеративна відповідь як основний інтерфейс з уточнювальними запитами	Розгортання запиту, охоплення під час відбору, сторінки-джерела
Bing/Copilot	Генеративні відповіді з посиланнями на джерела	Цитування, URL джерел, запити для пошуку джерел, динаміка цитувань
ChatGPT Search	Відповідь із вебджерелами та цитуванням; система може робити кілька пошукових запитів	Доступ для пошукового робота, можливість використання як джерела, переходи на сайт, наявність цитування

Таблиця 1.1. У 2026 році видимість у пошуку складається з кількох різних форматів. Механіка між ними частково спільна, але метрики не взаємозамінні.

МОЯ РОБОЧА МОДЕЛЬ

Ранжування != відбір документів != цитування != згадка. Сторінка може добре ранжуватися і не бути процитованою. Може потрапити у відбір, але не ввійти до фінальної відповіді. Може бути процитована без позиції в топ-3 органічної видачі. Бренд може бути згаданий як сутність без прямого URL. Якщо аналітика зводить усе це до середньої позиції, вона втрачає частину системи.

1.2. Дані: AI-відповідь реально змінює економіку кліку

Тут важливо не впадати у дві крайності. Перша: «AI забрав увесь SEO-трафік». Це неправда. Друга: «нічого не змінилося, бо користувачі все одно клікають». Це теж погано узгоджується з даними. Станом на 2026 рік є щонайменше два незалежні набори даних, які показують один напрям: коли AI-відповідь закриває інформаційну потребу безпосередньо у видачі, частота переходів на зовнішні сайти падає.

ДАНІ: AHREFS, 300 000 КЛЮЧОВИХ СЛІВ

Оновлене дослідження Ahrefs від 4 лютого 2026 року порівнювало 150 000 ключових слів з AI Overview та 150 000 інформаційних ключових слів без AI Overview. Для першої органічної позиції фактичний CTR у групі з AI Overview у грудні 2025 року був близько 1,57%. Контрфактична оцінка, побудована через зміну CTR контрольної інформаційної групи, становила близько 3,73%. Різниця: приблизно -58%. Це кореляційне дослідження, а не рандомізований експеримент, тому коректне формулювання таке: наявність AI Overview корелювала з приблизно 58% нижчим CTR для першої позиції.[3]

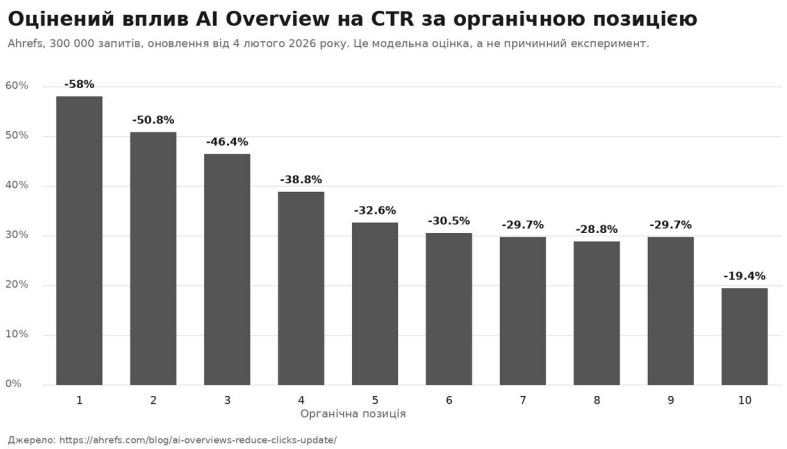


Рис. 1.2. Оцінений вплив AI Overview на органічний CTR залежно від позиції.

Дані: Ahrefs, 300 000 ключових слів, 4 лютого 2026 року. https://ahrefs.com/blog/ai-overviews-reduce-clicks-update/ Це модельна оцінка на основі спостережень, а не рандомізований причинний експеримент.

Органічна позиція	Оцінений вплив AI Overview на CTR
1	-58.0%
2	-50.8%
3	-46.4%
4	-38.8%
5	-32.6%
6	-30.5%
7	-29.7%
8	-28.8%
9	-29.7%
10	-19.4%

Таблиця 1.2. Оцінка Ahrefs для грудня 2025 року. Важливо: це модельна оцінка впливу, а не прямий причинний ефект, виміряний експериментом.[3]

Сама цифра -58% сильна, але її користь для Senior SEO не в тому, щоб вставити її в презентацію. Важливіша методологія. Ahrefs не просто порівняв поточний CTR запитів з AI Overview і без нього. Дослідники використали базовий рівень до масового поширення AI Overview та контрольну групу інформаційних запитів, щоб оцінити, яким міг би бути CTR запитів з AI Overview без цього фактора. Це все одно не усуває сторонні змінні: структуру запитів, склад SERP, зміни поведінки користувачів, відмінності між пристроями, сезонність та інші елементи видачі. Але така методологія сильніша за просте твердження «у запитів з AI Overview CTR нижчий».

ДАНІ: PEW RESEARCH CENTER, 68 879 ПОШУКОВИХ СЕСІЙ GOOGLE

Pew проаналізував дані про поведінку 900 дорослих у США за березень 2025 року: 68 879 унікальних пошукових сесій Google, із них 12 593 містили AI-відповідь. На сторінках з AI-відповіддю користувачі переходили на традиційний результат пошуку у 8% сесій; без AI-відповіді — у 15%. На посилання всередині AI-відповіді клікали приблизно в 1% сесій. Після SERP із AI-відповіддю пошукову сесію завершували у 26% випадків проти 16% без неї.[4]

Це інший тип дослідження. Pew не намагався оцінити CTR конкретної позиції. Дослідники дивилися на поведінку реальних користувачів після сторінки результатів Google. Напрям виявився тим самим: коли у видачі є AI-відповідь, частка сесій без переходу на зовнішній сайт зростає.

Ще одна корисна деталь із Pew: у їхній вибірці AI-відповідь з'являлася приблизно у 18% усіх пошукових сесій Google. Імовірність сильно залежала від форми запиту: лише 8% одно- або двослівних запитів мали AI-відповідь, тоді як серед запитів із 10 і більше словами частка становила 53%. Для запитів, що починалися з питальних слів на кшталт «хто», «що», «коли», «чому», частка сягала 60%.[4]

Практичний висновок важливіший за середню частку AI Overview по ринку. Якщо сайт живе на коротких навігаційних або транзакційних запитах, середній ринковий показник майже нічого не каже про ризик саме для цього сайту. Якщо сайт залежить від довгих інформаційних запитів, контенту, що вирішує конкретні проблеми, або сторінок із відповідями на питання, частота появи AI-відповідей може бути значно вищою. Тому ризик AI Overview треба рахувати на власному наборі запитів, а не переносити середній відсоток із дослідження на весь сайт.

1.3. Позиція без CTR тепер ще небезпечніша метрика

У старій звітності часто було достатньо побачити: позиція стабільна, покази ростуть, кліки падають. Далі починалися припущення про заголовок, фрагмент опису, сезонність або слабший попит. У 2026 році до цього списку треба додати зміну самого формату видачі.

Сценарій «покази зростають, кліки падають, середня позиція стабільна» більше не можна розбирати без сегментації за типом видачі. Якщо сторінка частіше з'являється у генеративному форматі, вона може отримувати видимість, яка не конвертується у звичайний клік так само, як класичне органічне посилання. Водночас не можна автоматично вважати кожне падіння CTR наслідком AI: CTR залежить від частки брендових і небрендових запитів, пристрою, країни,

довжини запиту, складу SERP, розподілу позицій та зміни самого попиту.

АНАЛІТИЧНЕ ПРАВИЛО

Не пояснюйте падіння кліків через AI, поки не розклали дані хоча б за країною, пристроєм, брендовими й небрендовими запитами, типом запиту, типом сторінки, діапазоном позицій, появою в генеративних форматах і часовим періодом. Фраза «AI забрав кліки» без такої декомпозиції — така сама слабка аналітика, як «оновлення Google забрало трафік».

1.4. Що Google офіційно підтверджує про AI-пошук

У 2026 році Google описав механіку досить прямо, щоб прибрати частину ринкового шуму навколо «GEO-хаків». Генеративні функції пошуку Google спираються на основні системи ранжування й оцінки якості. Для отримання актуальної інформації вони використовують RAG: знаходять релевантні сторінки в індексі й використовують їх як фактичну основу для відповіді. Додатково система може розгортати початковий запит у кілька паралельних підзапитів.[2]

Щоб сторінка могла з’явитися як посилання на джерело в AI Overviews або AI Mode, вона має бути проіндексована й придатна для звичайного показу в Google з фрагментом опису. Google прямо зазначає, що окремих технічних вимог лише для AI-функцій немає.[1] Це критично: якщо URL не проходить базовий ланцюжок пошуку, «оптимізація під AI» поверх цього не вирішує проблему.

МІФ / ДАНІ / ВИСНОВОК

МІФ: для Google AI-пошуку потрібні llms.txt, спеціальний Markdown, «AI schema» або обов’язкове дроблення сторінки на дрібні фрагменти. ДАНІ: Google у поточній документації прямо пише, що пошук Google не використовує llms.txt як спеціальний сигнал; не вимагає дробити контент на маленькі фрагменти; не вимагає окремої schema.org-розмітки для генеративного пошуку. ВИСНОВОК: ці речі можуть бути корисними іншим системам або внутрішнім процесам, але підстав продавати їх як вимогу Google для ранжування в AI немає.[2]

Тут є важливий нюанс. «Немає спеціальної розмітки для AI» не означає, що структуровані дані не потрібні. Google окремо уточнює, що вони залишаються корисними для звичайних елементів пошуку та розширених результатів. Просто не треба продавати їх як магічний фактор GEO. Те саме із заголовками, таблицями чи FAQ: вони можуть покращувати читабельність, структуру інформації та її виділення, але немає офіційної вимоги дробити сторінку на спеціальні фрагменти «для LLM».

1.5. Найбільша зміна для контенту: типова інформація дешевшає

AI змінив не лише SERP, а й економіку контенту. Якщо відповідь складається із загальнодоступних фактів, які десятки сторінок переказують однаково, пошуковій системі дедалі простіше сформувати її без окремого переходу на ще одну схожу сторінку. У документації для AI-пошуку Google робить акцент на унікальній інформації, експертному внеску, власному досвіді та матеріалах, які не є простим переказом того, що вже є в мережі.[2]

Для SEO-команди це перекладається на операційну мову дуже просто. «Ще одна стаття на ту саму тему» перестає бути достатньою одиницею виробництва контенту. Сторінка повинна мати хоча б один власний інформаційний актив: дані, реальний тест, інструмент, набір даних, зріз цін, методику порівняння, скріншоти, власні спостереження, калькулятор, дані з API, регіональну специфіку або іншу практичну цінність, якої немає у взаємозамінних конкурентів.

Це не романтична вимога «писати краще». Це економічна вимога створювати інформацію, яку генеративна система не може дешево синтезувати з десятка взаємозамінних документів. Якщо ваш текст можна майже без втрат відновити з топ-10 конкурентів, у нього слабка інформаційна перевага.

1.6. Вимірювання у 2026 році: що вже можна бачити напряму

Ще у 2025 році одна з головних проблем AI-пошуку полягала в тому, що SEO бачив кінцевий результат, але майже не мав даних про те, де й як сайт з'являється у генеративних відповідях. У 2026 році ситуація стала кращою.

GOOGLE SEARCH CONSOLE

3 червня 2026 року Google анонсував окремі звіти Search Console для генеративних функцій AI. Станом на 31 серпня 2026 року їх розгорнули глобально. В окремому звіті Google показує генеративні покази, сторінки, країни, пристрої та дати з погодинною, денною, тижневою і місячною деталізацією. У самому анонсі кліки не заявлені як окрема метрика цього звіту.[5]

Google нарешті дає власні дані про генеративну видимість, але це ще не повна воронка. Ми можемо бачити, які URL отримували покази в генеративних функціях, де і коли. Але стандартний звіт не відповідає повністю на питання: яке конкретне цитування дало який клік і яку конверсію. Для цього доводиться поєднувати кілька джерел даних.

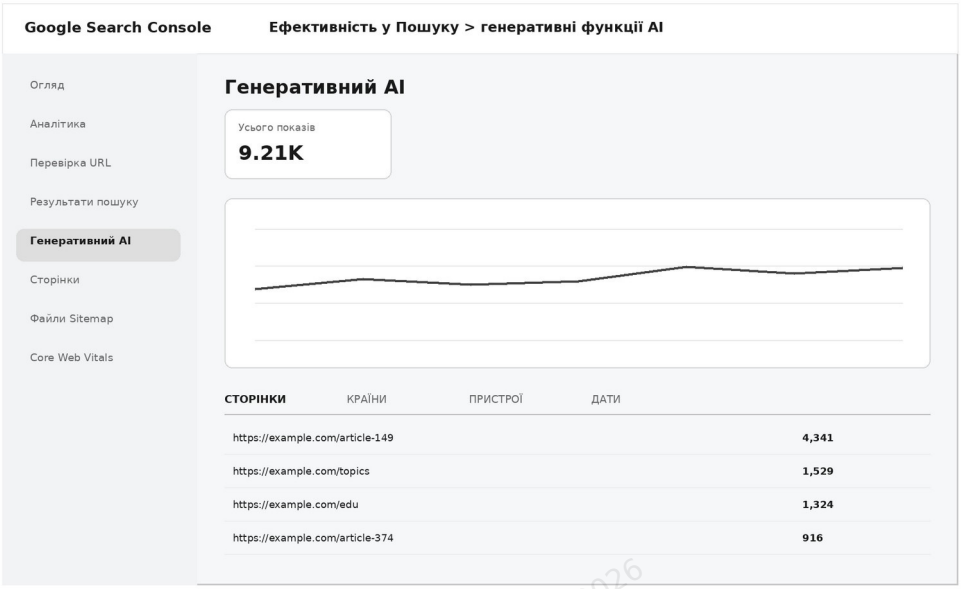


Рис. 1.3. Search Console: окремий звіт про ефективність генеративних функцій AI у пошуку.

Схематичне відтворення офіційного скріншота Google, без доданих метрик. Оригінал: https://developers.google.com/search/blog/2026/06/gen-ai-performance-reports

BING WEBMASTER TOOLS

З 10 лютого 2026 року Bing Webmaster Tools має AI Performance у публічному попередньому доступі. Панель показує загальну кількість цитувань, середню кількість цитованих сторінок, активність цитування на рівні URL, динаміку видимості та запити, які AI використовував для пошуку контенту, що згодом було процитовано. Bing окремо попереджає, що ці запити показуються як вибірка, а не як повний журнал.[6]

Bing дає ще глибший рівень діагностики. Якщо URL часто цитують, але набір запитів, за якими його знаходять, вузький, сторінка може мати сильну видимість лише в одному тематичному кластері. Якщо більшість цитувань припадає на кілька URL, можна порівняти ці сторінки з іншими й перевірити різницю в охопленні теми, чіткості сутностей, щільності фактів, актуальності, посилальній вазі та відповідності запиту. Але навіть тут кількість цитувань не можна називати «позицією в AI». Bing прямо пояснює, що активність

цитування не дорівнює місцю, авторитетності чи ролі сторінки в конкретній відповіді.[6]

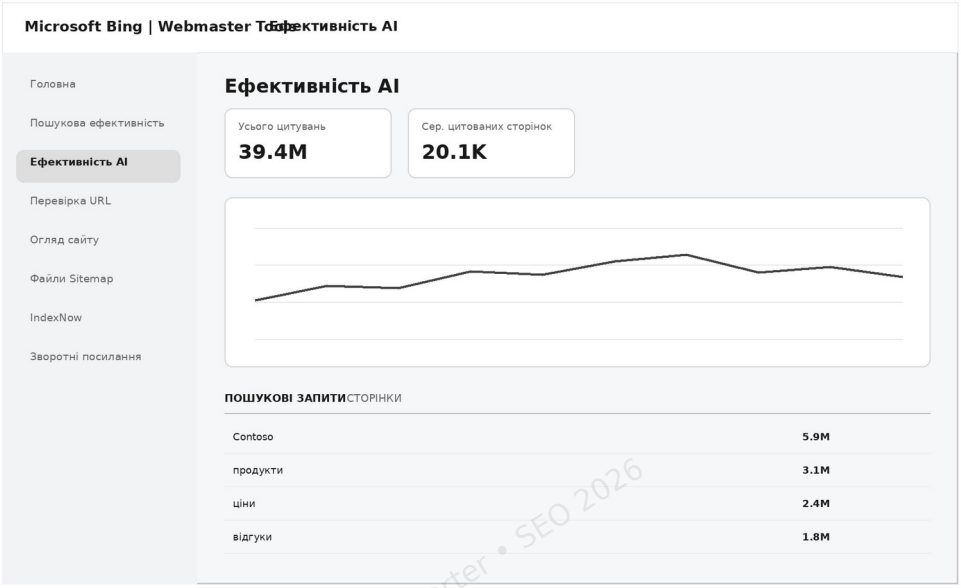


Рис. 1.4. Bing Webmaster Tools AI Performance: цитування, цитовані сторінки та запити, за якими система знаходила джерела.

Схематичне відтворення офіційного скріншота Microsoft Bing. Оригінал: https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview

1.7. Практична модель вимірювання для SEO-команди

Я б не будував окрему «GEO-панель», відірвану від SEO-аналітики. Корисніше розкласти видимість на чотири шари й дивитися, на якому саме етапі губиться цінність.

Шар	Ключове питання	Дані
Індексованість	Чи може система використати URL взагалі?	Статус індексації, канонічна версія, robots, можливість показу фрагмента опису, обхід і рендеринг

Шар	Ключове питання	Дані
Потрапляння у відбір	Чи потрапляє сторінка до набору документів-кандидатів?	Запити для пошуку джерел, тематичне охоплення, відповідність темі й сутностям, закономірності цитування сторінок
Видимість	Чи бачить користувач сторінку або джерело?	Органічні покази, генеративні покази, цитування, URL джерел, згадки бренду
Бізнес-результат	Чи створює ця видимість цінність для бізнесу?	Кліки, сесії, допоміжні конверсії, ліди, дохід, переходи з AI та інших джерел

Таблиця 1.3. Мінімальна модель, яка не змішує технічну доступність, потрапляння у відбір, видимість і бізнес-результат.

Після цього можна рахувати власні похідні метрики. Вони не є метриками Google і їх не треба видавати за стандарт індустрії. Це робочі аналітичні коефіцієнти, які дозволяють порівнювати власні дані між періодами.

Вимірювання у 2026 році: не змішувати різні показники

Один “показник AI-видимості” часто змішує різні події та різні джерела даних.

Індексованість	Канонічні URL в індексі	GSC Indexing, URL Inspection, логи
Потрапляння у відбір	Запити та теми, за якими сторінку відбирають	Bing AI Performance, контрольні набори запитів
Видимість у пошуку	Органічні покази та позиції	Google Search Console
Видимість в AI	Генеративні покази, цитування, URL джерел	GSC Generative AI, Bing AI Performance
Трафік	Органічні кліки, переходи з AI	GSC, GA4 / аналітика
Бізнес-результат	Ліди, дохід, допоміжні конверсії	GA4, CRM, BI

Правило: динаміку можна порівнювати лише для метрик з однаковим визначенням і знаменником.

Рис. 1.5. Модель вимірювання: індексованість, потрапляння у відбір, видимість, трафік і бізнес-результат не можна зводити до одного показника.

Авторська схема. Метрики та джерела даних відповідають моделі вимірювання цієї глави.

АЛГОРИТМ: 4 ПОХІДНІ МЕТРИКИ

1) Частка генеративних показів = генеративні покази / усі покази в пошуку. Використовуйте як індикатор динаміки, а не як «частку всіх запитів»: агреговані покази різних форматів не обов'язково дорівнюють кількості унікальних пошуків. 2) Концентрація цитувань = цитування топ-N URL / усі цитування сайту в Bing. Показує, наскільки AI-видимість залежить від кількох сторінок. 3) Охоплення запитів для пошуку джерел = кількість релевантних тематичних кластерів, де сайт цитують, / кількість цільових кластерів у вашій моделі. Це внутрішня метрика, і знаменник формує ваша власна класифікація ключових слів та сутностей. 4) Частка конверсій за участю AI = конверсії, де вимірюваний AI-перехід був основним або допоміжним каналом, / сесії з таких переходів. Для ChatGPT та інших джерел окремо перевіряйте коректність атрибуції.

Найважливіше: не змішувати різні знаменники. Генеративні покази в Google, цитування в Bing і переходи з ChatGPT описують різні події. Їх не можна просто скласти в один «показник AI-видимості» без нормалізації та чіткого визначення, що саме він означає. Інакше панель виглядає красиво, але перестає бути діагностичним інструментом.

1.8. Як перевіряти падіння органічних кліків: діагностика на рівні Senior SEO

Якщо органічні кліки впали на 20%, я не починаю з гіпотези «це AI». Я починаю з декомпозиції. Спочатку перевіряю попит: покази, Google Trends, динаміку рік до року, сезонність. Потім дивлюся розподіл позицій: чи справді вони не просіли всередині важливих груп сторінок і запитів. Далі перевіряю склад SERP: AI Overviews, локальні й товарні блоки, відео, новини та інші елементи. І лише після цього оцінюю, чи падіння віддачі в кліках збігається зі зростанням генеративних показів.

Для цього достатньо простої формули: віддача в кліках = кліки / покази для конкретного сегмента. Не для всього ресурсу одразу, а для однакового набору запитів, сторінок, країн і пристроїв. Якщо віддача падає за стабільних або кращих позицій, а генеративні покази

одночасно ростуть, гіпотеза про вплив нового формату видачі стає сильнішою. Якщо одночасно змінилися структура запитів або розподіл позицій, причинність не доведена.

УВАГА

Не порівнюйте «CTR до AI» і «CTR після AI» на різній структурі запитів. Якщо у 2025 році сайт ранжувався переважно за короткими комерційними запитами, а у 2026 році виросла частка довгих інформаційних запитів, загальний CTR майже гарантовано зміниться навіть без AI. Спочатку порівняйте однакові когорти, і лише потім робіть висновок.

1.8.1. Як читати патерни в даних без гадання

Окремий сигнал майже ніколи не дає готового діагнозу. Потрібна комбінація сигналів. Нижче не «алгоритм Google», а діагностична матриця: вона показує, яку гіпотезу перевіряти першою, а не оголошує причину без тесту.

Що бачимо	Перша гіпотеза	Що перевірити далі
Позиції впали + кліки впали + генеративні покази без змін	Втрата позицій або релевантності	Когорти URL і запитів, конкуренти, дати оновлень Google, зміна наміру, технічні зміни
Позиції стабільні + покази стабільні + віддача в кліках впала + генеративні покази зросли	Канібалізація кліків новими елементами видачі	Наявність AI Overview у вибраному наборі запитів, розріз за пристроями й країнами, склад SERP, порівняння періодів до і після зміни
Генеративні покази зросли + органічні кліки не ростуть або падають	Видимість без еквівалентної цінності в кліках	Які сторінки отримують покази, чи зростає видимість комерційних сторінок, допоміжні конверсії
Цитування Bing зростають + запити для пошуку джерел сконцентровані в одній темі	Вузьке тематичне охоплення під час відбору	Розширити фактичне наповнення сусідніх цільових кластерів; порівняти цитовані й нецитовані сторінки

Що бачимо	Перша гіпотеза	Що перевірити далі
Згадки бренду й цитування зростають + переходи зростають + конверсії слабкі	Невідповідність після кліку	Намір посадкової сторінки, CTA, пропозиція, швидкість, відстеження та шлях до конверсії

Таблиця 1.4. Діагностична матриця: комбінації сигналів і наступна перевірка. Це модель для постановки гіпотез, а не доказ причинності.

1.9. Що це змінює в SEO-стратегії прямо зараз

Перша зміна: SEO-план більше не можна будувати лише навколо задачі «підняти позицію». Для частини інформаційних кластерів позиція може рости, а додаткового трафіку майже не буде. Пріоритизація повинна враховувати потенціал кліків у конкретному форматі видачі, а не лише частотність запиту та поточну позицію.

Друга зміна: дослідження ключових слів має перейти від списку фраз до моделі простору запитів. Я все одно використовую ключові слова, частотність, SERP і кластеризацію, але фінальна одиниця планування для мене не окреме ключове слово, а кластер наміру, сутності або проблеми. Це краще відповідає розгортанню запитів і реальній структурі сучасного пошуку.

Третя зміна: контент-план повинен відрізняти взаємозамінні матеріали від контенту, який важко скопіювати. Якщо сторінка існує лише тому, що «у конкурента є така стаття», її стратегічна цінність слабка. Якщо сторінка містить власні дані, інструмент, методику порівняння, робочий процес або інформацію, якої немає в інших джерелах, вона має більше шансів залишатися потрібною і користувачеві, і системам відбору документів.

Четверта зміна: звітність за видимістю треба розділити. Органічна позиція, органічний клік, генеративний показ, цитування, URL джерела, запит, за яким сторінку знайшли для AI, перехід із AI та конверсія — це різні події. Коли їх змішують в одну метрику, команда перестає розуміти, на якому саме етапі виникла проблема.

П’ята зміна: технічний SEO стає не менш, а більш важливим. Для посилань на джерела в AI Google використовує ту саму базову придатність сторінки до пошуку: вона має бути проіндексована й

придатна для показу з фрагментом опису. Якщо на сайті хаос із канонічними версіями, проблеми з рендерингом, noindex, м'які 404, зламана внутрішня перелінковка або сторінки просто не потрапляють до індексу, «GEO-оптимізації» нема на що спиратися.[1]

1.10. 60-хвилинна перевірка: чи AI вже змінює вашу органіку

Це не повний аудит. Це швидка діагностична перевірка, після якої вже зрозуміло, чи проблема взагалі варта глибшого дослідження.

Час	Що робити
00-10 хв	Вибрати 20-50 посадкових сторінок, які дали найбільшу абсолютну втрату кліків рік до року або період до періоду. Дивитися не на відсоток падіння, а на фактично втрачені кліки.
10-20 хв	Для кожної сторінки розкласти запити на брендові/небрендові, за наміром і довжиною. Перевірити діапазони позицій, країни та пристрої.
20-30 хв	Подивитися звіт Search Console про генеративні функції AI: чи є ці URL серед сторінок із генеративними показами, як змінилася їхня видимість у часі, у яких країнах і на яких пристроях.
30-40 хв	У Bing Webmaster Tools перевірити цитовані URL та запити, за якими AI знаходив джерела. Дивитися не лише на кількість цитувань, а на теми, що приводять до них.
40-50 хв	Перевірити SERP для репрезентативного набору запитів через трекер позицій або SERP API: наявність AI Overview, інші елементи видачі, зміни її складу. Зафіксувати однакову країну й пристрій.
50-60 хв	Сформувати три окремі гіпотези: падіння попиту, втрата позицій, втрата кліків через зміну формату видачі. Не об'єднувати їх в одну причину. Для кожної визначити дані, які можуть її спростувати.

Таблиця 1.5. Швидка діагностика, яка відділяє вплив формату видачі від падіння попиту або позицій.

1.11. Де закінчуються факти і починаються гіпотези

Ми точно знаємо, що Google використовує індекс пошуку, системи ранжування й оцінки якості, RAG та розгортання запиту у генеративних функціях пошуку.[1][2] Ми точно знаємо, що Google не вимагає llms.txt, спеціальної розмітки для AI або окремої schema.org-розмітки для появи в цих функціях.[2] Маємо сильні зовнішні дані, що AI-відповіді корелюють із нижчою частотою кліків в інформаційній видачі.[3][4] Також маємо власні звіти Google та Bing, які вже дозволяють бачити генеративні покази, цитування, цитовані сторінки й запити, за якими система знаходила джерела.[5][6]

Чого ми не знаємо з публічних джерел: точних ваг сигналів для потрапляння в AI-відповідь; універсальної формули ймовірності цитування; повної залежності між органічною позицією і цитуванням; повного набору внутрішніх сигналів відбору документів; стабільного «GEO-фактора ранжування», який можна оптимізувати одним технічним трюком. Якщо хтось продає це як встановлений факт, вимагайте докази, методологію та розмір вибірки.

ВИСНОВОК ГЛАВИ

SEO у 2026 році не став іншою професією. Він став багатошаровішою системою вимірювання. Базовий ланцюжок пошуку залишився, але після відбору документів з’явився генеративний шар, який може забирати частину цінності кліку й створювати новий тип видимості. Тому сильний SEO більше не питає лише «де ми ранжуємося?». Він питає: чи доступний URL, чи потрапляє він у відбір, чи використовують його як джерело, чи бачить його користувач, чи отримуємо клік і чи перетворюється ця видимість у бізнес-результат.

Джерела до глави

Перевірено станом на 12 вересня 2026 року. Для динамічних продуктів та інтерфейсів перед фінальним друком потрібен повторний фактчек.

1. Google Search Central. AI features and your website.. https://developers.google.com/search/docs/appearance/ai-features Офіційна документація: умови появи в AI Overviews та AI Mode, відбір документів і розгортання запиту.

2. Google Search Central. Optimizing your website for generative AI features on Google Search.. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide Офіційна документація 2026 року: RAG, робота з джерелами, технічні вимоги та спростування поширених міфів про AI-оптимізацію.

3. Ahrefs. Update: AI Overviews Reduce Clicks by 58%.. https://ahrefs.com/blog/ai-overviews-reduce-clicks-update/ 4 лютого 2026 року. Спостережне дослідження на 300 000 ключових слів із модельною оцінкою впливу AI Overview на CTR.

4. Pew Research Center. Google users are less likely to click on links when an AI summary appears in the results.. https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/ Дослідження поведінки 900 дорослих у США: 68 879 унікальних пошукових сесій Google.

5. Google Search Central Blog. Introducing Search Generative AI performance reports in Search Console.. https://developers.google.com/search/blog/2026/06/gen-ai-performance-reports 3 червня 2026 року; примітка оновлена після глобального розгортання звіту 31 серпня 2026 року.

6. Bing Webmaster Blog. Introducing AI Performance in Bing Webmaster Tools Public Preview.. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview 10 лютого 2026 року. Цитування, цитовані сторінки, запити для пошуку джерел і активність на рівні URL.

7. OpenAI Help Center. Searching the web with ChatGPT.. https://help.openai.com/en/articles/9237897 Актуальна документація продукту: веб-пошук, цитування та джерела.

8. OpenAI Help Center. Publishers and Developers - FAQ.. https://help.openai.com/en/articles/12627856-publishers-and-developers-faq Актуальні рекомендації для видавців і пошукових роботів, зокрема щодо OAI-SearchBot та відстеження переходів.

ГЛАВА 2

AI Overviews, AI Mode та LLM-пошук: одна задача, різні інтерфейси

Чому одна й та сама інформаційна потреба дає різні джерела в Google, ChatGPT, Perplexity та Copilot, і як це правильно вимірювати в SEO

Стан продуктів і документації: 12 вересня 2026 року

У 2026 році вже недостатньо сказати, що сторінка «видима в AI». Видима де саме? В AI Overview у звичайній видачі Google? В AI Mode? У ChatGPT Search? У Perplexity? У Copilot? Це не косметична різниця. Один і той самий запит може привести ці системи до подібного висновку, але через різні підзапити, різні набори документів і різні правила показу джерел.

Моя робоча модель проста: не існує однієї універсальної «AI-видачі». Є кілька інтерфейсів доступу до інформації. Вони частково використовують пошукові індекси, ранжування, відбір документів і генеративні моделі, але не зобов’язані однаково інтерпретувати запит або цитувати ті самі URL. Саме тому оптимізація «під AI взагалі» без уточнення платформи майже завжди перетворюється на набір загальних порад.

Рис. 2.1. Одна інформаційна потреба може пройти через різні пошукові інтерфейси й завершитися різним набором джерел.

Авторська схема. Вона описує різницю між інтерфейсами, а не внутрішню архітектуру конкретної платформи.

2.1. AI Overview і AI Mode вже не можна вважати двома розмірами одного блоку

На рівні продукту Google сам розділяє ці режими. AI Overview з’являється всередині звичайної сторінки результатів, коли системи Google вважають генеративну відповідь корисним доповненням до класичного пошуку. AI Mode створений для складніших запитів, порівнянь, дослідження теми й уточнювальних питань. Обидва формати можуть використовувати розгортання запиту, але Google прямо зазначає: вони можуть працювати на різних моделях і техніках, тому відповідь та набір посилань можуть відрізнятися.[1]

У січні 2026 року Google ще сильніше зблизив ці інтерфейси: користувач отримав можливість ставити уточнювальні питання прямо з AI Overview і переходити в продовження діалогу. На I/O 2026 Google вже описував це як безшовний перехід від сторінки результатів з AI Overview до розмови в AI Mode із збереженням контексту.[2] Для користувача межа між «SERP» і «чатом» стає менш помітною. Для SEO ця межа, навпаки, критична, бо різні формати можуть показати різні URL.

Станом на оновлення Google від 31 серпня 2026 року AI Overviews має понад 2,5 млрд активних користувачів на місяць, а AI Mode перевищив 1 млрд активних користувачів на місяць.[3] Це не частка всіх пошукових запитів і не показник кліків. Це показники аудиторії продуктів. Але вони достатні, щоб не розглядати генеративний пошук як лабораторний експеримент, який можна винести за межі основної SEO-стратегії.

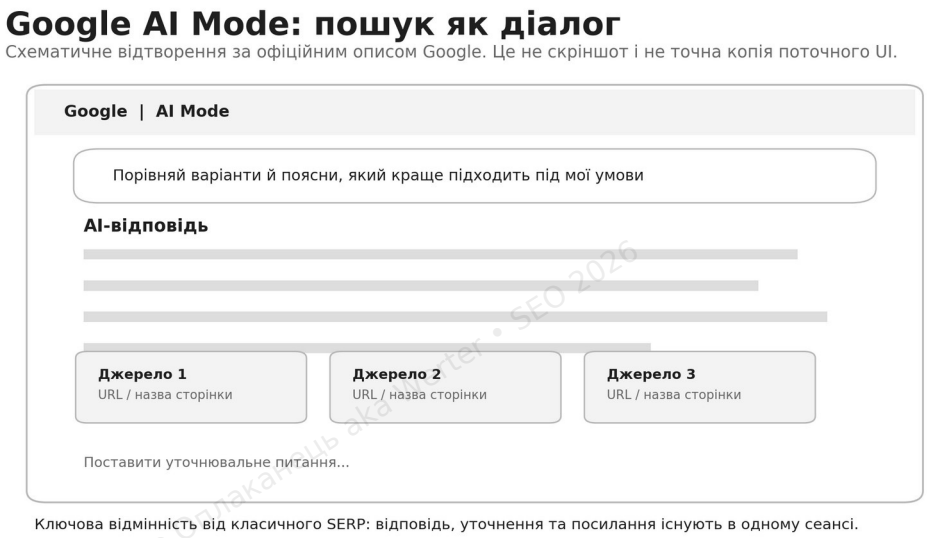


Рис. 2.2. AI Mode як окремий пошуковий інтерфейс: відповідь, джерела і продовження діалогу знаходяться в одному сеансі.

Схематичне відтворення за офіційним описом Google. Оригінальний матеріал: https://blog.google/products-and-platforms/products/search/ai-mode-development/

Таблиця 2.1. Те, що AI Overview і AI Mode живуть усередині Google Search, не робить їх одним і тим самим форматом.[1][2]

Ознака	AI Overview	AI Mode
Точка входу	Звичайна сторінка Google Search	Окремий AI-режим / продовження з AI Overview
Коли корисний	Коротке узагальнення складнішого запиту	Дослідження, порівняння, багатокрокові питання
Уточнювальні питання	Так, з переходом у діалог	Так, це базова частина інтерфейсу
Розгортання запиту	Може використовуватися	Може використовуватися

Ознака	AI Overview	AI Mode
Набір джерел	Не зобов’язаний збігатися з AI Mode	Не зобов’язаний збігатися з AI Overview
Базова SEO-придатність	URL має бути проіндексований і придатний до показу з фрагментом	Ті самі базові вимоги Google Search

ДАНІ

Google прямо повідомляє, що AI Overview та AI Mode можуть використовувати різні моделі й техніки. Це важливіше за будь-який SEO-термін: якщо джерельний набір різний на рівні продукту, одна «позиція в AI» не може бути коректною універсальною метрикою.

2.2. Дані: схожі відповіді не означають схожі джерела

Найкращий спосіб побачити різницю між AI Overview і AI Mode - не сперечатися про інтерфейс, а подивитися на перетин джерел. Ahrefs у дослідженні, опублікованому в грудні 2025 року, зіставив 540 тис. пар відповідей для аналізу цитувань і 730 тис. пар для аналізу тексту. Дані були зібрані в США у вересні 2025 року.[9]

Результат добре показує, чому модель «AI Mode просто розгортає AI Overview» слабка. Семантична схожість відповідей була високою: близько 86%. Тобто системи часто приходили до близького змістового висновку. Але перетин цитованих URL становив лише 13,7%, а перетин унікальних слів - близько 16%.[9] Інакше кажучи, висновок може бути схожим, а шлях до нього - зовсім іншим.

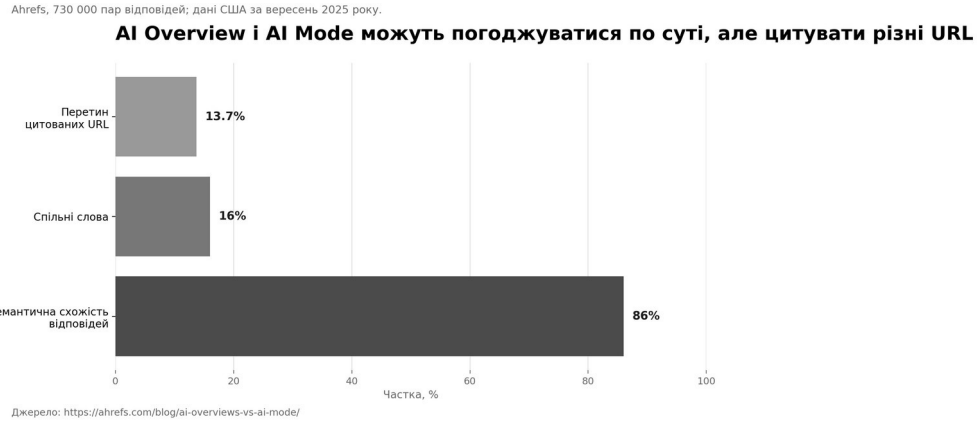


Рис. 2.3. AI Overview і AI Mode часто погоджуються по суті, але цитують різні URL.

Дані: Ahrefs, https://ahrefs.com/blog/ai-overviews-vs-ai-mode/ Вибірка: 540 тис. пар для URL та 730 тис. пар для аналізу відповідей, США, вересень 2025 року.

У тому самому дослідженні AI Mode у середньому формував приблизно у чотири рази довші відповіді й містив у 2,5 раза більше сутностей людей і брендів. Якщо бренд уже згадувався в AI Overview, імовірність його появи в AI Mode у цій вибірці становила 61%. Водночас 11% AI Overviews і 3% відповідей AI Mode не містили цитованих джерел.[9] Ці цифри не треба переносити як константу на 2026 рік: моделі й інтерфейси вже змінилися. Їхня цінність в іншому - вони показують сам факт розходження між системами на великій вибірці.

Для практичного SEO це означає, що навіть усередині Google треба розділяти щонайменше дві задачі: присутність у AI Overview та присутність в AI Mode. Якщо сайт добре цитують в одному форматі, це ще не доказ, що другий формат бачить його так само. І навпаки, відсутність у AI Overview не означає автоматичну відсутність у AI Mode.

2.3. ChatGPT Search: пошуковий запит формується вже після питання користувача

У ChatGPT логіка входу інша. Користувач не обов’язково вирішує, що зараз він «іде в пошук». ChatGPT може запустити веб-пошук автоматично, коли для відповіді потрібна актуальна інформація, або

користувач може вибрати пошук вручну.[4] Це змінює не лише інтерфейс, а й те, що SEO вважає запитом.

OpenAI прямо описує механіку переписування: початкове питання може бути перетворене в один або кілька цільових пошукових запитів. Після перегляду перших результатів система може відправити додаткові, більш специфічні запити.[4] Тобто фраза, яку написав користувач, не обов'язково збігається з фразою, за якою система шукала ваш документ.

Це важлива причина, чому пряме зіставлення початкового запиту, топ-10 Google і джерел ChatGPT часто дає слабкий перетин. Якщо ChatGPT розклав потребу на кілька інших пошукових формулювань, порівнювати його джерела тільки з SERP для початкової фрази методологічно недостатньо.

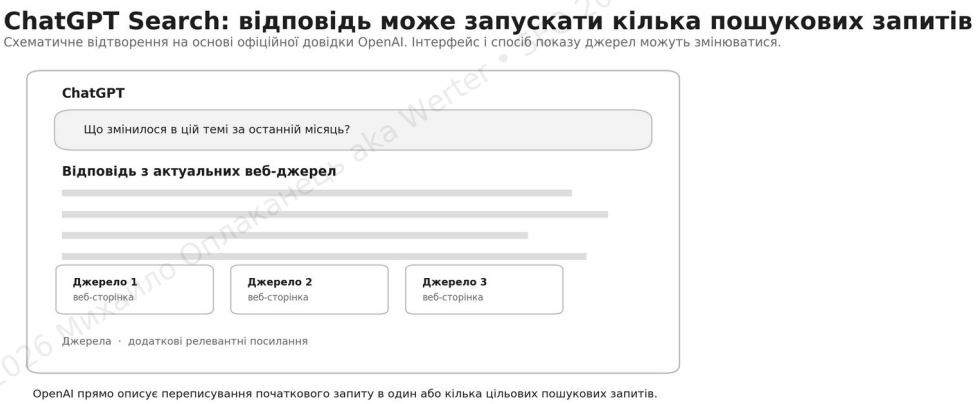


Рис. 2.4. ChatGPT Search: веб-пошук є частиною формування відповіді, а не окремою сторінкою результатів.

Схематичне відтворення на основі офіційної довідки OpenAI: https://help.openai.com/en/articles/9237897

Для власника сайту OpenAI окремо розділяє пошукове виявлення і навчання моделей. Щоб контент міг потрапляти у зведення й фрагменти ChatGPT Search, не треба блокувати OAI-SearchBot. GPTBot - інший робот, який використовується для керування потенційним використанням контенту в навчанні.[5] Це принципова технічна різниця: заборона навчання не повинна автоматично

перетворюватися на заборону пошукового виявлення, якщо бізнес хоче отримувати переходи та цитування з ChatGPT.

OpenAI також повідомляє, що переходи з ChatGPT Search автоматично містять utm_source=chatgpt.com.[5] Це одна з небагатьох ділянок AI-пошуку, де джерело переходу можна визначити без моделі або стороннього сервісу: трафік із ChatGPT реально відокремлюється в аналітиці. Але це вимірює тільки переходи. Бренд може бути згаданий або процитований без кліку, і GA4 цього не покаже.

Рис. 2.5. Початковий запит користувача може породити кілька пошукових підзапитів.

Авторська схема. Основа: Google Search Central описує розгортання запиту; OpenAI Help Center описує переписування питання в один або кілька цільових запитів.[1][4]

2.4. Perplexity і Copilot: відповідь стоїть попереду списку посилань

Perplexity від початку позиціонує себе як пошукову систему з генеративною відповіддю. В офіційній довідці сервіс описує процес досить просто: інтерпретація питання, пошук в інтернеті в реальному часі, узагальнення знайденої інформації та посилання на джерела.[6] Для SEO важливе не маркетингове формулювання, а факт: сторінка конкурує не лише за місце в списку результатів, а за включення в набір джерел, з яких формується відповідь.

У липні 2026 року Perplexity окремо оновив документацію щодо сканування. PerplexityBot використовується для індексації сторінок і

появи сайтів у результатах Perplexity. Окремий Perplexity-User використовується для запитів, ініційованих користувачем. У документації Perplexity зазначає, що PerplexityBot дотримується robots.txt; для бота, який виконує запит користувача, діє інша логіка. [7] Це ще один приклад того, чому «дозволити всіх AI-ботів» або «заблокувати всіх AI-ботів» - поганий технічний підхід.

У Microsoft картина ще одна. Bing Webmaster Tools з лютого 2026 року показує AI Performance для Microsoft Copilot, генеративних підсумків Bing та окремих партнерських інтеграцій. Звіт містить загальну кількість цитувань, середню кількість цитованих сторінок, цитування на рівні URL, динаміку та вибірку запитів для пошуку джерел - пошукових фраз, через які AI отримував контент, використаний у відповіді.[8] Це дуже цінно саме тому, що Bing показує частину проміжного шару між запитом і цитуванням.

Таблиця 2.2. П'ять форматів можуть відповідати на одну потребу, але дають SEO різні сигнали й різну прозорість.[1][4][6][8]

Платформа / формат	Як формується результат	Що бачить SEO напряму	Що не можна припускати
Google AI Overview	Генеративна відповідь усередині Search; може використовувати розгортання запиту	Покази у звітах Google; видимі джерела в самій відповіді	Що топ-3 органічної видачі автоматично стане третім або кращим джерелом
Google AI Mode	Діалогова відповідь з пошуковими джерелами та уточненнями	Посилання в відповіді; генеративна видимість у Google	Що джерела збігаються з AI Overview
ChatGPT Search	Може переписувати питання в кілька пошукових запитів і шукати автоматично	Цитати / Sources, переходи з utm_source=chatgpt.com	Що початковий текст користувача дорівнює реальному пошуковому запиту
Perplexity	Пошук у вебі + генеративне узагальнення + джерела	Цитовані джерела; логи ботів	Що логіка відбору збігається з Google
Microsoft Copilot / Bing AI	Генеративні відповіді на базі пошуку і пошук джерел	Bing AI Performance: цитування, цитовані сторінки, запити для пошуку джерел	Що кількість цитувань дорівнює позиції або авторитету

2.5. Органічне ранжування і AI-цитування перетинаються, але не є одним і тим самим

Теза «щоб цитуватися в AI, достатньо бути в топ-10» звучить привабливо, бо переносить стару модель SEO в новий інтерфейс. Дані не дають підстав так спрощувати. У березні 2026 року Ahrefs оновив дослідження AI Overview на 863 тис. SERP і приблизно 4 млн URL, показаних у AI Overview. За оновленою методологією 38% сторінок, процитованих у AI Overview, одночасно ранжувалися в топ-10 органічної видачі для того самого запиту.[10]

Це сильний перетин, але не тотожність. Якщо 38% цитованих сторінок є у топ-10, більшість цитованих URL у цій конкретній методології знаходилась поза топ-10 або в інших блоках. Ще важливіше: попередня версія дослідження Ahrefs 2025 року давала значно вищу оцінку перетину. Сам Ahrefs пояснив оновлення зміною парсингу й методології.[10] Для мене це корисний урок не про точну цифру, а про дисципліну роботи з дослідженнями: AI Search змінюється настільки швидко, що навіть відомий відсоток треба зберігати разом із датою, вибіркою і способом підрахунку.

Окреме дослідження Ahrefs на 15 тис. запитів для ChatGPT, Gemini, Copilot і Perplexity показало ще слабший зв’язок із топ-10 Google. У середньому лише близько 12% URL, процитованих ChatGPT, Gemini та Copilot, були присутні в топ-10 Google для початкового запиту; Perplexity був ближче до класичної видачі: майже третина цитованих URL збігалася з топ-10 Google.[11] Це не пряме порівняння з дослідженням AI Overview: інший продукт, інша вибірка, інший метод. Але напрям однаковий: шар цитувань не є копією шару органічного ранжування.

МІФ

Міф: «Якщо сторінка в топ-3 Google, вона автоматично стане джерелом AI-відповіді». Факт: сильний органічний ранжування може збільшувати шанс бути серед кандидатів, особливо в Google AI Search, але публічні дані не підтверджують правила 1:1. Потрапляння у цитування залежить від окремого відбору джерел після інтерпретації й розгортання запиту.

2.6. Нестабільність джерел: один скріншот нічого не доводить

Класичне відстеження позицій привчив SEO до відносно стабільної одиниці спостереження: запит, позиція, дата, країна, пристрій. Генеративна відповідь додає ще один рівень випадковості. Ahrefs у листопаді 2025 року проаналізував понад 43 тис. запитів, для кожного з яких було щонайменше 16 записаних AI Overviews протягом місяця. Між сусідніми спостереженнями текст відповіді змінювався приблизно у 70% випадків, а середня тривалість одного стану відповіді становила 2,15 дня.[12]

Ще показовіші цитування: коли AI Overview оновлювався, в середньому 45,5% цитованих URL змінювалися. При цьому зміст залишався значно стабільнішим: середня семантична схожість між послідовними відповідями становила 0,95 з 1.[12] Це майже та сама картина, яку ми вже бачили при порівнянні AI Overview з AI Mode: відповідь може залишатися змістово близькою, а джерела - обертатися.

Рис. 2.6. AI Overview може зберігати змістовий консенсус, але міняти конкретні джерела.

Дані: Ahrefs, понад 43 тис. запитів, щонайменше 16 спостережень на запит протягом місяця. https://ahrefs.com/blog/ai-overview-change/

Це знищує практику, яку я бачу в багатьох «GEO-аудитах»: людина вводить десять запитів один раз, робить скріншоти і рахує частку видимості. Така цифра може бути корисною як моментальний зріз, але її не можна видавати за стабільну частку видимості. Якщо майже половина цитованих URL здатна змінитися між послідовними відповідями, потрібні повторні вимірювання й агрегація.

2.6.1. Яку одиницю вимірювання я використовую

Для AI-видимості я зберігаю не просто запит і відповідь. Мінімальна одиниця спостереження має включати точний текст запиту, платформу, країну, мову, дату, тип сеансу, наявність попереднього контексту, список цитованих URL, згадані бренди й сам факт переходу або відсутності переходу. Якщо платформа персоналізує відповідь, це теж треба фіксувати.

Рис. 2.7. Мінімальна одиниця спостереження для AI-видимості.

Авторська схема. Значення в рядках показують структуру запису, а не реальні результати тесту.

Мій робочий стандарт для дослідження, яке впливатиме на бюджет або контент-план: один запуск не вважаю достатнім. Якщо дозволяє вартість збору, роблю кілька повторів у різні дні та аналізую частоту появи, а не лише факт появи. Це не «правило Google», а методологічна відповідь на високу волатильність генеративної видачі.

2.7. Чому запит користувача вже не можна плутати з пошуковим запитом системи

У класичному SEO ми звикли будувати семантику з того, що користувач набирає в пошуковому полі. У генеративному середовищі між фразою користувача і пошуком документів з’являється додатковий шар. Google називає цей механізм розгортанням запиту (query fan-out): система може виконувати кілька пов’язаних пошуків за підтемами й джерелами даних.[1] OpenAI описує схожу за функцією поведінку: ChatGPT Search переписує питання в один або

кілька цільових запитів і після перших результатів може запускати додаткові уточнення.[4]

Це не означає, що збір і аналіз ключових слів став непотрібним. Навпаки, нам потрібні ключові слова, щоб бачити реальний попит, частотність, SERP, конкуренцію і мову користувача. Але ключове слово перестає бути єдиною одиницею, під яку система шукає документ. Для складного питання треба мислити простором підзадач: які факти потрібно знайти, які критерії порівняти, які обмеження перевірити, які сутності зв'язати між собою.

Практичний приклад. Запит «який ноутбук краще взяти для постійних перельотів, до 1200 доларів, щоб працювати з великими таблицями і не носити зарядку весь день» може вимагати окремих перевірок ваги, автономності, ціни, продуктивності процесора, обсягу пам'яті, доступності конкретної моделі й реальних тестів батареї. Сторінка не повинна буквально містити весь довгий запит. Вона повинна бути сильною відповіддю на одну або кілька інформаційних підзадач, які реально виникають у такому сценарії.

2.8. Контроль доступу: пошуковий бот, бот для навчання і бот для запитів користувача - не одне й те саме

Технічна частина AI-пошук часто починається з robots.txt, і тут дуже легко зробити грубу помилку. У різних платформ є різні роботи для різних задач. OpenAI відділяє OAI-SearchBot, потрібний для пошукового виявлення і появи контенту в ChatGPT Search, від GPTBot, який використовується для керування потенційним використанням контенту в навчанні.[5] Perplexity відділяє PerplexityBot для побудови пошукового індексу від Perplexity-User для запитів, ініційованих користувачем.[7]

Google також відділяє пошук від інших способів використання контенту. Для AI Overviews і AI Mode базовим механізмом керування обходом залишається Googlebot. Якщо власник хоче обмежити текст, що може показуватися в пошуку, Google рекомендує стандартні

налаштування фрагментів: nosnippet, data-nosnippet, max-snippet або noindex. Google-Extended стосується окремих інших AI-систем Google, а не є окремим індексаційним перемикачем для AI Overview чи AI Mode.[1]

Таблиця 2.3. Керування доступом треба робити за конкретною функцією робота, а не за ярликом «AI-бот».[1][5][7]

Система	Робот / контроль	Що він контролює	Практична помилка
Google Search AI	Googlebot + налаштування фрагментів у Search	Сканування, індексація та обсяг фрагментів у Search	Блокувати Search через страх перед навчанням моделей
ChatGPT Search	OAI-SearchBot	Виявлення контенту для результатів і зведень ChatGPT Search	Плутати OAI-SearchBot з GPTBot
OpenAI навчання моделей	GPTBot	Потенційне використання веб-контенту для навчання	Вважати його обов’язковим для появи в Search
Perplexity	PerplexityBot	Індексація для результатів Perplexity	Блокувати робота, а потім вимагати стабільних цитування
Perplexity запит користувача	Perplexity-User	Запит сторінки у відповідь на дію користувача	Вважати його звичайним пошуковим бот

ПОПЕРЕДЖЕННЯ

Масове блокування всіх ботів із «AI» у User-Agent може одночасно відрізати пошукову видимість, яку бізнес насправді хоче зберегти. Перед будь-якою правилом WAF треба відповісти на три питання: хто сканує, для якої функції, і що саме бізнес хоче дозволити або заборонити.

2.9. Що реально можна оптимізувати, а що ми не контролюємо

Після всього цього легко перейти в протилежну крайність: якщо джерела нестабільні й системи різні, начебто оптимізувати нічого не

можна. Це теж неправильно. Ми не контролюємо конкретний список URL у відповіді, але контролюємо велику частину вхідних умов, за якими сторінка взагалі може бути кандидатом.

Для Google фундамент залишається класичним: сторінка має бути доступна Googlebot, проіндексована, придатна до показу з фрагментом, пов'язана внутрішніми посиланнями, містити важливу інформацію у текстовій формі й відповідати правилам пошуку Google.[1] Для ChatGPT треба не блокувати OAI-SearchBot, якщо ми хочемо появи у зведеннях і цитуваннях.[5] Для Perplexity треба дозволити PerplexityBot, якщо видимість у сервісі є бізнес-ціллю.[7]

Далі починається не «секретна GEO-оптимізація», а конкуренція за корисність конкретного документа. Чіткі факти, власні дані, первинні джерела, зрозумілі таблиці, актуальність, однозначні назви сутностей, реальні характеристики продукту, ціни, умови, дати, методологія дослідження - усе це дає системі інформацію, яку можна точно витягти й використати. Це не гарантує цитування. Але це створює набагато сильніший документ-кандидат, ніж загальний текст, який нічого не додає до наявного масиву вебдокументів.

Таблиця 2.4. У генеративному пошуку є великий контроль над вхідними умовами, але немає прямого контролю над кінцевим цитуванням.

Шар	Що контролюємо	Що не контролюємо
Доступ	robots.txt, WAF, статус-коди, внутрішні посилання	Частоту повторного сканування конкретною платформою
Індекс / виявлення	канонікалізацію, noindex, якість і стабільність URL	Гарантію включення в індекс сторонньої системи
Зміст	факти, структуру, власні дані, оновлення, сутності	Точний підзапит, який система згенерує
Відбір джерел	релевантність документа до інформаційної підзадачі	Конкретну вагу сигналів і порядок цитування
Відображення	налаштування фрагментів там, де платформа їх підтримує	Формат генеративної відповіді та інтерфейс

Шар	Що контролюємо	Що не контролюємо
Бізнес-результат	посадкова сторінка, пропозицію, аналітику, конверсію	Чи клікне користувач після отримання відповіді без переходу

2.10. Як я будую міжплатформний тест без самообману

Якщо задача - зрозуміти, де бренд або сайт реально присутній у генеративному пошуку, я не починаю з випадкового списку запитів, придуманого SEO-спеціалістом. Я починаю з реального простору попиту: GSC, внутрішнього пошуку сайту, даних продажів, запитів служби підтримки, People Also Ask, форумів, комерційних модифікаторів і питань, які реально виникають перед конверсією. Потім із цього формую контрольний набір задач користувача.

Другий принцип - не змішувати наміри. Інформаційне питання «як працює X», порівняння «X проти Y», рекомендація «що вибрати», локальна задача «де знайти», транзакційна задача «купити/забронювати» і навігаційна задача «офіційний сайт бренду» повинні аналізуватися окремо. Інакше середня частка цитувань просто змішає зовсім різну поведінку систем.

Третій принцип - однакове середовище. Для кожного запуску я фіксую мову, країну, стан авторизації, чистий або продовжений діалог і дату. Для платформ із персоналізацією бажано мати окремий «чистий» профіль. Якщо система використовує історію діалогу, першу і четверту репліку не можна порівнювати як однаковий запит.

Четвертий принцип - зберігати не тільки факт згадки бренду. Я окремо записую: чи є бренд у тексті; чи є URL бренду серед джерел; який саме URL; скільки усього джерел у відповіді; чи є конкурент; чи є посилання клікабельним; який тип сторінки процитовано; чи це головна, категорія, стаття, відео, форум, профіль або стороння згадка.

П’ятий принцип - повторювати. Через нестабільність джерел одна відповідь має низьку доказову цінність. Для великих наборів запитів

я радше зменшу кількість платформ або частоту збору, ніж залишу по одному неповтореному спостереженню і назву його стабільною часткою видимості.

Таблиця 2.5. Мінімальна структура даних для відтворюваного відстеження AI-видимості.

Поле в датасеті	Навіщо потрібне
query_id / task_id	Стабільний ідентифікатор задачі, незалежний від тексту prompt
exact_query	Точний текст, який ввів користувач або тест
intent / funnel stage	Щоб не змішувати інформаційні й комерційні сценарії
platform / surface	AI Overview, AI Mode, ChatGPT, Perplexity, Copilot тощо
country / language	Локаль та мова можуть змінити джерела
session_state	Чистий сеанс, авторизований профіль, уточнювальний запит
captured_at	Відповіді й цитування змінюються з часом
brand_mentioned	Факт присутності бренду в тексті
cited_urls	Конкретні URL, а не лише домени
competitor_mentions	Порівняльна видимість
answer_hash / snapshot	Дає змогу бачити, чи змінилася сама відповідь
referral_clicks	Тільки там, де трафік реально можна атрибутувати

2.11. Як читати метрики, щоб не створити новий DR

AI-пошук уже породжує десятки власних показників: оцінку видимості, частку цитувань, частку відповідей, частку згадок, частку джерел, охоплення запитів. Проблема не в самих метриках. Проблема починається, коли похідний показник стороннього сервісу сприймають як прямий сигнал платформи. Це та сама помилка, яку SEO роками робило з DR або DA.

Будь-яка власна AI-метрика має починатися з чіткого знаменника. «Частка цитувань 22%» нічого не означає без відповіді: 22% від чого? Від усіх відповідей? Від відповідей, де взагалі були джерела? Від усіх цитованих URL? Від трьох найпомітніших цитувань? За один запуск чи за середнє з повторів? Який набір запитів? Яка країна? Яка платформа?

Для керування командою я використовую кілька простих показників. Охоплення: у скількох контрольних задачах бренд або сайт з'явився хоча б раз за період. Частота цитувань: у скількох спостереженнях був процитований конкретний домен або URL. Концентрація джерел: яка частка всіх цитувань припадає на кілька сторінок. Розрив із конкурентами: у яких групах задач конкурент стабільно присутній, а ми ні. Нестабільність: як часто склад джерел змінюється між повторними спостереженнями. Це не стандарти Google чи OpenAI. Це робочі коефіцієнти для власної аналітики.

2.12. 90-хвилинний практичний аудит: чи однаково вас бачать різні AI-пошуки

Цей аудит не замінює постійне відстеження. Його задача - за півтори години зрозуміти, чи проблема взагалі існує і де її шукати.

Таблиця 2.6. Швидкий міжплатформний аудит. Це діагностика, а не доказ причинності.

Час	Що робимо	Що фіксуємо
0-15 хв	Беремо 10-15 реальних задач із GSC, продажів і служби підтримки; ділимо за наміром	query_id, намір, країна, мова
15-35 хв	Перевіряємо Google AI Overview / AI Mode там, де вони доступні	чи є AI-відповідь, бренд, цитований URL, конкурент
35-55 хв	Ті самі задачі запускаємо в ChatGPT Search і Perplexity	точний список цитований URL і типи сторінок
55-70 хв	Порівнюємо URL-джерела між платформами	перетин доменів, перетин URL, сторінки-конкуренти

Час	Що робимо	Що фіксуємо
70-80 хв	Перевіряємо доступ ботів і логи	Googlebot, OAI-SearchBot, PerplexityBot, блокування WAF
80-90 хв	Формуємо 3-5 гіпотез, які можна перевірити даними	технічна проблема, прогалина в контенті, слабка сторінка-джерело, відсутність підтвердження сутності

Якщо результат показує, що сайт добре ранжується в Google, але майже не з'являється в ChatGPT і Perplexity, я не починаю з переписування всього контенту «під LLM». Спочатку перевіряю, чи взагалі доступні потрібні сторінки відповідним роботам. Потім дивлюся, які типи документів цитують конкуренти: первинні джерела, огляди, категорії, довідники, форуми, відео. Лише після цього ставлю контентну гіпотезу.

Якщо навпаки: сайт часто цитують у AI, але переходів майже немає, проблема може бути не в SEO. Можливо, сама відповідь закриває інформаційну потребу без кліку. Тоді треба оцінювати, чи є сенс зміщувати контент у бік задач, де користувачеві все одно потрібен сайт: калькулятор, персональна пропозиція, актуальна база даних, інструмент, бронювання, оформлення замовлення, детальне порівняння, власне дослідження. Саме тут видимість знову перетворюється на дію.

2.13. Що ми точно знаємо, а що лишається гіпотезою

Ми точно знаємо з офіційної документації, що Google AI Overview та AI Mode можуть використовувати розгортання запиту і можуть показувати різні відповіді та джерела.[1] Ми знаємо, що для появи як посилання-джерела в Google AI Search сторінка має бути проіндексована й придатна для звичайного показу з фрагментом.[1] Ми знаємо, що ChatGPT Search може переписувати питання в один або кілька пошукових запитів і додавати уточнювальні пошуки.[4] Ми знаємо, що OAI-SearchBot і GPTBot мають різні функції.[5] Ми знаємо, що Perplexity має окремого пошукового бота і бота для

запитів користувача.[7] Ми знаємо, що Bing уже показує цитування та запити для пошуку джерел у Webmaster Tools.[8]

Ми не знаємо з публічних джерел точних ваг сигналів відбору URL у Google AI Mode, ChatGPT Search, Perplexity або Copilot. Ми не знаємо універсального «фактора цитування». Ми не знаємо формули, яка перетворює органічну позицію в імовірність цитування. Ми не можемо з одного скріншота визначити стабільну частку видимості. І ми не маємо підстав називати будь-яку конкретну довжину абзацу, кількість списків або розмітку schema.org універсальним способом «потрапити в LLM».

Саме тому сильна стратегія у 2026 році виглядає менш магічно, ніж маркетинг навколо GEO. Доступ до документа, зрозуміла архітектура, сильна сторінка-джерело, точні факти, власні дані, стабільні сутності, актуальність, зовнішні підтвердження, повторне вимірювання. Новий інтерфейс додає нові точки відбору й нові метрики, але не скасовує необхідність бути найкращим доступним джерелом для конкретної інформаційної задачі.

НОТАТКА З ПРАКТИКИ

Якщо після AI-аудиту ви не можете показати конкретний URL конкурента, конкретний запит, конкретну платформу, дату спостереження і причину, чому цей документ сильніший як джерело, аудит поки не дає задачі для команди. «Покращити GEO» - не задача. «Додати на /pricing/ актуальну таблицю тарифів із датою оновлення, бо саме такі сторінки конкурента цитуються у 14 із 30 контрольних запитів» - уже задача.

Джерела до глави

1. Google Search Central. AI features and your website.
https://developers.google.com/search/docs/appearance/ai-features Офіційна документація Google про AI Overviews, AI Mode, розгортання запиту, технічну придатність та налаштування фрагментів.
2. Google. A new era for AI Search. I/O 2026.
https://blog.google/products-and-platforms/products/search/search-io-2026/ Оновлення AI Mode, безшовний перехід з AI Overview, агенти та новий Search box.
3. Google. New opportunities, control and insights for website owners.
https://blog.google/products-and-platforms/products/search/new-controls-website-owners/ Опубліковано 3 червня 2026, оновлено 31 серпня 2026; аудиторія AI Overviews і AI Mode.

4. OpenAI Help Center. Searching the web with ChatGPT. https://help.openai.com/en/articles/9237897 Опис веб-пошуку, citations і переписування запиту в один або кілька цільових пошуків.
5. OpenAI Help Center. Publishers and Developers - FAQ. https://help.openai.com/en/articles/12627856 OAI-SearchBot, GPTBot, noindex і utm_source=chatgpt.com.
6. Perplexity Help Center. How does Perplexity work?. https://www.perplexity.ai/help-center/en/articles/10352895-how-does-perplexity-work Офіційний опис real-time web search, узагальнення й citations.
7. Perplexity. Crawlers. https://docs.perplexity.ai/docs/resources/perplexity-crawlers PerplexityBot, Perplexity-User, robots.txt та IP ranges.
8. Microsoft Bing. Introducing AI Performance in Bing Webmaster Tools. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview Total Citations, Average Cited Pages, grounding queries та page-level citation activity.
9. Ahrefs. Are AI Mode and AI Overviews Just Different Versions of the Same Answer?. https://ahrefs.com/blog/ai-overviews-vs-ai-mode/ 730 тис. пар відповідей; citation overlap, semantic similarity та entity overlap.
10. Ahrefs. Update: 38% of AI Overview Citations Pull From the Top 10. https://ahrefs.com/blog/ai-overview-citations-top-10/ 863 тис. SERP і близько 4 млн AI Overview URLs, березень 2026.
11. Ahrefs. Only 12% of AI Cited URLs Rank in Google's Top 10 for the Original Prompt. https://ahrefs.com/blog/ai-search-overlap/ 15 тис. запитів; ChatGPT, Gemini, Copilot і Perplexity, серпень 2025.
12. Ahrefs. AI Overviews Change Every 2 Days. https://ahrefs.com/blog/ai-overview-change/ Понад 43 тис. запитів; волатильність відповідей, citations та entities, листопад 2025.

ГЛАВА 3

Zero-click і нова економіка органічного кліку

Чому висока позиція вже не гарантує переходу, як відрізнити падіння попиту від стискання CTR і які метрики мають сенс, коли частина пошуку закінчується прямо у видачі

Стан даних і джерел: 12 вересня 2026 року

Клік став дефіцитнішим ресурсом. Це не означає, що органічний пошук перестав працювати. Це означає, що стара логіка «вищі позиції -> більше кліків -> більше бізнес-результату» стала неповною. У 2026 році між позицією URL і переходом стоїть набагато більше шарів: AI Overview, відповіді без переходу, локальні блоки, товарні блоки, відео, панелі знань, уточнювальні запити, повторний пошук і дедалі більша кількість задач, які користувач закриває ще до відкриття сайту.

Для SEO це неприємна зміна не через сам факт меншої кількості кліків. Неприємне інше: падіння CTR легко переплутати з падінням позицій, проблемою контенту або оновленням Google. Якщо діагноз неправильний, команда починає переписувати сторінки, нарощувати посилання або міняти шаблони там, де реальна причина знаходиться в SERP. Тому zero-click я розглядаю не як маркетинговий термін, а як окремий шар вимірювання між видимістю і сесією.

3.1. Zero-click почався задовго до AI Overviews

Пошук без кліку не є винаходом генеративного AI. Google роками закривав частину інформаційних задач без переходу на сайт: прогноз погоди, курс валют, калькулятори, спортивні результати, час,

визначення, локальні блоки, панелі знань, виділені фрагменти. AI Overview різко розширив клас питань, які можна закрити таким способом, але сам механізм існував давно.

Саме тому я не використовую формулу «AI забрав X% органічного трафіку», якщо немає контрольованого дизайну дослідження. У більшості реальних даних одночасно змінюються тип запиту, склад SERP, частота AI Overview, реклама, намір запиту, пристрій і поведінка користувачів. Ми можемо виміряти асоціацію і напрямок. Значно важче довести, яка частина втрати спричинена саме одним елементом.

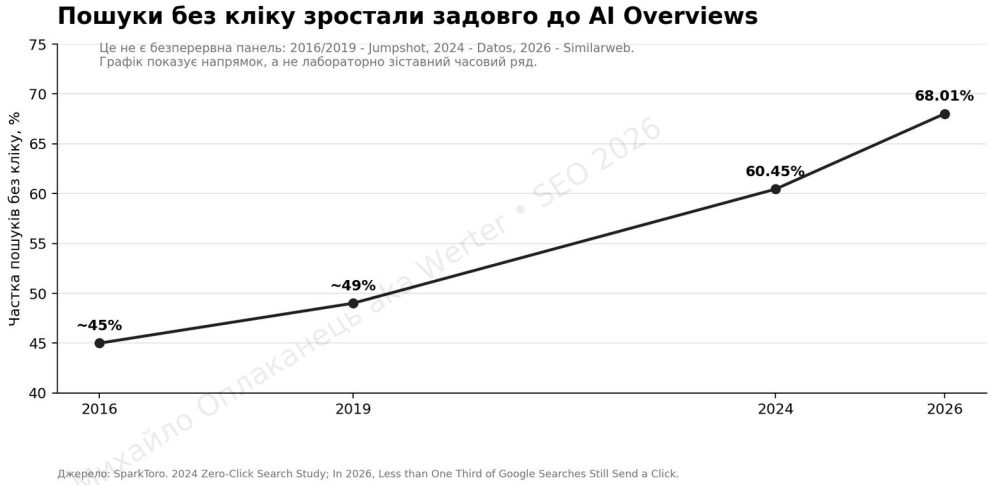


Рис. 3.1. Частка пошуків без кліку зростала ще до масового запуску AI Overviews. Різні роки спираються на різні панелі поведінкових даних, тому графік не можна читати як ідеально зіставний часовий ряд.

ДАНІ / У США SparkToro оцінював частку пошуків Google без кліку приблизно у 60.45% у 2024 році та 68.01% у січні-квітні 2026 року. Це +7.56 процентного пункту за два роки. Але постачальник панелі змінився: 2024 рік базувався на Datos, 2026 рік - на Similarweb. Порівняння корисне як напрямок, а не як лабораторний експеримент.[1][3]

3.2. Спочатку треба домовитися, що саме називаємо «пошуком без кліку»

У SEO zero-click часто вживають так, ніби це одна універсальна метрика. Насправді під цим словом можуть ховатися щонайменше п’ять різних подій. Якщо їх змішати, два дослідження з однаковою цифрою можуть вимірювати різні речі.

Подія	Що сталося	Чому це не те саме
Пошук без будь-якого кліку	Користувач не відкрив ні органічний результат, ні рекламу, ні ресурс Google	Найсуворіше визначення. Саме так SparkToro описує zero-click у дослідженні 2026 року.
Немає кліку до відкритого вебу	Клік міг піти у Maps, YouTube або інший ресурс Alphabet	Для видавця це теж втрата зовнішнього переходу, але формально клік відбувся.
Немає органічного кліку	Користувач міг натиснути рекламу або товарний блок	Погано для органічного трафіку, але не є zero-click у буквальному сенсі.
AI-відповідь без кліку	Користувач прочитав відповідь або побачив бренд, але не відкрив джерело	Є видимість і потенційний вплив, але немає прямої сесії.
Відкладений перехід	Після відповіді користувач пізніше вводить бренд, заходить напряму або конвертується через інший канал	В аналітиці останнього кліку вплив пошуку може зникнути.

Таблиця 3.1. «Zero-click» треба визначати до того, як команда почне порівнювати цифри між джерелами.

SparkToro у 2026 році визначав zero-click як пошук, після якого не було жодного кліку по результатах. Сюди входять два сценарії: користувач завершує сесію або запускає новий пошук через стандартне поле Google. Клік по YouTube, Maps, рекламі або органічному URL уже виводить пошук із zero-click категорії.[1]

Pew Research Center використовував іншу поведінкову схему. Для кожної відвіданої сторінки Google вони дивилися наступну URL-подію: клік по звичайному результату, клік по AI-підсумку, новий пошук, перехід кудись іще або завершення браузерної сесії. Тому їхні

8%, 15% чи 26% не можна напряму підставляти в ту саму формулу, що й 68.01% SparkToro.[4]

МІФ / «68% zero-click означає, що Google забрав 68% ваших органічних кліків». Ні. 68% - макропоказник поведінки всіх пошуків у конкретній панелі поведінкових даних. Він не говорить, яким був CTR вашого сайту, скільки цих запитів були релевантні вашому бізнесу і скільки кліків ви мали б отримати в альтернативному SERP.

3.3. Що реально відбувається після Google-пошуку у 2026 році

Для США у січні-квітні 2026 року SparkToro та Similarweb отримали доволі наочну картину. Приблизно 32% пошуків завершилися хоча б одним кліком. Близько 39% завершили саму сесію, ще близько 29% привели до нового пошуку. Разом ці дві останні групи дають близько 68% пошуків без кліку.[1]

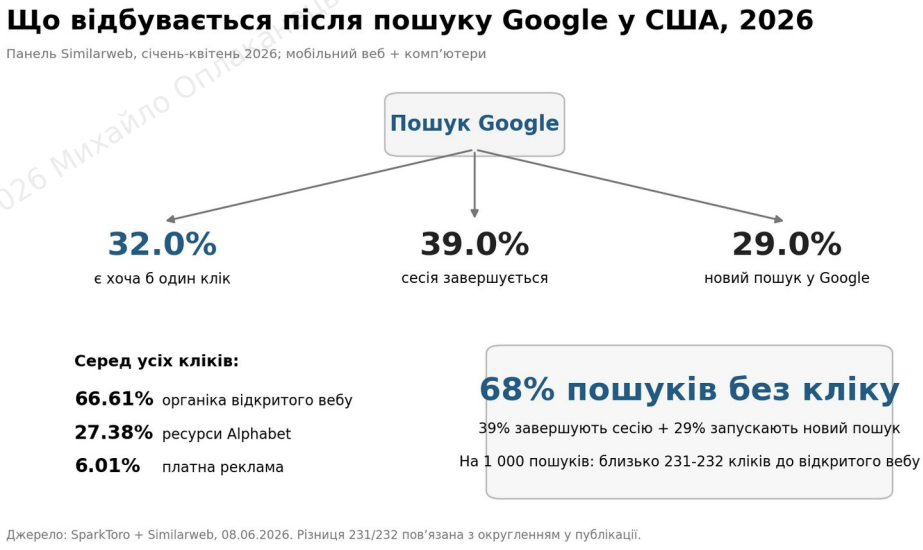


Рис. 3.2. Розподіл дій після Google-пошуку у США. Усередині групи кліків SparkToro окремо показує відкритий веб, ресурси Alphabet і рекламу.

Тут є важлива деталь, яку зазвичай пропускають у заголовках. «Є клік» не означає «є органічний клік на зовнішній сайт». За даними

SparkToro, серед кліків у цій панелі 66.61% припадали на органічні результати відкритого вебу, 27.38% - на ресурси Alphabet, а 6.01% - на платну рекламу. У підсумку на 1 000 Google-пошуків у США дослідження оцінює близько 231-232 кліків до відкритого вебу залежно від округлення в окремих таблицях і візуалізаціях.[1][2]

Для SEO Lead це практично корисніше, ніж сама цифра 68%. Макроринок органічного кліку треба мислити не як «1000 пошуків = 1000 можливостей», а як послідовність втрат між попитом і зовнішнім переходом. Частина запитів закривається одразу. Частина веде у власні продукти Google. Частина монетизується рекламою. І тільки після цього починається конкуренція між органічними URL.

3.4. Zero-click залежить від країни, інтерфейсу і поведінки аудиторії

Ще одна причина не використовувати одну глобальну норму CTR - помітна різниця між країнами. У червні 2026 року SparkToro опублікував зіставний зріз Similarweb для шести ринків за той самий період. Частка пошуків без кліку коливалась від 62.1% у Німеччині до 69.5% у Великій Британії.[2]

Рис. 3.3. Частка Google-пошуків без кліку в шести країнах за даними однієї панелі Similarweb.

Країна	Zero-click	Новий пошук	Сесія завершилась	Кліки до відкритого вебу / 1000
Німеччина	62.1%	22.5%	39.5%	287
Італія	63.4%	24.7%	38.8%	280
Канада	63.8%	25.6%	38.2%	268
Франція	65.3%	22.9%	42.3%	271
США	68.0%	29.0%	39.0%	231
Велика Британія	69.5%	26.4%	43.1%	232

Таблиця 3.2. Один і той самий Google дає різну економіку кліку залежно від ринку. Джерело: SparkToro + Similarweb, січень-квітень 2026.

Різниця між Німеччиною та Великою Британією становить 7.4 процентного пункту zero-click. Ще важливіше, у Німеччині дослідження нараховує 287 кліків до відкритого вебу на 1 000 пошуків, а у США - 231. Це приблизно на 24% більше кліків до відкритого вебу на однаковий обсяг пошуків. Причину з цих даних довести не можна. SparkToro припускає вплив регуляції, культури, мови та відмінностей інтерфейсу, але прямо маркує це як можливі пояснення, а не встановлену причинність.[2]

НОТАТКА З ПРАКТИКИ / Коли я бачу орієнтир CTR без країни й пристрою, для мене він майже не має операційної цінності. Ринок, мобільні пристрої / комп'ютери і тип SERP можуть змінити клік-потенціал сильніше, ніж різниця між двома сусідніми позиціями.

3.5. AI Overview справді асоціюється з нижчим CTR, але цифри треба читати правильно

Найсильніші незалежні дані тут дають два різні типи досліджень. Pew спостерігав реальну поведінку користувачів. Ahrefs порівнював CTR великої вибірки ключових слів з AI Overview та без нього. Обидва підходи показують той самий напрямок, але відповідають на різні питання.

Pew Research Center проаналізував браузерну активність 900 дорослих у США. У вибірці було 68 879 унікальних Google-пошуків у березні 2025 року; 12 593 з них мали AI-підсумок у результатах, коли Pew повторно збирав SERP у квітні. На сторінках з AI-підсумком користувачі переходили по звичайному результату у 8% відвідувань. Без AI-підсумку - у 15%. По посиланню всередині самого AI-підсумку клікали приблизно в 1% відвідувань. Завершення браузерної сесії після сторінки з AI-підсумком траплялося у 26% випадків проти 16% без нього.[4]

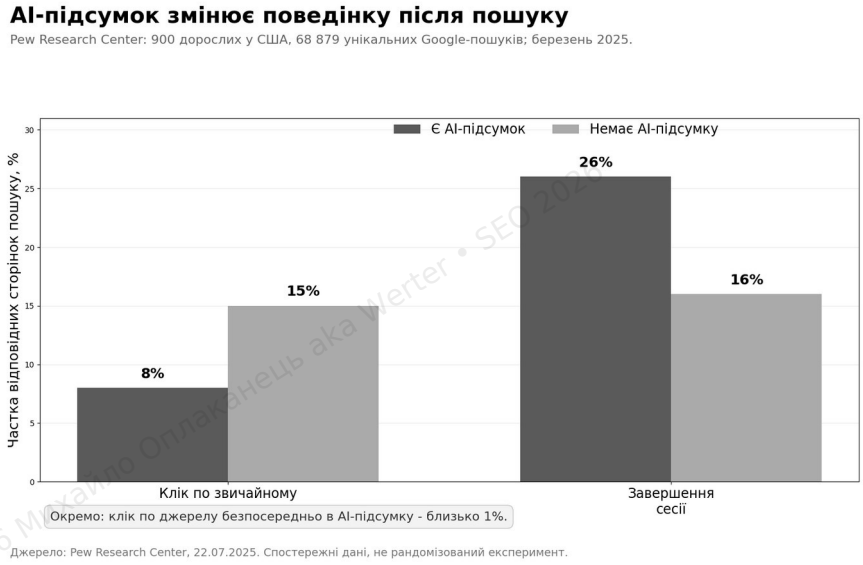


Рис. 3.4. Поведінка користувачів у дослідженні Pew. Це спостереження, а не рандомізований експеримент.

Ahrefs у лютому 2026 року повторив своє CTR-дослідження на 300 000 ключових словах: 150 000 із AI Overview та 150 000 інформаційних без нього. Вони використали агреговані дані Google Search Console і порівнювали грудень 2023 року з груднем 2025 року. Після нормалізації загального падіння CTR присутність AI Overview корелювала приблизно з 58% нижчим середнім CTR сторінки на позиції 1.[5]

Джерело	Що вимірювали	Вибірка / період	Основний результат	Головне обмеження
Pew Research Center	Наступну дію реального користувача після Google SERP	900 дорослих у США; 68 879 пошуків; березень 2025	8% кліків по звичайних результатах з AI-підсумком проти 15% без; 1% по джерелах AI-підсумку	Не експеримент; AI-підсумок міг з'являтися на іншому типі запитів.
Ahrefs	CTR позицій для ключових слів з AI Overview і без	300 000 ключових слів; GSC; грудень 2023 vs грудень 2025	Для позиції 1 оцінена різниця близько -58% CTR	Групи не рандомізовані; намір запиту і склад SERP можуть відрізнятися.
Google	Сукупний органічний обсяг кліків і «якісні кліки»	Внутрішні дані Google; серпень 2025	Google заявив про відносно стабільний обсяг органічних кліків рік до року і трохи більшу кількість якісних кліків	Немає опублікованого сирого набору даних і повної методології для незалежної перевірки.

Таблиця 3.3. Три джерела не суперечать одне одному автоматично: вони вимірюють різні рівні системи.

З цього не можна чесно зробити висновок «AI Overview забирає рівно 58% трафіку». Ahrefs вимірює середню асоціацію на конкретній вибірці, а не універсальний коефіцієнт для будь-якого запиту. Pew також не рандомізував AI-підсумок між однаковими пошуками. І це важливо: Google частіше показує AI-відповіді на довших, питальних та інформаційних запитах, які самі по собі можуть мати нижчу схильність до кліку.[4]

ПОПЕРЕДЖЕННЯ / Якщо дослідження порівнює «з AI Overview» проти «без AI Overview», перше питання - чи були групи однаковими за наміром запиту, довжиною запиту, позиціями, рекламою, пристроєм і складом SERP. Без цього ми маємо кореляцію, а не чистий причинний ефект.

3.6. Чому Google може говорити про стабільні кліки, а видавці одночасно бачити падіння

У серпні 2025 року Google публічно заявив, що загальний обсяг органічних кліків із Search на сайти був «відносно стабільним» рік до року, а кількість так званих якісних кліків трохи зросла. Під якісним кліком Google має на увазі перехід, після якого користувач не повертається одразу назад у видачу. Компанія також стверджує, що AI Overviews стимулюють більше пошуків і показують більше посилань. [6]

Це не спростовує дані Ahrefs або Pew. Агрегований ринок і CTR конкретної групи запитів - різні знаменники. Якщо кількість пошуків зростає, а CTR на один SERP зменшується, сукупний обсяг кліків теоретично може залишитися стабільним. Якщо трафік перерозподіляється між сайтами, частина видавців може падати, а інша частина - рости, навіть коли загальна кількість зовнішніх переходів майже не змінюється.

Google повідомляв, що у великих ринках AI Overviews підвищували використання Search більш ніж на 10% для типів запитів, на яких вони з'являються.[7] Це показник використання, а не зовнішніх кліків. Тому речення «користувачі шукають більше» і «окремий результат отримує менший CTR» можуть бути правдивими одночасно.

ДАНІ / Не плутайте три різні рівні: кількість пошуків у Google, кількість SERP із вашим показом і кількість переходів на ваш URL. Один рівень може рости, другий бути стабільним, а третій падати.

3.7. Нова економіка кліку: менше переходів не завжди означає менше цінності

Для бізнесу клік сам по собі ніколи не був кінцевою метрикою. Важливий результат після кліку. Zero-click робить це особливо помітним: якщо перехід стає рідшим, треба дивитися не тільки на

CTR, а й на конверсію, дохід на сесію, маржу, повторні візити й асоційовані конверсії.

Рис. 3.5. Авторська модель економіки органічного показу. Позиція впливає на один етап, але фінальна бізнес-цінність утворюється добутком кількох імовірностей.

Практично я дивлюся на органічний показ як на актив із очікуваною цінністю. У найпростішій формі: цінність одного показу = CTR × коефіцієнт конверсії × цінність однієї конверсії. Формула не є моделлю Google. Це модель бізнес-економіки. Якщо CTR падає на 30%, але конверсія переходу зростає на 50%, фінальний дохід може поводитися зовсім не так, як графік кліків.

Є перші дані, що частина переходів із AI-джерел справді має вищий намір. Adobe Digital Insights на даних роздрібної торгівлі США повідомляв, що переходи з AI-джерел у березні 2026 року конвертували на 42% краще за переходи з інших джерел, у травні - на 54% краще, у липні - на 60% краще. Це не органічний трафік Google і не доказ того, що будь-який AI-клік кращий. Але це сильний приклад того, чому обсяг і якість переходу треба вимірювати окремо.[9][10]

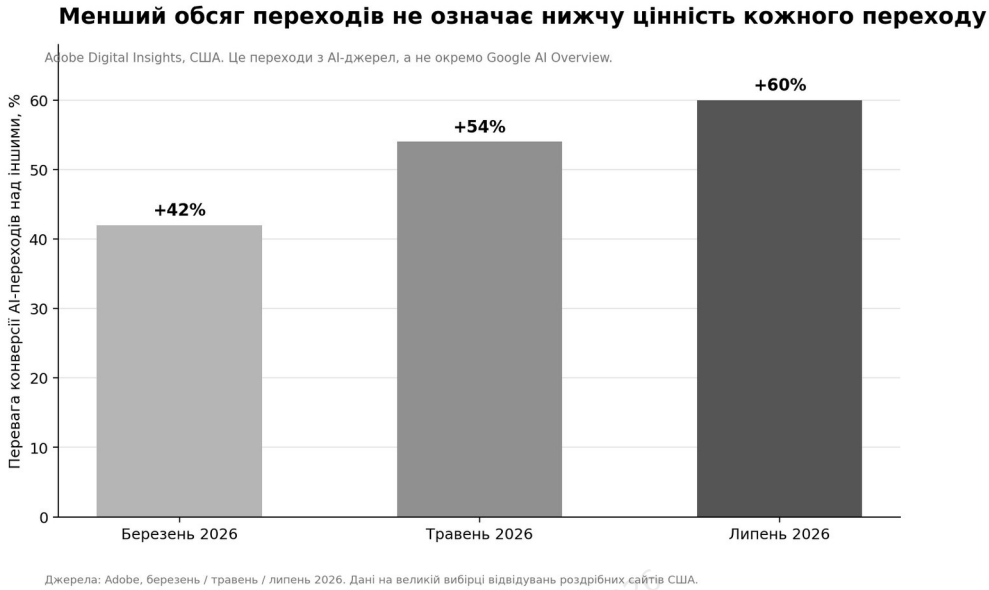


Рис. 3.6. За даними Adobe, переходи з AI-джерел на сайти роздрібної торгівлі США у 2026 році конвертували краще за інші переходи. Це окремий канал і його не можна напряму прирівнювати до Google AI Overview.

Similarweb оцінював середній обсяг переходів з AI-джерел приблизно у 770.7 млн візитів на місяць у світі за червень 2025 - травень 2026, +117.4% рік до року. При цьому для більшості сайтів переходи з AI-джерел усе ще лишалися низькою однозначною часткою загального трафіку.[8] Це хороша ілюстрація нової ситуації: канал може бути малим у абсолюті, швидко рости і водночас мати непропорційний вплив на рішення користувача.

3.8. Як відрізнити втрату позицій від стискання CTR

На реальному сайті я не починаю діагностику з питання «чи з’явився AI Overview». Починаю з базових сигналів. Кліки впали - що сталося з показами? Якщо покази теж впали, причина часто ближча до попиту, сезонності, втрати запитів або індексації. Якщо покази стабільні, дивлюся позиції. Якщо позиції погіршились, це вже задача ранжування. І тільки коли покази та позиції відносно стабільні, а CTR помітно просідає, я серйозно дивлюся на зміну SERP.

Рис. 3.7. Мінімальний порядок діагностики падіння кліків. Він не доводить причинність, але сильно зменшує ризик лікувати не ту проблему.

Сигнал у даних	Що це частіше означає	Що перевірити до змін на сторінці
Кліки ↓, покази ↓, позиція стабільна	Менший попит або зміна структури запитів	YoY, сезонність, країна, пристрій, брендові / небрендові запити, видалені елементи SERP.
Кліки ↓, покази стабільні, позиція ↓	Втрата ранжування	Конкуренти, оновлення Google, канонікалізація, індексація, зворотні посилання, зміна наміру запиту.
Кліки ↓, покази стабільні, позиція стабільна, CTR ↓	Стискання можливості отримати клік у SERP	AI Overview, реклама, локальні й товарні блоки, виділені фрагменти, зміна заголовка або снипета.
Кліки ↓, AI-покази ↑	Можлива видимість без переходу	звіт про генеративний AI у GSC, сторінки, країни, пристрої; зіставити зі звичайним звітом ефективності.
Кліки ↓, конверсії стабільні або ↑	Зміна якості трафіку	CVR, дохід на сесію, маржа, приріст брендового попиту, асоційовані конверсії.

Таблиця 3.4. Падіння кліків не є діагнозом. Це симптом, який треба розкласти на попит, ранжування, SERP і якість трафіку.

3.9. Що реально можна виміряти в Google Search Console у 2026 році

У звичайному звіті «Ефективність» Google Search Console ми маємо кліки, покази, CTR і середню позицію. Google визначає CTR просто як кліки / покази, а середню позицію - як середню найвищу позицію результату сайту або URL залежно від агрегації.[12] Цього достатньо, щоб знайти сторінки й запити, де віддача кліків падає без явної втрати видимості.

З 31 серпня 2026 року Google розгорнув для всіх сайтів окремий звіт про ефективність генеративного AI. Він показує покази в AI Overviews та AI Mode, дозволяє групувати їх за сторінками, країнами, датами й пристроями. Але окремих кліків, CTR, позиції та виміру запитів у цьому звіті немає.[11] Тому ми можемо бачити, що сторінка отримує AI-видимість, але не можемо напряму з GSC порахувати «CTR AI Overview» або визначити точний запит.

Ще одне операційне обмеження: станом на вересень 2026 року звіт про генеративний AI експортується через кнопку «Експорт», але не входить у стандартний Search Console API або масовий експорт до BigQuery. Для звичайного звіту ефективності BigQuery доступний і залишається найзручнішим способом великого порівняння CTR, позицій і показів.[11][13]

ПОПЕРЕДЖЕННЯ / Не будуйте «AI CTR» як кліки / покази генеративного AI, якщо кліки взяті із загального звіту ефективності. Чисельник і знаменник належать різним наборам подій. Така метрика виглядає переконливо, але математично не відповідає тому, що ви хочете виміряти.

3.10. SQL: знайти кандидати на «стискання кліку», а не просто сторінки з падінням трафіку

Для великого сайту я починаю з URL, де покази не впали суттєво, середня позиція не погіршилась суттєво, але CTR і кліки просіли. Це не доводить вплив AI Overview. Це просто короткий список сторінок, де є сенс вручну перевірити SERP і звіт про генеративний AI.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ

Я завантажу два CSV з Google Search Console за два сусідні 28-денні періоди на рівні сторінок. Перевір назви колонок і не вигадуй відсутні дані. Зістав URL між періодами, порахуй зміну показів, кліків, CTR і середньої позиції. Виведи URL, де в обох періодах щонайменше 1000 показів, покази змінились не більше ніж на 10%, позиція не більше ніж на 0,5, а CTR впав щонайменше на 20%. Відсортуй за найбільшим падінням кліків і поясни, що це лише кандидати на ручну перевірку SERP, а не доказ впливу AI.

АЛЬТЕРНАТИВА БЕЗ ШІ

У Google Sheets/Excel зведіть два експорти через XLOOKUP або Power Query, порахуйте відносну зміну показів, кліків і CTR та абсолютну зміну позиції, а потім застосуйте ті самі фільтри. Для повного масового експорту без ліміту рядків SQL лишається надійнішим.

QR: КОД

Скануйте камерою: QR містить компактну виконувану версію коду без коментарів. Коментований код наведено нижче.

ДЛЯ АВТОМАТИЗАЦІЇ: SQL ДЛЯ BIGQUERY

```
-- Завдання: знайти URL, де видимість майже не змінилась,
-- але CTR помітно впав. Це кандидати на ручну перевірку SERP.
-- Замініть YOUR_PROJECT.searchconsole на свій dataset масового експорту GSC.

WITH base AS (
  SELECT
    url,

    -- Ділимо 56 днів на два сусідні 28-денні періоди.
    CASE
      WHEN data_date BETWEEN DATE_SUB(CURRENT_DATE(), INTERVAL 28 DAY)
                         AND DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY)
```

```
      THEN 'current'
    WHEN data_date BETWEEN DATE_SUB(CURRENT_DATE(), INTERVAL 56 DAY)
                       AND DATE_SUB(CURRENT_DATE(), INTERVAL 29 DAY)
      THEN 'previous'
  END AS period,

  -- Агрегуємо базові метрики по URL і періоду.
  SUM(impressions) AS impressions,
  SUM(clicks) AS clicks,
  SAFE_DIVIDE(SUM(clicks), SUM(impressions)) AS ctr,

  -- У масовому експорті GSC позиція зберігається як sum_position.
  SAFE_DIVIDE(SUM(sum_position), SUM(impressions)) + 1 AS avg_position

 FROM `YOUR_PROJECT.searchconsole.searchdata_url_impression`
 WHERE search_type = 'WEB'
  AND data_date BETWEEN DATE_SUB(CURRENT_DATE(), INTERVAL 56 DAY)
                    AND DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY)
 GROUP BY url, period
),

p AS (
 SELECT
  url,

  -- Розкладаємо два періоди в окремі колонки.
  MAX(IF(period='current', impressions, NULL)) AS imp_cur,
  MAX(IF(period='previous', impressions, NULL)) AS imp_prev,
  MAX(IF(period='current', clicks, NULL)) AS clicks_cur,
  MAX(IF(period='previous', clicks, NULL)) AS clicks_prev,
  MAX(IF(period='current', ctr, NULL)) AS ctr_cur,
  MAX(IF(period='previous', ctr, NULL)) AS ctr_prev,
  MAX(IF(period='current', avg_position, NULL)) AS pos_cur,
  MAX(IF(period='previous', avg_position, NULL)) AS pos_prev
 FROM base
 GROUP BY url
)

SELECT
 *,
 -- Відносна зміна показів, кліків і CTR; позиція - абсолютна різниця.
 SAFE_DIVIDE(imp_cur - imp_prev, imp_prev) AS impressions_change,
 SAFE_DIVIDE(clicks_cur - clicks_prev, clicks_prev) AS clicks_change,
 SAFE_DIVIDE(ctr_cur - ctr_prev, ctr_prev) AS ctr_change,
 pos_cur - pos_prev AS position_change
FROM p
WHERE imp_cur >= 1000
 AND imp_prev >= 1000

 -- Покази приблизно стабільні: +/-10%.
 AND ABS(SAFE_DIVIDE(imp_cur - imp_prev, imp_prev)) <= 0.10

 -- Середня позиція змінилась не більше ніж на 0.5.
 AND ABS(pos_cur - pos_prev) <= 0.5

 -- CTR впав щонайменше на 20%.
 AND SAFE_DIVIDE(ctr_cur - ctr_prev, ctr_prev) <= -0.20
ORDER BY clicks_change ASC;
```

КОД 3.1. Запит використовує офіційну схему масового експорту Search Console. Порогові значення 10%, 0.5 позиції та -20% CTR є робочими фільтрами для первинної діагностики, а не універсальними стандартами.

У цьому SQL принципово важливі дві речі. По-перше, середня позиція для URL рахується через SUM(sum_position) / SUM(impressions) + 1, як описує Google. По-друге, дані обов'язково агрегуються через SUM, бо Google прямо попереджає, що рядки масового експорту не гарантовано консолідовані за датою, URL чи запитом.[13]

Далі я беру 20-50 URL із найбільшим падінням CTR і вже вручну дивлюся SERP, історію елементів SERP у сторонньому інструменті, AI-покази, розподіл за пристроями та структуру запитів. Сам SQL не каже «винен AI». Він прибирає тисячі сторінок, де причина очевидно інша.

3.11. Різні бізнес-моделі мають різну ціну zero-click

Для видавця zero-click часто напряму б'є по монетизації, бо бізнес заробляє на переглядах сторінок, рекламі або підписці після відвідування. Для інтернет-магазину ситуація інша: користувач може отримати коротке порівняння у пошуку Google, але для покупки все одно має перейти на сайт або маркетплейс. Для локального бізнесу частина цінної дії взагалі відбувається в Google: дзвінок, маршрут, бронювання. Для партнерського сайту цінність особливо залежить від типу сторінки: інформаційна відповідь може повністю закритися в SERP, а порівняльний або бонусний намір усе ще потребує переходу, якщо користувач хоче перевірити умови чи виконати транзакцію.

Модель	Що zero-click стискає найбільше	Що я ставлю поруч із кліками
Видавець / медіа	Перегляди сторінок і рекламні покази	Повторні користувачі, прямі заходи, розсилка, брендовий пошук, дохід на відвідувача.
Партнерський сайт	Переходи з інформаційних і частини порівняльних запитів	EPC, конверсія в перехід до рекламодавця, CTR комерційних сторінок, брендовий попит.

Модель	Що zero-click стискає найбільше	Що я ставлю поруч із кліками
Інтернет-магазин	Верх воронки й частину дослідження товару	Дохід, CVR, асоційовані конверсії, частка брендових і небрендових запитів, відвідування карток товарів.
Локальний бізнес	Відвідування сайту може не відбутися взагалі	Дзвінки, маршрути, бронювання, взаємодії з профілем, локальні конверсії.
SaaS / генерація лідів	Частину освітніх запитів	Частка демо або реєстрацій, воронка продажів, брендовий пошук, асоційовані конверсії, повторні візити.

Таблиця 3.5. Однакова втрата органічних кліків має різну економічну вагу залежно від того, де фактично відбувається цінна дія.

Тому я не підтримую ідею просто «замінити трафік на видимість». Для медіа трафік може залишатися головною економічною одиницею. Для інтернет-магазину дохід важливіший за сесії. Для бренду без прямої онлайн-транзакції важливіші брендовий попит і відкладені конверсії. Правильна метрика визначається бізнес-моделлю, а не модним словом у SEO.

3.12. 60-хвилинний аудит падіння органічних кліків

Цей аудит потрібен не для фінального висновку, а для первинної діагностики. За годину я хочу зрозуміти, в який бік копати: попит, позиції, SERP або якість трафіку. Якщо команда після цього все ще не може сформулювати гіпотезу одним реченням, даних недостатньо.

Час	Що робимо	Що має вийти
0-10 хв	GSC: 28 днів проти попередніх 28 та порівняння рік до року. Розбиваємо кліки, покази, CTR, позицію за країною та пристроєм.	Розуміння: падіння попиту, видимості чи саме віддачі кліків.
10-20 хв	Брендові / небрендові запити і головні кластери запитів.	Чи не змінився сам склад попиту.

Час	Що робимо	Що має вийти
20-35 хв	20-50 URL зі стабільними показами й позиціями, але нижчим CTR.	Кандидати на стискання CTR у SERP.
35-45 хв	Перевіряємо елементи SERP та AI Overview для ключових запитів; дивимося історію там, де є.	Що саме з'явилося між URL і користувачем.
45-55 хв	Звіт про генеративний AI: сторінки, країни, пристрої; зіставляємо зі звичайним звітом ефективності.	Чи росте AI-видимість на сторінках із падінням CTR.
55-60 хв	GA4 / CRM / дохід: CVR, дохід на органічну сесію, маржа.	Чи падіння кліків реально погіршило бізнес-результат.

Таблиця 3.6. Швидкий аудит zero-click / стискання CTR. Його мета - сформувати перевірювану гіпотезу, а не знайти винного за одну годину.

НОТАТКА З ПРАКТИКИ / Якщо після аудиту є лише речення «AI забрав трафік», аудит провалений. Хороша гіпотеза звучить конкретніше: «На мобільних небрендових інформаційних запитах у США покази й позиція майже не змінилися, CTR впав після появи AI Overview, а AI-покази цих URL ростуть». Таку гіпотезу вже можна перевіряти.

3.13. Що ми точно знаємо, а що лишається гіпотезою

Ми точно знаємо, що пошуки без кліку становлять більшість Google-пошуків у великих дослідженнях поведінки користувачів 2024-2026 років, хоча точний відсоток залежить від панелі, країни й методології.[1][2][3] Ми точно знаємо, що в дослідженні Pew користувачі рідше клікали по звичайних результатах, коли на сторінці був AI-підсумок, і майже не клікали по джерелах усередині AI-підсумку.[4] Ми знаємо, що Ahrefs на великій вибірці отримав сильну негативну асоціацію між AI Overview та CTR сторінок на найвищих позиціях.[5]

Ми також знаємо, що Google публічно стверджує про відносно стабільний загальний органічний обсяг кліків і вищу якість частини

кліків, але не публікує сирий набір даних, який дозволив би незалежно відтворити цей висновок.[6] Це треба подавати саме як заяву Google, а не як встановлений зовнішнім дослідженням факт.

Ми не знаємо універсального «штрафу CTR від AI» для будь-якого URL. Не знаємо, скільки втрачених прямих кліків повертається потім через бренд, прямий захід або інший канал. Не можемо з GSC напряму порахувати CTR AI Overview у 2026 році, бо окремий звіт про генеративний AI показує покази, але не дає окремих кліків і запитів. [11]

Тому нормальна стратегія не починається з боротьби з zero-click. Вона починається з розуміння, де саме бізнес заробляє. Якщо сайт монетизує кожен перегляд сторінки, стискання CTR - критичний ризик. Якщо органічний пошук формує попит, який закривається пізніше, треба вміти виміряти цей відкладений ефект. А якщо кліки стають рідшими, але краще конвертуються, KPI має показати це, а не просто намалювати червону стрілку біля сесій.

У наступній главі я розберу саме цей рівень: що означає видимість без кліку, які її форми реально можна виміряти і де починаються метрики, які виглядають красиво, але не мають надійного знаменника.

Джерела до глави

1. SparkToro. In 2026, Less than One Third of Google Searches Still Send a Click. https://sparktoro.com/blog/in-2026-less-than-one-third-of-google-searches-still-send-a-click/ Панель Similarweb для мобільних пристроїв і комп'ютерів, США, січень-квітень 2026; 68.01% пошуків без кліку; методологія та обмеження.

2. SparkToro. Zero-Click Searches: Highest in the UK, Lowest in Germany, and France has the Most Efficient Searchers. https://sparktoro.com/blog/zero-click-searches-highest-in-the-uk-lowest-in-germany-and-france-has-the-most-efficient-searchers/ Порівняння шести країн на даних Similarweb, включно з часткою пошуків без кліку та кліками до відкритого вебу на 1 000 пошуків.

3. SparkToro. 2024 Zero-Click Search Study. https://sparktoro.com/blog/2024-zero-click-search-study-for-every-1000-us-google-searches-only-374-clicks-go-to-the-open-web-in-the-eu-its-360/ Панель поведінкових даних Datos; США та ЄС; 2024. Використовується для історичного контексту із застереженням про різні панелі.

4. Pew Research Center. Google users are less likely to click on links when an AI summary appears in the results.
https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/ 900 дорослих у США, 68 879 унікальних Google-пошуків; поведінка після SERP з AI-підсумком і без нього.

5. Ahrefs. Update: AI Overviews Reduce Clicks by 58%. https://ahrefs.com/blog/ai-overviews-reduce-clicks-update/ 300 000 ключових слів; GSC; порівняння CTR для SERP з AI Overview та без нього.

6. Google. AI in Search is driving more queries and higher quality clicks.
https://blog.google/products-and-platforms/products/search/ai-search-driving-more-queries-higher-quality-clicks/ Офіційна позиція Google щодо сукупних органічних кліків та їхньої якості. Внутрішні дані Google без опублікованого сирого набору даних.

7. Google. AI in Search: Going beyond information to intelligence / AI Overviews expansion. https://blog.google/products-and-platforms/products/search/google-search-ai-mode-update/ Заява Google про понад 10% зростання використання для типів запитів, де показуються AI Overviews.

8. Similarweb. AI Referral Traffic by Industry: What 770 Million Monthly Visits Reveal. https://aisearch.similarweb.com/blog/ai-referral-traffic-by-industry/ 770.7 млн середніх щомісячних переходів з AI-джерел, червень 2025 - травень 2026; +117.4% рік до року.

9. Adobe. AI traffic grows but retail sites lag in AI search visibility.
https://business.adobe.com/uk/blog/ai-traffic-surge-retail-sites-not-machine-readable Березень 2026: переходи з AI-джерел на роздрібні сайти США конвертували на 42% краще за інші переходи.

10. Adobe. How AI is reshaping travel planning / retail update.
https://business.adobe.com/blog/us-consumers-are-embracing-llms-to-make-travel-plans Липень 2026: переходи з AI-джерел на роздрібні сайти США конвертували на 60% краще; показники взаємодії та відмов.

11. Google Search Console Help. Generative AI performance report (Search).
https://support.google.com/webmasters/answer/16984139?hl=en Окремі покази генеративного AI для AI Overviews та AI Mode; виміри, обмеження, експорт.

12. Google Search Console Help. What are impressions, position, and clicks?
https://support.google.com/webmasters/answer/7042828?hl=en Офіційні визначення показів, кліків, CTR та позиції.

13. Google Search Console Help. Query guidelines and sample queries / Table guidelines and reference. https://support.google.com/webmasters/answer/12917174?hl=en Офіційна схема масового експорту до BigQuery, агрегація, SUM(impressions), SUM(clicks), sum_position і приклади SQL.

ГЛАВА 4

Видимість без кліку: що саме тепер має вимірювати SEO

Як розділити покази, згадки, цитування, переходи та конверсії, не змішувати дані Google з модельними оцінками сторонніх сервісів і побудувати звітність, яка не створює фальшивого відчуття точності

Стан даних і джерел: 12 вересня 2026 року

Проблема сучасної SEO-аналітики вже не в нестачі метрик. Їх стало забагато. Позиції, покази, кліки, генеративні покази, цитування, згадки бренду, частка голосу в AI-відповідях, переходи з ChatGPT, асоційовані конверсії, брендовий пошук, прямий трафік. Кожен інструмент додає власний індекс, а в дашборді все це легко перетворюється на одну красиву цифру, яка нічого конкретного не означає.

У цій главі я навмисно не шукаю одну головну метрику AI-видимості. Її немає. Є кілька різних подій, які відбуваються на різних етапах шляху користувача і вимірюються різними джерелами. Якщо команда не розділяє ці події, вона починає порівнювати покази Google з цитуваннями Bing, згадки сторонньої системи моніторингу з реальними сесіями GA4, а оцінену частку присутності - з фактичним ринковим попитом. На рівні Senior SEO або SEO Lead це вже не похибка. Це помилка моделі.

Рис. 4.1. Видимість, згадка, цитування, клік і конверсія - різні події. Між ними може бути зв'язок, але жодна з них автоматично не доводить наступну.

4.1. Спочатку визначаємо подію, а вже потім метрику

Класична органічна аналітика була відносно зручною, бо основні події формували зрозумілий ланцюг: показ у SERP -> клік -> сесія -> конверсія. Навіть тут були проблеми з агрегацією, атрибуцією і різницею між Search Console та аналітикою сайту, але принаймні самі події були зрозумілими.

Генеративний пошук додає кілька подій до кліку. Сторінка може з'явитися в AI Overview. Бренд може бути названий у тексті без посилання. URL може бути процитований як джерело. Документ може бути знайдений системою, але не отримати видиме цитування. Після відповіді користувач може не клікнути взагалі, а через день виконати брендовий запит. Усі ці сценарії мають різний економічний зміст.

Подія	Що реально сталося	Що НЕ можна автоматично стверджувати
Показ у класичному Search	Посилання сайту отримало impression за правилами Search Console	Що користувач його прочитав або запам'ятав.
Показ у генеративній функції Google	Посилання сайту було показане в AI Overview або AI Mode	Що був клік, citation у сторонній LLM або вплив на бренд.
Згадка бренду	Назва бренду з'явилася у відповіді моделі	Що модель використала сайт бренду як джерело.

Подія	Що реально сталося	Що НЕ можна автоматично стверджувати
Цитування URL	Сторінку показано як джерело в AI-відповіді	Що користувач відкрив URL або що citation була першою / найпомітнішою.
Перехід	Користувач відкрив сайт із конкретного джерела	Що цей канал був єдиною причиною рішення.
Конверсія	Відбулася визначена бізнес-дія	Що останній канал перед конверсією створив увесь попит.

Таблиця 4.1. Найпоширеніша помилка - надавати одній події значення іншої.

ПРАВИЛО / Кожен KPI у звіті має відповідати на одне речення: «Яку конкретну подію ми порахували і який у неї знаменник?». Якщо на це питання немає короткої відповіді, метрика ще не готова до управлінського звіту.

4.2. Жоден інструмент не бачить увесь шлях

У 2026 році найбезпечніша архітектура вимірювання - не шукати один сервіс, який «міряє AI», а зшивати кілька шарів даних. Пошукова система бачить власні покази та частину взаємодій. Вебаналітика бачить реальні переходи після кліку. CRM бачить продаж або лід. Стороння система моніторингу бачить лише той набір промптів і відповідей, які сам перевірив.

Рис. 4.2. Джерела даних бачать різні частини воронки. Матриця показує нативну доступність сигналу, а не загальну «якість» інструмента.

Google: офіційна документація Search Console. Bing: офіційний опис звіту AI Performance. ChatGPT: офіційне UTM-маркування переходів. Стороння система моніторингу: вибіркова перевірка промптів, а не доступ до приватних сесій користувачів.

Це принципова межа. Сторонній інструмент не має доступу до всіх приватних діалогів ChatGPT, Gemini або Perplexity. Ahrefs прямо описує Brand Radar як вибірковий підхід: сервіс запускає великі набори промптів і записує відповіді, але не бачить кожну реальну згадку бренду в усіх користувацьких сесіях.[6] Саме тому «AI-видимість» стороннього сервісу - це індекс або структурована вибірка, а не перепис населення.

І навпаки, Google Search Console має дані самої платформи про Google, але не показує ChatGPT, Copilot чи Perplexity. Bing Webmaster Tools показує власну екосистему і частину партнерських інтеграцій, але не є глобальним AI-аналітичним центром. GA4 побачить клік із ChatGPT, але не побачить десять відповідей ChatGPT, у яких ваш бренд згадали без переходу.

4.3. Google Search Console: вперше є окремий вимір генеративної видимості

3 червня 2026 року Google анонсував окремі звіти про генеративний AI у Пошуку, а 31 серпня повідомив про глобальне розгортання для всіх сайтів. Окремий звіт включає AI Overviews та AI Mode і показує саме покази - скільки разів посилання на сайт були показані користувачу в генеративній функції Пошуку Google.[1][2]

У звіті доступні сторінки, країни, дати та пристрої. Дані по сторінках переважно прив’язані до канонічного URL. Графік агрегується на рівні ресурсу, а таблиця змінює агрегацію залежно від вибраного виміру. На звіт також поширюється стандартне обмеження таблиці Search Console у 1 000 рядків.[2]

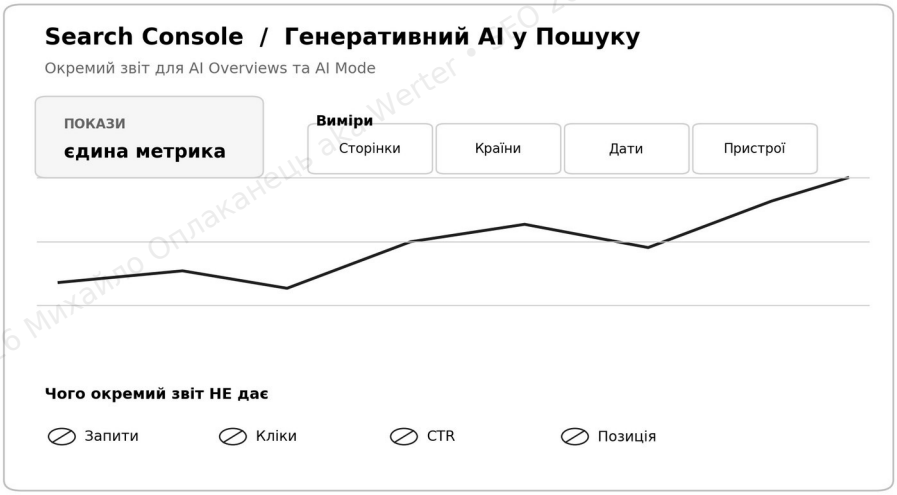


Рис. 4.3. Що дає окремий звіт Google Search Console про генеративний AI станом на 12.09.2026.

Схематичне відтворення. Джерело функціональності: Google Search Console Help, Generative AI performance report (Search).

Найважливіше тут не те, що Google нарешті показує AI-покази. Найважливіше - чого окремий звіт не показує. Станом на дату зрізу в ньому немає окремих кліків, CTR, середньої позиції та виміру запитів. Тобто ми можемо побачити, що /guide/seo-audit/ отримує генеративні покази, але не можемо з цього звіту відповісти, які саме запити дали

ці покази і скільки переходів прийшло саме з AI Overview або AI Mode.[2]

При цьому AI Overviews та AI Mode входять у звичайний звіт «Ефективність». Google окремо описує, що клік по зовнішньому посиланню в AI Mode або AI Overview рахується як клік, а покази та позиція підпорядковуються стандартним правилам Search Console. Для AI Overview всі посилання в блоці отримують одну позицію самого блоку; уточнювальне питання в AI Mode рахується як новий запит.[3] Це важливо, бо загальні кліки й покази вже містять AI-події, але окремий генеративний звіт дає лише виділений зріз показів.

ПОПЕРЕДЖЕННЯ / Не діліть загальні кліків із звичайний звіт «Ефективність» на impressions з окремого звіту про генеративний AI і не називайте це «CTR генеративних функцій». Чисельник містить різні типи Search-переходів, а знаменник - лише генеративні покази. Метрика математично не відповідає назві.

4.4. Корисна метрика з GSC: частка генеративних показів, але з правильною назвою

З окремого звіту Google вже можна побудувати просту операційну метрику: частку генеративних показів серед усіх показів вебпошуку для того самого ресурсу, періоду, країни і пристрою. Я називаю її часткою генеративної експозиції, а не «часткою AI-запитів».

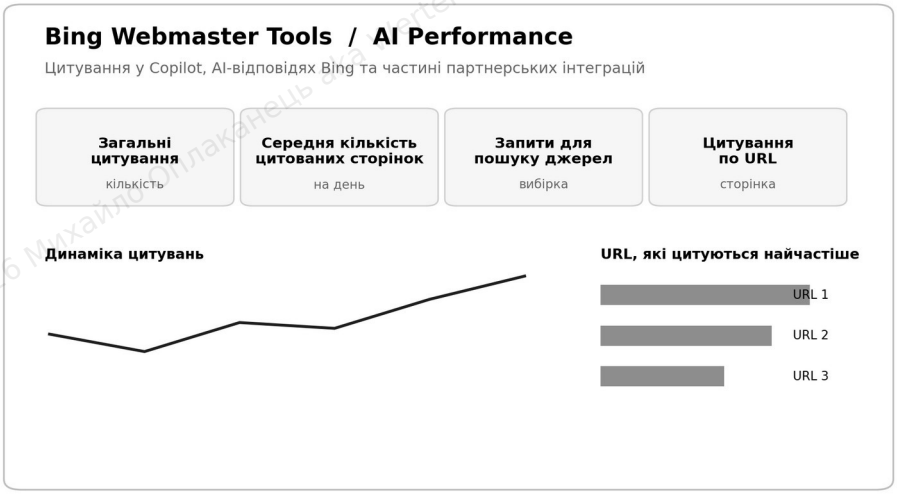
```
Частка генеративної експозиції =
генеративні AI-покази / усі покази вебпошуку

Використовувати тільки з однаковими:
- датами;
- країною;
- пристроєм;
- рівнем агрегації.

Не називати це:
- часткою пошуків;
- часткою користувачів;
- CTR генеративних функцій;
- часткою трафіку.
```

Формула 4.1. Це відношення показів вашого ресурсу, а не оцінка частки всіх запитів Google.

Наприклад, якщо генеративні покази ростуть, а загальні покази стабільні, це означає, що більша частина вашої видимості відбувається всередині генеративних функцій Google. Це ще не говорить, добре це чи погано. Далі треба дивитися CTR у звичайному звіті «Ефективність», сторінки, країни, пристрої та бізнес-результат. Але вже можна побачити структурний зсув у типі видимості.

4.5. Bing Webmaster Tools дає інший шматок системи: цитування та запити для пошуку джерел

10 лютого 2026 року Microsoft відкрив публічну тестову версію звіту AI Performance у Bing Webmaster Tools. За офіційним описом звіт охоплює Microsoft Copilot, AI-підсумки у Bing та частину партнерських інтеграцій. На відміну від Google, Bing у цьому звіті концентрується не на показах, а на активності цитувань.[4]



Схематичне відтворення набору метрик з офіційного опису Microsoft.

Рис. 4.4. Набір даних Bing AI Performance принципово інший: цитування URL та запити, які використовувалися для пошуку джерел.

Схематичне відтворення офіційно описаних метрик Microsoft, а не скріншот реального ресурсу.

Показник «Загальні цитування» (Total Citations) показує кількість цитувань сайту як джерела в AI-відповідях. «Середня кількість цитованих сторінок» (Average Cited Pages) - середню кількість

унікальних сторінок сайту, що цитувалися за день. «Активність цитувань на рівні сторінок» (Page-level citation activity) розкладає кількість цитувань по URL. «Запити для пошуку джерел» (Grounding queries) показують ключові фрази, які AI використовував при отриманні контенту, що потім був процитований.[4]

Остання метрика особливо корисна, але її легко переоцінити. Microsoft прямо пише, що дані «Запитів для пошуку джерел» у звіті є вибіркою загальної активності цитувань, тобто вибіркою, а не повним журналом усіх внутрішніх запитів системи. Так само Показник Total Citations не показує місце або спосіб показу цитування усередині конкретної відповіді, а Average Cited Pages не є оцінкою авторитету чи ранжування.[4]

ДАНІ / Google у 2026 році дає окремі AI-покази. Bing дає дані про цитування і вибірку запитів для пошуку джерел. Це не конкуруючі версії однієї метрики, а різні спостереження за різними етапами системи.

4.6. ChatGPT: реальні переходи можна виміряти краще, ніж згадки без кліку

Для ChatGPT ситуація дзеркальна. OpenAI не надає вебмайстру Search Console з усіма показами або згадками. Але для переходів є дуже чистий сигнал: OpenAI офіційно вказує, що ChatGPT автоматично додає до вихідних URL параметр utm_source=chatgpt.com. Видавець може відстежувати ці переходи у Google Analytics або іншій аналітиці.[5]

Це означає, що сесія з ChatGPT є значно твердішим фактом, ніж «оцінена видимість у ChatGPT» стороннього сервісу. Ми знаємо, що людина натиснула посилання і прийшла. Далі можна порівнювати посадкові сторінки, залучені сесії, CVR, дохід, якість лідів і повторні візити. Але й тут UTM не показує, скільки разів бренд був згаданий без кліку або скільки відповідей ChatGPT містили цитування, яку користувач не відкрив.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ

Я завантажу CSV з GA4, де є дата, landing page + query string, sessions і, якщо доступно, key events або revenue. Перевір колонки. Відфільтруй посадкові URL, які містять utm_source=chatgpt.com, порахуй сесії за датою і посадковою сторінкою, а також конверсії та дохід, якщо ці поля є. Не прирівнюй цей трафік до всіх згадок бренду в ChatGPT: це лише зафіксовані переходи.

АЛЬТЕРНАТИВА БЕЗ ШІ

У GA4 Explorations додайте вимір Landing page + query string, метрики Sessions та Key events і фільтр contains "utm_source=chatgpt.com". Для регулярної автоматизації використовуйте BigQuery-запит нижче.

QR: КОД

Скануйте камерою: QR містить компактну виконувану версію коду без коментарів. Коментований код наведено нижче.

ДЛЯ АВТОМАТИЗАЦІЇ: SQL ДЛЯ GA4 BIGQUERY

```
-- Завдання: порахувати GA4-сесії, що почались із URL,
-- де OpenAI додав utm_source=chatgpt.com.
-- Замініть PROJECT.analytics_XXXXXXXXX на свій GA4 BigQuery dataset.

WITH starts AS (
  SELECT
    PARSE_DATE('%Y%m%d', event_date) AS date,
    user_pseudo_id,

    -- Ідентифікатор сесії з event_params.
    (SELECT value.int_value
       FROM UNNEST(event_params)
      WHERE key = 'ga_session_id') AS ga_session_id,

    -- Повний URL сторінки, на яку прийшов користувач.
    (SELECT value.string_value
       FROM UNNEST(event_params)
      WHERE key = 'page_location') AS page_location

  FROM `PROJECT.analytics_XXXXXXXXX.events_*`
  WHERE event_name IN ('session_start','page_view')
)

SELECT
  date,

  -- Рахуємо унікальні сесії, а не кількість подій.
  COUNT(DISTINCT CONCAT(
    user_pseudo_id, '-', CAST(ga_session_id AS STRING)
  )) AS sessions
```

```
FROM starts
WHERE REGEXP_CONTAINS(
  page_location,
  r'(?i)[?&]utm_source=chatgpt\.com(?:&|$)'
)
GROUP BY date
ORDER BY date;
```

Код 4.1. Мінімальний спосіб виділити переходи ChatGPT за офіційним UTM. Для робочого звіту я б далі дедуплікував першу подію на посадковій сторінці і приєднав конверсії та дохід на рівні ідентифікатора сесії.

4.7. Згадка і цитування - не одне й те саме

На рівні стороннього моніторингу я завжди розділяю згадку і цитування. Згадка означає, що бренд присутній у тексті AI-відповіді. Цитування означає, що домен або конкретна сторінка показані як джерело. Ahrefs саме так розділяє свої метрики: одна згадка рахується, якщо бренд з'явився хоча б раз у відповіді, а одне цитування - якщо сторінка або домен були процитовані хоча б раз.[7]

Це дає чотири практично різні сценарії. Бренд може бути згаданий і процитований. Може бути згаданий без цитування, наприклад як відомий продукт. Сайт може бути процитований без явної згадки бренду в основному тексті. А може не з'явитися взагалі. Якщо звести все до одного індексу, ми втрачаємо діагноз.

Сценарій	Що бачимо	Ймовірна задача
Згадка + цитування власного URL	Бренд присутній, сайт використано як джерело	Масштабувати теми й URL, де вже є сильна присутність.
Згадка без цитування	Бренд відомий моделі, але джерелом виступає інший сайт	Перевірити сторонні джерела, PR, сторінки оглядів і власне покриття теми.
Цитування без явної згадки бренду	Сторінка працює як джерело фактів	Посилити зв'язок із брендом, авторство, продуктову ідентичність, конверсійний шлях.
Немає згадки й цитування	У вибірці бренд не присутній	Перевірити релевантність набору промптів, покриття теми, авторитет і джерела конкурентів.

Таблиця 4.2. Згадка і цитування відповідають на різні питання. Відсутність цитування не завжди означає відсутність брендової видимості.

4.8. Сторонні AI-метрики корисні, якщо не видавати вибірку за повний масив реального використання

Стороння система моніторингу потрібен там, де сама платформа не дає достатньо даних. Він може стабільно перевіряти однаковий набір промптів, фіксувати відповіді, цитування, згадки, конкурентів і джерела. Для аналізу динаміки це дуже корисно. Проблема починається, коли отриманий індекс називають «реальною часткою присутності серед усіх користувачів ChatGPT».

Ahrefs станом на серпень 2026 року заявляє 459+ млн промптів у шести AI-індексах. Найбільший сегмент - AI Overviews, 308.6 млн; окремі індекси Gemini, Perplexity, ChatGPT і Copilot мають близько 31 млн промптів, AI Mode - 25.3 млн.[8] Це дуже велика вибірка, але вона все одно є побудованим індексом, а не приватними журналами реального використання платформ.

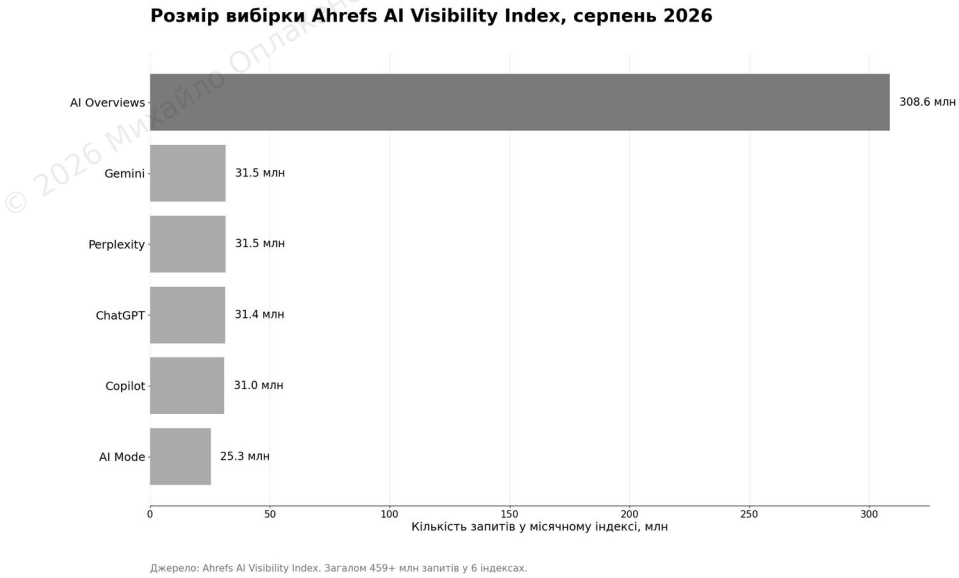


Рис. 4.5. Кількість промптів у шести індексах Ahrefs AI Visibility Index станом на серпень 2026 року. Дані самого Ahrefs.

Ahrefs моделює промпти з реального пошукового попиту та People Also Ask. Великий розмір вибірки не перетворює її на повний лог усіх взаємодій користувачів із LLM.

Ще важливіше читати визначення метрик. Метрика Ahrefs Estimated Impressions («оцінені покази») - це не реальні покази самої платформи ChatGPT. Сервіс оцінює їх через частотність пошуку пов'язаних запитів Google / People Also Ask; частка голосу в AI - відносна частка цих оцінених показів серед відстежуваних брендів.[7] Semrush використовує іншу систему: AI Visibility як порівняльний індекс 0-100, згадки, оцінку місячної аудиторії, цитування та інші власні метрики.[9] Тому число 37 у одному сервісі не можна порівнювати з 37 в іншому.

Метрика	Робоче визначення для книги	Правильний знаменник
Частота згадок	Частка перевірених відповідей, де бренд згаданий хоча б раз	Кількість відповідей у зафіксованій вибірці промптів.
Частота цитувань	Частка перевірених відповідей, де домен / URL процитований	Кількість відповідей у тій самій вибірці.
Співвідношення цитувань до згадок	Скільки відповідей зі згадкою бренду одночасно цитують наш домен	Кількість відповідей зі згадкою бренду.
Частка голосу в AI	Частка згадок / цитувань бренду відносно визначеного набору конкурентів	Тільки конкретний набір промптів, платформа, країна, дата і список конкурентів.
CVR переходів з AI	Конверсії з ідентифікованих AI-сесій / AI-сесії	Реальні сесії з переходів у власній аналітиці.
Дохід на AI-сесію	Дохід з переходів з AI / AI-сесії	Реальні сесії, а не оцінені покази.

Таблиця 4.3. Власні метрики мають бути відтворюваними. Назва без знаменника створює красиву, але слабку аналітику.

4.9. Найнебезпечніша помилка - знаменник, який змінюється непомітно

У класичному відстеженні позицій ми звикли до відносно стабільного набору ключових слів. У AI-моніторингу набір промптів

може змінюватися набагато сильніше: сервіс додає нові запити, міняє механізм формування набору промптів, підключає іншу модель, змінює географію, оновлює конкурентів. Після цього частка присутності може змінитися навіть без реальної зміни присутності бренду.

Тому для власного моніторингу я фіксую принаймні п’ять речей: точний текст промпту або його стабільний ідентифікатор, платформу і модель, країну або мову, дату перевірки та список конкурентів. Якщо сервіс дозволяє щоденний моніторинг власних промптів, історію не можна змішувати з готовий індекс без окремої мітки. Це різні вибірки.

МІФ / «частка голосу в AI = частка реального ринку». Ні. Це частка присутності всередині конкретної системи вимірювання. Змінився набір промптів, конкурентів, моделей або ваг - змінився і знаменник.

4.10. Як я розділяю джерела за доказовою вагою

Не всі дані однаково близькі до грошей. Якщо CRM показує 120 оплачуваних контрактів, це сильніший бізнес-факт, ніж 18 000 оцінених AI-показів стороннього сервісу. Це не означає, що покази не потрібні. Вони потрібні для ранньої діагностики. Але управляти бюджетом лише верхнім шаром небезпечно.

Рис. 4.6. Ієрархія доказової ваги для SEO-звітності. Верхні рівні ближчі до фактичного бізнес-результату; нижні корисні для напрямку та діагностики.

Звідси практичний принцип: у звіті керівнику я не ставлю «індекс AI-видимості» вище за ліди, дохід і реальні переходи. Але для SEO-команди частота цитувань або частка генеративних показів може бути раннім індикатором, який показує зміни раніше, ніж вони дійдуть до доходу. Хороша система не відкидає слабші сигнали. Вона просто не плутає їхню доказову вагу.

4.11. Видимість без кліку і бренд: виміряти можна, довести причинність важче

Найскладніша зона починається там, де користувач бачить бренд у AI-відповіді, не клікає, а потім приходить іншим шляхом. Стандартна вебаналітика не має події першого контакту, тому атрибуція конверсії може піти брендовий органічний трафік, прямий трафік, платний пошук або навіть пошта. Це не баг аналітики. Події справді немає на вашому сайті.

Рис. 4.7. Один із можливих шляхів впливу без кліку. Без додаткового дизайну вимірювання він правдоподібний, але не доведений.

Що можна зробити практично. Перше - дивитися попит на брендові запити у Search Console і тренд прямого та брендового органічного трафіку поруч зі змінами AI-видимістю, але трактувати це як кореляцію. Друге - використовувати контрольні ринки, GEO або часові вікна, якщо бізнес дозволяє природний експеримент. Третє - для генерації лідів питати джерело знайомства в CRM або формі, але не змушувати користувача вибирати лише один канал. Четверте - дивитися асоційовані конверсії і дані шляхів, розуміючи, що видимість без кліку до сайту там усе одно не з'явиться.

Якщо після росту AI-згадок брендові запити виросли на 20%, це ще не доказ, що одне спричинило інше. Можливо, одночасно була PR-кампанія, реклама, сезонність або реліз продукту. У книзі я навмисно не перетворюю цю красиву історію на причинний факт без контролю.

4.12. Мінімальна звітна панель Senior SEO у 2026 році

Моя базова звітна панель тепер має не один графік «Органічний трафік», а щонайменше чотири шари. Перший - попит і класичний Пошук. Другий - генеративна видимість у джерелах самої платформи. Третій - вибіркова присутність у сторонніх AI-платформах. Четвертий - реальні сесії та бізнес-результат.

Шар	Мінімальні KPI	Розрізи	Частота
Класичний Пошук	Кліки, покази, CTR, середня позиція, брендові / небрендові запити	Сторінка, кластер, країна, пристрій	Щотижня + 28d/YoY
Генеративний Пошук Google	Генеративні покази, частка генеративної експозиції	Сторінка, країна, пристрій, дата	Щотижня
Bing AI	Цитування, цитовані URL, вибірка запитів для пошуку джерел	URL, тема, дата	Щотижня / щомісяця
Моніторинг LLM	Частота згадок, частота цитувань, share of voice у фіксованому набір промптів	Платформа, GEO, тема, конкурент	Щотижня або щомісяця
Переходи з AI-платформ	Сесії, залучені сесії, CVR, дохід / якість лідів	Джерело, посадкова сторінка, пристрій, країна	Щотижня
Бізнес-результат	Ліди, продажі, дохід, маржа, CAC / EPC де доречно	Продукт, ринок, етап воронки	Щотижня / щомісяця

Таблиця 4.4. Мінімальна система вимірювання. Не кожному бізнесу потрібні всі рядки, але кожен KPI має мати зрозуміле джерело і власника.

Ключове правило порівнянь: однакова гранулярність. Не порівнюйте щоденний моніторинг власних промптів із місячним індексом. Не порівнюйте генеративні покази на рівні URL з загальним показником на рівні ресурсу без перевірки агрегації. Не змішуйте США з глобальними даними. Не робіть висновок по одному дню, якщо платформа й модель мають високу варіативність відповіді.

4.13. Три типи помилок у звітних панелях, які я бачу найчастіше

Помилка	Як виглядає	Чому небезпечно	Як виправити
Фальшивий CTR генеративних функцій	Загальні кліки GSC / генеративні покази	Різні набори подій у чисельнику й знаменнику	Не рахувати до появи окремих кліків або надійного джерела на рівні подій.

Помилка	Як виглядає	Чому небезпечно	Як виправити
AI-індекс як абсолютна істина	«Індекс видимості = 62/100, отже ми займаємо 62% ринку»	Індекс залежить від методології сервісу	Показувати динаміку і методологію; поруч - згадки / цитування та покриття промптів.
Сума різних платформ	покази Google + цитування Bing + згадки ChatGPT = «загальна видимість»	Одиниці виміру неадитивні	Тримати окремі панельні метрики; об'єднувати тільки нормалізованим індексом з явною формулою.
Відсутність бізнес-шару	Звіт закінчується цитуваннями	Немає відповіді, чи створюється цінність	Додати сесії з переходів, конверсії, дохід / ліди та якість.
Рухомий набір промптів	Щомісяця додаються промпти, але тренд читають як чистий ріст	Змінюється знаменник	Фіксувати основний набір промптів і окремо експериментальний / дослідницький набір.

Таблиця 4.5. Більшість помилок AI-аналітики - не в даних, а в неправильному знаменнику або змішуванні одиниць виміру.

4.14. Як побудувати власний набір промптів, який не перетвориться на театр метрик

Якщо я сам задаю набір промптів для моніторингу, я не починаю з питання «які фрази змусити AI сказати наш бренд». Це створює самопідтверджувальний орієнтир. Починаю з реального простору запитів: Search Console, розмов із відділом продажів, звернень у підтримку, People Also Ask, внутрішній пошук, запити до продукту, конкурентні порівняння. Далі групую промпти за задачею користувача, а не за тим, де бренд уже перемагає.

Клас промпту	Приклад структури	Навіщо відстежувати
Категорійний	«Які сервіси / продукти найкращі для X?»	Чи присутній бренд у широкому первинному комерційному виборі.
Порівняльний	«X проти Y для задачі Z»	Конкурентна позиція та аргументи.

Клас промпту	Приклад структури	Навіщо відстежувати
Проблемний	«Як вирішити проблему X?»	Чи бренд / контент потрапляє у простір можливих рішень до прямого продуктового запиту.
Доказовий	«Які джерела / дані підтверджують X?»	Видимість через цитування для досліджень, статистики й експертного контенту.
Транзакційний	«Що вибрати / купити / забронювати за умов X?»	Найближчий до бізнес-результату сегмент.
Брендовий	«Чи надійний бренд X? Які альтернативи?»	Наратив, тональність, конкуренти та джерела репутації.

Таблиця 4.6. Основний набір промптів має представляти реальні задачі користувача, а не лише запити, зручні для бренду.

Я тримаю основний набір стабільним для тренду. Нові промпти йдуть у дослідницький набір і не впливають на основний часовий ряд, доки не зафіксована нова база. Це простий спосіб не намалювати ріст або падіння тільки тому, що змінилися питання.

4.15. Операційний стандарт: що перевіряти щотижня, а що щомісяця

Ритм	Що перевіряємо	Яке рішення це підтримує
Щотижня	генеративні покази GSC для основних URL; цитування Bing; сесії з AI-переходів; аномалії згадки / цитування у основний набір промптів	Чи зламалась видимість, чи з’явилась нова можливість, чи треба швидка діагностика.
Раз на 28 днів	Частка генеративних показів; динаміка класичного Search проти AI; покриття промптів за темами; концентрацію цитувань за сторінками; CVR переходів	Чи є структурний зсув, а не денний шум.

Ритм	Що перевіряємо	Яке рішення це підтримує
Щомісяця	Конкурентна частка присутності у стабільній вибірці; попит на брендові запити; асоційовані конверсії; бізнес-результат	Пріоритизація контенту, PR, отримання посилань, бюджет.
Щокварталу	Перегляд основний набір промптів, джерел даних, методології індексів, переліку платформ	Чи сам система вимірювання не застарів після зміни продуктів і моделей.

Таблиця 4.7. Частота вимірювання має відповідати швидкості зміни сигналу. Щоденний шум не повинен керувати квартальним бюджетом.

4.16. Що ми точно знаємо, а що поки не можемо виміряти нормально

Станом на 12 вересня 2026 року ми точно можемо виміряти окремі генеративні покази у Google Search Console для AI Overviews та AI Mode.[1][2] Ми можемо отримати дані про цитування і вибірку запитів для пошуку джерел у Bing Webmaster Tools.[4] Ми можемо надійно виділяти переходи з ChatGPT, оскільки OpenAI автоматично додає utm_source=chatgpt.com.[5] Ми можемо запускати стабільний набір промптів у сторонніх трекерах і рахувати згадки й цитування та конкурентну присутність, якщо пам’ятаємо, що це вибірка.[6][7]

Ми не маємо універсального лічильника від самої платформи всіх згадок бренду в усіх приватних сесіях LLM. Не маємо одного стандарту частки голосу в AI між різними сервісами. Не можемо з окремого GSC звіту про генеративний AI порахувати точний CTR генеративних функцій, бо звіт не містить окремих кліків. Не можемо довести, що зростання брендового пошуку після росту AI-згадок спричинене саме AI, якщо немає контрольного дизайну.

Тому правильна звітність у 2026 році не намагається приховати прогалини. Вона показує їх. Для мене сильна звітна панель - це не той, де є одна магічна цифра. Це той, де видно, які події виміряні даними самої платформи, які відновлені через власну аналітику, які отримані на вибірці і де ми маємо лише робочу гіпотезу.

На цьому Частина I закінчується. Далі переходимо від інтерфейсів і вимірювання до механіки самої пошукової системи. Почнемо з найпершого питання, яке часто пропускають: звідки пошукова система взагалі дізнається, що URL існує.

Джерела до глави

1. Google Search Central Blog. Introducing Search Generative AI performance reports in Search Console. https://developers.google.com/search/blog/2026/06/gen-ai-performance-reports Анонс 3 червня 2026 року; глобальне розгортання 31 серпня; покази, сторінки, країни, пристрої, дати.

2. Google Search Console Help. Generative AI performance report (Search). https://support.google.com/webmasters/answer/16984139?hl=en AI Overviews та AI Mode; окремі генеративні покази; сторінки, країни, дати, пристрої; агрегація, 1 000 рядків, кнопка експорту.

3. Google Search Console Help. What are impressions, position, and clicks? https://support.google.com/webmasters/answer/7042828?hl=en Правила для AI Mode та AI Overviews: click, impression, position, follow-up query.

4. Microsoft Bing Webmaster Blog. Introducing AI Performance in Bing Webmaster Tools Public Preview. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview Total Citations, Average Cited Pages, grounding queries sample, page-level citation activity, trends.

5. OpenAI Help Center. Publishers and Developers FAQ. https://help.openai.com/uk-ua/articles/12627856-publishers-and-developers-faq OpenAI підтверджує автоматичне utm_source=chatgpt.com для переходів із результатів пошуку ChatGPT.

6. Ahrefs FAQ. AI-видимістю & Brand Radar. https://ahrefs.com/faq Пряме застереження: AI-видимістю monitoring є sampling-based, не real-time і не має доступу до приватних розмов або внутрішніх даних OpenAI.

7. Ahrefs Help Center. AI Visibility Metrics. https://help.ahrefs.com/en/articles/15501968-ai-visibility-metrics Визначення Згадки, Цитування, Found in, Estimated Impressions та AI Share of Voice; метод оцінювання показів через частотність пошуку Google.

8. Ahrefs. AI Visibility Index. https://ahrefs.com/ai-visibility-index 459+ млн промптів у шести індексах; кількість промптів по AI Overviews, Gemini, Perplexity, ChatGPT, Copilot та AI Mode; серпень 2026.

9. Semrush Knowledge Base. AI Visibility Metrics. https://www.semrush.com/kb/1594-ai-seo-metrics AI Visibility 0-100, Mentions, Monthly Audience, Citations, Share of Voice та інші власні метрики. Використано для демонстрації відмінностей методологій.

10. Google Search Central Community. Generative AI report data / BigQuery export. https://support.google.com/webmasters/thread/464075295/google-search-console-data-under-the-generative-ai?hl=en Станом на 1 вересня 2026 року в спільноті підтверджувалося: окремого масового експорту цього звіту в BigQuery немає; доступний Export. Це джерело зі спільноти, а не формальна продуктова документація.

ГЛАВА 5

Виявлення URL: як пошукова система дізнається про сторінку

Посилання, sitemap, URL Inspection, ізольовані сторінки, robots.txt, редиректи та IndexNow. Де закінчується факт існування URL і починається реальне сканування.

Стан даних і джерел: 12 вересня 2026 року

У технічному SEO дуже легко перескочити через перший етап і одразу питати, чому сторінка не індексується. Але до індексації є простіше питання: звідки пошукова система взагалі знає, що цей URL існує? Google прямо називає цей етап виявленням URL, тобто виявлення URL. Централізованого реєстру всіх сторінок вебу немає. Пошукова система постійно додає нові адреси до набору вже відомих і лише після цього вирішує, коли їх сканувати.[1]

Це розділення практично важливе. Якщо URL не потрапив у відомий пошуковій системі набір, технічні деталі самої сторінки поки не мають значення: Googlebot ще не отримав HTML, не бачив title, canonical, noindex, текст або структуровані дані. Якщо URL уже відомий, але не просканований, ми маємо інший клас проблеми. А якщо просканований, але не індексується, це вже наступний етап системи.

Рис. 5.1. Виявлення URL передує скануванню. Знання адреси не означає, що пошукова система вже отримала або оцінила сторінку.

Схема автора на основі Google Search Central та Bing Webmaster Guidelines.

5.1. Виявлено не означає просканово

Офіційний опис Google формулює процес досить прямо: частина сторінок уже відома з попередніх відвідувань, інші URL знаходяться через посилання на відомих сторінках, а власник сайту може передати список адрес через sitemap.[1] Після виявлення URL Googlebot може відвідати його, але це окреме рішення системи. Сам Google також не гарантує, що сторінку буде просканово, проіндексовано або показано в результатах, навіть якщо вона відповідає технічним вимогам.[1]

Це добре видно у Search Console. Статус «Discovered - currently not indexed» означає, що Google знайшов URL, але ще не просканував його. У такому стані відсутність час останнього сканування не є загадкою: контент сторінки ще не був отриманий. Тому я ніколи не починаю діагностику такого статусу з переписування title або додавання schema. Спочатку треба зрозуміти, як URL потрапив у систему і чому не отримує наступний крок - сканування.[7]

НОТАТКА З ПРАКТИКИ Для одного URL статус «виявлено, але не просканово» може бути просто питанням часу. Для великого шаблону з тисячами сторінок це вже системний сигнал: архітектура, sitemap, обсяг URL, crawl demand або серверна місткість можуть впливати на те, які адреси система бере в роботу першими. Причину не можна визначити з одного ярлика Search Console.

5.2. Звідки пошукова система може дізнатися про URL

Для практичного аудиту я розділяю джерела на п’ять груп. Чотири з них універсальні для класичного вебу: внутрішні посилання, зовнішні посилання, sitemap і вже відомі пошуковій системі адреси. П’ята група - активні механізми повідомлення, наприклад IndexNow у Bing. Вони не скасовують граф посилань, але скорочують шлях від зміни на сайті до сигналу пошуковій системі.[10][11]

Джерело	Що реально повідомляє	Що не гарантує
Внутрішнє посилання	На доступній сторінці існує посилання на інший URL	Що ціль буде швидко просканована або індексована
Зовнішнє посилання	Інший сайт посилається на URL	Що URL стане канонічним або ранжуватиметься
Sitemap	Власник сайту заявляє набір бажаних URL	Що Google завантажить, просканує або проіндексує кожен URL
Раніше відомий URL	Адреса вже була в системі з попередніх обходів або історії	Що поточний контент за адресою вже повторно перевірений
IndexNow для Bing	Сайт активно повідомляє про додавання, зміну або видалення URL	Що Bing обов’язково просканує чи проіндексує сторінку

Таблиця 5.1. Канал виявлення повідомляє про існування URL. Він не дорівнює рішенню про сканування або індексацію.

5.3. Внутрішнє посилання - це не тільки PageRank

У SEO внутрішні посилання часто обговорюють через розподіл PageRank або тематичні зв’язки. Але їхня перша функція простіша: вони створюють шлях, по якому краулер знаходить інші URL. Google

прямо пише, що використовує посилання, зокрема, для пошуку нових сторінок для сканування.[2]

Тому «сторінка є в меню користувача» і «у HTML існує доступне для краулера посилання» - не завжди одне й те саме. Інтерфейс може виглядати як посилання, але технічно бути кнопкою, span з JavaScript-обробником або елементом, у якого URL з'являється тільки після взаємодії. Для Google найбезпечніший базовий варіант - HTML-елемент <a> з атрибутом href.[2]

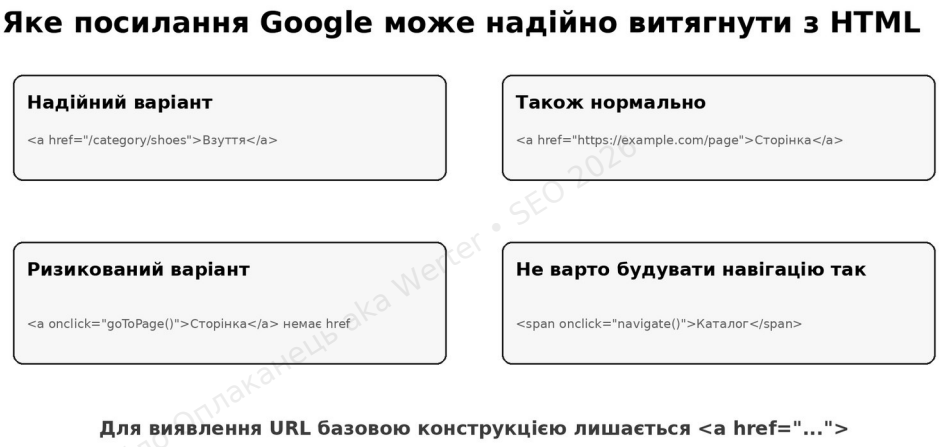


Рис. 5.2. Візуально однакові елементи можуть мати різну цінність для виявлення URL.

Приклади синтаксису адаптовані з Google Search Central, Link best practices.

HTML / поведінка	Надійність для виявлення	Коментар
<a href="/product/123">	Висока	Стандартний посилальний елемент з href.
<a href="https://example.com/page">	Висока	Повний абсолютний URL також нормально обробляється.
<a onclick="go()">	Низька	Немає href. Google не обіцяє надійно витягувати URL з такого елемента.
	Низька	Це не посилальний елемент. Не варто будувати критичну навігацію лише так.

HTML / поведінка	Надійність для виявлення	Коментар
URL, що з’являється тільки після дії користувача	Залежить від реалізації	Для критичного шляху відкриття сторінок краще мати звичайні посилання.

Таблиця 5.2. Для виявлення URL важлива не візуальна форма елемента, а те, що реально бачить і може розібрати краулер.

5.4. Мобільна версія може непомітно обрізати граф посилань

Після переходу Google на сканування з пріоритетом мобільної версії особливо небезпечна різниця між навігацією на десктопною і мобільною версіями. В офіційній документації Google прямо радить давати на мобільній версії той самий набір посилань, що й на десктопній. Якщо це неможливо, URL варто принаймні підтримати через sitemap; обмежений набір мобільних посилань може сповільнити виявлення нових сторінок.[5]

На практиці це типова проблема великих каталогів. На десктопній версії у категорії може бути повна пагінація, посилання на підкатегорії та фільтри, а мобільна версія показує тільки першу порцію карток і кнопку «показати ще». Якщо додаткові URL не існують у початковому HTML і не мають інших шляхів виявлення, ми створюємо різний граф сайту для користувача і для основного краулера Google.

ПОПЕРЕДЖЕННЯ Не робіть висновок «Google виконує JavaScript, отже будь-яка навігація через JavaScript рівноцінна HTML-посиланню». Виконання JavaScript і надійне виявлення всіх URL - різні задачі. Якщо URL критичний для органічного пошуку, звичайний <a href> лишається найпростішою страховкою.

5.5. Sitemap - це інвентар, а не заміна архітектури

Google визначає sitemap як спосіб повідомити про сторінки, які є новими або оновленими. Для добре пов’язаного невеликого сайту

sitemap може бути не критичним для самого виявлення: Google пише, що якщо всі важливі сторінки доступні через навігацію й внутрішні посилання, більшість сайту зазвичай можна знайти без нього. Водночас sitemap особливо корисний для великих сайтів, нових сайтів з малою кількістю зовнішніх посилань та спеціалізованого медіаконтенту.[3]

Тут важлива методологічна деталь. Sitemap не доводить, що сторінка добре інтегрована в сайт. Він лише додає ще один канал виявлення. URL може бути в sitemap і не мати жодного внутрішнього посилання. Пошукова система знатиме адресу, але з погляду структури сайту це все одно ізольована сторінка. Для комерційної або інформаційної сторінки, яку ми хочемо ранжувати, це зазвичай слабка конструкція.

Рис. 5.3. Технічні межі sitemap не є лімітом індексації.

Google Search Central: до 50 000 URL або 50 MB без стиснення в одному sitemap; sitemap index може містити до 50 000 loc.[4][12]

Правило sitemap	Що робити на практиці
До 50 000 URL або 50 MB без стиснення	Ділити більші набори на окремі sitemap.
Використовувати повні абсолютні URL	Не передавати відносні /page або інші неоднозначні адреси.
Включати URL, які ви хочете бачити в Пошуку	Не засмічувати sitemap редиректами, 404 та технічними варіантами.

Правило sitemap	Що робити на практиці
Переважно вказувати канонічні URL	Sitemap є одним із сигналів бажаної канонічної версії, але не є наказом.
Sitemap - підказка	Наявність URL у файлі не гарантує сканування або індексації.
Порядок URL у sitemap не має значення для Google	Не витрачати час на штучне сортування за «пріоритетом».

Таблиця 5.3. Sitemap має бути чистим джерелом URL, а не складом усіх адрес, які здатен згенерувати сайт.

5.6. lastmod працює тільки тоді, коли йому можна довіряти

Google окремо пише, що використовує значення <lastmod>, якщо воно стабільно й перевірно відображає суттєву зміну сторінки. Зміна основного контенту, структурованих даних або посилань може бути суттєвою. Зміна дати copyright - ні.[4]

Тому шаблон «щодня ставити поточну дату всім URL» не прискорює систему магічно. Він знижує інформаційну цінність сигналу. Для великих сайтів я краще мати 20% URL з чесним lastmod, ніж 100% з датою, яка змінюється кожну ніч без реальної зміни сторінки.

МІФ «Якщо щодня пересабмітити sitemap, Google швидше обійде весь сайт». Google прямо радить не надсилати той самий незмінений sitemap багато разів на день і не очікувати, що всі URL у sitemap будуть просканові негайно.[5]

5.7. Ізольована сторінка - це визначення з вашого набору даних, а не абсолютний факт

Термін «ізольована сторінка» часто використовують занадто впевнено. Краулер Screaming Frog або будь-який інший інструмент не знає всіх шляхів, якими Google міг колись знайти URL. Якщо сторінка не має внутрішніх вхідних посилань у вашому скануванні сайту, вона

може бути в sitemap, мати зовнішнє посилання, бути відомою з минулої версії сайту або вже бути в індексі.

Тому я використовую точніші класи. «Немає внутрішніх вхідних посилань» - факт нашого crawl. «Є тільки в sitemap» - факт по двох джерелах. «Не знайдено внутрішніх посилань, sitemap і запитів Googlebot за обраний період» - сильніший сигнал, але теж не доказ, що Google ніколи не знав URL. Такі формулювання роблять аудит набагато чеснішим.

Практична класифікація URL за трьома джерелами даних

Внутрішнє посилання	Sitemap	Запит Googlebot	Практичний висновок
Так	Так	Так	Нормально інтегрований URL
Ні	Так	Так	URL відомий через sitemap, але внутрішня архітектура слабка
Так	Ні	Так	Виявлення через граф посилань; перевірити політику sitemap
Ні	Ні	Так	URL відомий з іншого джерела або з історії
Ні	Так	Ні	URL відомий через sitemap, але ще не просканований
Ні	Ні	Ні	У вашому наборі даних URL фактично невидимий

Рис. 5.4. Практична класифікація URL за внутрішнім графом, sitemap і серверними логами.

Це робоча модель аудиту, а не внутрішня класифікація Google.

Клас URL	Інтерпретація	Типова дія
Внутрішні посилання + sitemap	Нормально інтегрований у дві системи виявлення	Перевіряти вже наступні етапи: сканування, canonical, індексація.
Лише sitemap	Адреса передана напряму, але не інтегрована у внутрішній граф	Зрозуміти, чи сторінка справді заслуговує на індексацію. Якщо так - додати логічні внутрішні посилання.
Лише внутрішні посилання	Сторінка доступна через граф, але відсутня у sitemap	Не завжди помилка. Перевірити політику sitemap і канонічність.
Лише логи	Googlebot уже запитував URL, але поточне сканування сайту і sitemap його не показують	Перевірити історичні URL, редиректи, зовнішні посилання, параметри.

Клас URL	Інтерпретація	Типова дія
Ніде у вибірці	Немає доказу доступного шляху в поточних даних	Шукати джерело генерації URL або додати потрібний шлях виявлення.

Таблиця 5.4. Термін «ізольована сторінка» краще замінити конкретним описом того, в яких джерелах URL є або немає.

5.8. URL Inspection дає підказки про джерело виявлення, але не повний журнал

У Search Console блок Discovery може показати sitemap і Referring page. Referring page - це сторінка, яку Google, імовірно, використав для виявлення URL. Вона може посилатися прямо або бути вищою ланкою ланцюга. Якщо поле порожнє, це не означає, що сторінки-джерела не існує. Сам Google попереджає, що інформація може бути недоступна в інструменті й URL міг бути знайдений іншим способом.[8]

Те саме видно в API URL Inspection. Поле sitemap[] містить відомі Google sitemap, де присутній URL, але документація прямо каже, що список не гарантовано вичерпний. Поле referringUrls[] повертає URL, які посилаються на перевірену адресу прямо або опосередковано.[9] Це корисно для автоматизації, але не перетворює API на повний лог discovery.

Рис. 5.5. Search Console показує окремі сигнали виявлення, а не повну історію того, звідки Google коли-небудь знав URL.

Офіційні поля: Sitemaps і Referring page; URL Inspection API: sitemap[] та referringUrls[].[8][9]

5.9. robots.txt не ховає сам факт існування URL

robots.txt контролює доступ краулера, а не знання про адресу. Google прямо попереджає: якщо інші сторінки посилаються на URL, заблокований у robots.txt, Google все одно може знати адресу і навіть показати її в результатах без опису, не завантажуючи сам контент.[6]

Звідси випливає важлива технічна помилка: не можна одночасно блокувати сторінку в robots.txt і очікувати, що Google гарантовано прочитає meta robots noindex у її HTML. Щоб побачити noindex, сторінку треба отримати. Те саме стосується canonical у HTML: він стає доступним лише після завантаження й обробки документа.

Рис. 5.6. robots.txt, noindex і canonical вирішують різні задачі. Жоден із них не є універсальною кнопкою «не знати цей URL».

Схема спрощує механіку для діагностики. Детальні правила індексації та канонікалізації розбираються в окремих главах.

Механізм	Впливає на виявлення URL	Впливає на сканування	Впливає на індексацію
robots.txt Disallow	Не гарантує приховування	Так, блокує запит відповідного краулера	Опосередковано; URL може лишатися відомим і з'явитися без контенту
meta robots noindex	Ні	Ні, сторінку треба отримати, щоб прочитати директиву	Так, якщо Google може прочитати директиву
X-Robots-Tag: noindex	Ні	Ні, потрібна HTTP-відповідь	Так, після отримання заголовка
rel=canonical	Ні	Ні	Сигнал вибору канонічної версії, не абсолютна команда
видалення внутрішніх посилань	Зменшує один із шляхів	Може вплинути на майбутню частоту через зміну графа	Саме по собі не видаляє URL з індексу

Таблиця 5.5. Різні директиви працюють на різних етапах. Змішування виявлення URL, сканування та керування індексацією створює більшість технічних помилок.

5.10. Редирект може привести краулер до нової адреси, але це вже наступний крок після виявлення старої

Якщо Googlebot запитує відомий URL і отримує 301, 308, 302 або інший підтримуваний редирект, він переходить до цільової адреси. Для постійних серверних редиректів Google також використовує ціль як сильний сигнал канонічної версії.[13]

Це важлива різниця для міграцій. Карта редиректів не є заміною внутрішніх посилань на нові URL і sitemap з новими URL. Редирект працює тільки після того, як стару адресу вже запитали. Якщо новий розділ має бути частиною поточної архітектури, я хочу, щоб він був доступний напряму через внутрішній граф і sitemap, а не тільки як кінцева точка старих 301.

5.11. Google і Bing мають різні способи отримати сигнал про новий URL

Google у публічній документації робить акцент на посиланнях, sitemap і запиті на повторне сканування через URL Inspection для невеликої кількості URL.[1][8][14] Bing, крім класичних посилань і sitemap, підтримує IndexNow - протокол активного повідомлення про додані, змінені або видалені URL.[10][11]

Bing прямо показує в IndexNow Insights надіслані URL, час подання, джерело, а для вибірки - статус сканування, статус індексації та час першої індексації. При цьому Microsoft так само застерігає: подання URL через IndexNow не гарантує сканування або індексацію.[11]

Рис. 5.7. Класичне виявлення через граф і sitemap та активне повідомлення IndexNow вирішують одну задачу різними шляхами.

Google не вказаний у книзі як учасник IndexNow. Для Google використовуються його офіційно задокументовані механізми; IndexNow тут показаний як окремий механізм Bing.

5.12. Серверний лог доводить сканування, але не момент виявлення

Логи дуже сильні як факт того, що конкретний агент користувача зробив HTTP-запит до конкретного URL у конкретний час. Але лог нічого не каже про те, скільки часу URL уже був відомий системі до першого запиту. Якщо Googlebot уперше звернувся 12 вересня о 03:14, це не означає, що URL був виявлений саме о 03:14.

Тому в аудиті я не використовую «перший зафіксований запит Googlebot» як синонім «дата виявлення». Це перша зафіксована нами дата сканування в наявному періоді логів. Виявлення могло відбутися раніше через sitemap, посилання або інше джерело. Search Console також може показувати referring page або sitemap, але не гарантує повноту цієї історії.[8][9]

5.13. Метрики виявлення, які реально можна порахувати на своєму сайті

Google не публікує універсальну норму на кшталт «95% сторінок повинні бути знайдені через внутрішні посилання». Тому я не використовую вигадані галузеву норму. Замість цього рахую власні стабільні метрики по шаблонах і дивлюся на зміну в часі.

```
Покриття внутрішніми посиланнями = цільові URL, дозволені до індексації, з >=1 доступним внутрішнім посиланням / усі цільові URL, дозволені до індексації
```

Формула 5.1. Покриття внутрішнім графом.

Мета - не досягти магічних 100%, а знайти шаблони, де важливі URL існують тільки поза внутрішньою архітектурою.

```
Частка URL лише в sitemap = URL у sitemap без доступних внутрішніх посилань / усі URL у sitemap
```

Формула 5.2. Частка URL, які тримає тільки sitemap.

Розраховуйте окремо за типами сторінок. Один загальний відсоток може приховати проблему в конкретному шаблоні.

```
Покриття sitemap = цільові канонічні URL у sitemap / усі цільові канонічні URL
```

Формула 5.3. Покриття sitemap цільовими URL.

Метрика	Знаменник	Що діагностує	Чого не доводить
Покриття внутрішніми посиланнями	Цільові URL, дозволені до індексації	Чи є шлях у внутрішньому графі	Що Google вже пройшов цим шляхом
Частка URL лише в sitemap	URL у sitemap	Наскільки sitemap компенсує слабку архітектуру	Що сторінки погані або не індексуються
Покриття sitemap	Цільові канонічні URL	Чистоту інвентарю бажаних URL	Що sitemap потрібен кожній сторінці для виявлення
URL з першим зафіксованим запитом Googlebot	URL у логах за період	Факт першого зафіксованого запиту в логах	Справжню дату первинного виявлення
«Виявлено, але не проіндексовано»	URL у відповідному статусі GSC	Google знає URL, але ще не просканував	Конкретну причину затримки для кожного URL

Таблиця 5.6. Метрика повинна мати чіткий знаменник і не претендувати на те, чого джерело даних не бачить.

5.14. Мій порядок аудиту, якщо нові URL “не підхоплюються”

Я починаю не з кнопки «Запит на індексування», а з вибірки URL одного типу. Наприклад, 500 нових товарів, 300 масово згенерованих посадкових сторінок або 2 000 нових профілів. Потім накладаю чотири джерела: внутрішній crawl, sitemap, Search Console URL Inspection для вибірки та серверні логи Googlebot. Це дає значно більше інформації, ніж ручна перевірка десяти випадкових сторінок.

Крок	Питання	Дані	Рішення
1. Внутрішній граф	Чи є хоча б один доступне внутрішнє посилання?	Краулер сайту / база посилань	Якщо ні - виправити архітектуру або визнати URL нецільовим.
2. Sitemap	Чи є цільовий канонічний URL у правильному sitemap?	XML/RSS/текстовий sitemap	Прибрати сміття, додати цільові URL, перевірити lastmod.
3. URL Inspection	Що Google показує в розділі Discovery?	Sitemaps, Referring page, статус покриття	Не трактувати відсутність поля як доказ відсутності джерела.
4. Логи	Чи був реальний запит Googlebot?	Серверні журнали доступу	Розділити «відомий, але не просканований» і «просканований».
5. Масштаб шаблону	Проблема одинична чи системна?	Групування за типом сторінки / директорією	Пріоритизувати шаблон, а не ремонтувати URL по одному.
6. Наступний етап	Що відбувається після сканування?	HTTP, рендеринг, canonical, статус індексації	Переходити до наступних глав, якщо виявлення URL вже не проблема.

Таблиця 5.7. Аудит виявлення URL має рухатися від архітектури до фактичного запиту Googlebot, а не навпаки.

5.15. Простий Python-аудит: внутрішні посилання + sitemap + логи

Нижче - не “магічний SEO-скрипт”, а мінімальна модель злиття трьох джерел. На вході потрібні три CSV: список цільових URL з кількістю внутрішніх вхідних посилань, список URL із sitemap і

список URL, які запитував підтверджений Googlebot за обраний період. Назви колонок легко адаптувати під ваш експорт.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ

Я завантажу три CSV: internal.csv з колонками url та inlinks, sitemap.csv з url і googlebot.csv з url та first_seen. Спочатку перевір структуру файлів. Побудуй повний список URL з усіх трьох джерел і для кожного визнач: чи є внутрішнє вхідне посилання, чи є URL у sitemap, чи бачив його Googlebot. Класифікуй URL за цими трьома сигналами, порахуй кількість у кожному класі й покажи по 20 найважливіших проблемних URL.

АЛЬТЕРНАТИВА БЕЗ ШІ

У Sheets/Excel створіть master-список URL і три колонки-прапорці через XLOOKUP/COUNTIF: внутрішні посилання, sitemap, логи. Потім зробіть зведену таблицю за комбінаціями TRUE/FALSE. Python потрібен лише коли обсяг або регулярність уже незручні для таблиці.

QR: КОД

Скануйте камерою: QR містить компактну виконувану версію коду без коментарів. Коментований код наведено нижче.

ДЛЯ АВТОМАТИЗАЦІЇ: PYTHON ДЛЯ ЗЛИТТЯ З ДЖЕРЕЛ

```
import pandas as pd

# Очікувані файли:
# internal.csv: url, inlinks
# sitemap.csv: url
# googlebot.csv: url, first_seen
internal = pd.read_csv("internal.csv")
sitemap = pd.read_csv("sitemap.csv")
logs = pd.read_csv("googlebot.csv")

# 1) Створюємо по одному булевому сигналу для кожного джерела.
internal["has_internal_link"] = internal["inlinks"].fillna(0).gt(0)
sitemap["in_sitemap"] = True
logs["seen_googlebot"] = True

# 2) Робимо outer join, щоб не втратити URL,
# які присутні тільки в одному або двох джерелах.
df = (
    internal[["url", "has_internal_link"]]
    .merge(sitemap[["url", "in_sitemap"]], on="url", how="outer")
    .merge(logs[["url", "seen_googlebot", "first_seen"]], on="url", how="outer")
)

# 3) Відсутній сигнал означає False, а не "невідомо".
```

```
for col in ["has_internal_link", "in_sitemap", "seen_googlebot"]:
    df[col] = df[col].fillna(False)

# 4) Класифікуємо URL за комбінацією трьох сигналів.
def classify(r):
    if r.has_internal_link and r.in_sitemap and r.seen_googlebot:
        return "integrated"
    if (not r.has_internal_link) and r.in_sitemap and r.seen_googlebot:
        return "sitemap_only_but_crawled"
    if (not r.has_internal_link) and r.in_sitemap and (not r.seen_googlebot):
        return "sitemap_only_not_crawled"
    if r.has_internal_link and (not r.in_sitemap) and r.seen_googlebot:
        return "internal_only_but_crawled"
    if (not r.has_internal_link) and (not r.in_sitemap) and r.seen_googlebot:
        return "known_from_other_source_or_history"
    return "no_evidence_in_current_dataset"

# 5) Додаємо клас до кожного URL і рахуємо розподіл.
df["discovery_class"] = df.apply(classify, axis=1)
print(df["discovery_class"].value_counts(dropna=False))

# За потреби збережіть результат для ручної перевірки:
# df.to_csv("discovery_audit.csv", index=False)
```

Код 5.1. Мінімальна класифікація URL за внутрішніми посиланнями, sitemap і серверними логами.

У реальній системі я додаю тип сторінки, цільовий canonical, HTTP-статус, можливість індексації, дату публікації і кілька часових полів. Тоді можна відповісти не «скільки ізольованих сторінок», а «який відсоток нових товарів категорії X через 48 годин має внутрішнє вхідне посилання, присутній у sitemap і вже отримав перший запит Googlebot». Це вже операційна метрика, яку можна порівнювати між релізами.

5.16. Що робити з “Discovered - currently not indexed”

Найгірша реакція - масово натискати «Запит на індексування». Для великої вибірки це не усуває системну причину. Сам Google радить URL Inspection для невеликої кількості URL, а для великих наборів - sitemap. Повторний запит одного й того самого URL не змусить його скануватися швидше.[14]

Я спочатку сегментую проблему. Якщо статус має 12 URL з 50 000 - це одна історія. Якщо 70% нового шаблону через два тижні лишаються у статусі «виявлено», це інша. Далі перевіряю, чи вони доступні через нормальні внутрішні посилання, чи sitemap містить

саме канонічні 200 URL, чи не створено мільйони альтернативних адрес параметрами, чи сервер стабільно віддає відповіді, і чи Googlebot взагалі доходить до цього шаблону.

Важливо не видавати пояснення з довідковий матеріал спільноти за гарантовану причину конкретного сайту. Офіційне значення статусу просте: URL знайдено, але не просканово. Чому саме конкретна адреса довго не отримує сканування - це вже діагностика системи, а не текст одного статусу.[7]

5.17. Що ми точно знаємо

Ми точно знаємо, що Google відокремлює виявлення URL від сканування.[1] Знаємо, що нові URL можуть бути знайдені через посилання і sitemap.[1][2][3] Знаємо, що стандартне <a href> є базовою надійною конструкцією для посилань, які Google може розібрати.[2] Знаємо жорсткі межі sitemap: 50 000 URL або 50 MB без стиснення на файл, а sitemap index може містити до 50 000 записів loc.[4][12]

Ми також знаємо, що sitemap є підказкою, а не гарантією сканування чи індексації.[4] robots.txt не є механізмом приховування самого URL з Google, якщо адресу можна знайти з інших місць.[6] URL Inspection може показувати sitemap і сторінку-джерело, але ці дані не є повним журналом усіх шляхів виявлення.[8][9] Bing окремо підтримує IndexNow як активний канал повідомлення про змінені URL, не гарантуючи при цьому індексацію.[10][11]

Чого ми не знаємо з публічної документації: точну внутрішню вагу кожного каналу виявлення, універсальний час від виявлення до сканування, «ідеальну» кількість внутрішніх вхідних посилань для нового URL або загальний відсоток сторінок, відомих лише через sitemap, після якого сайт обов’язково матиме проблему. Якщо хтось називає такі числа без власного контрольованого набору даних і методології, це вже не факт Google, а гіпотеза.

Головний практичний висновок цієї глави простий. До того як аналізувати рендеринг, canonical або якість контенту, треба довести, що потрібний URL узагалі існує для пошукової системи як адреса і має нормальний шлях до черги сканування. Наступна глава

починається саме там: URL уже відомий. Тепер питання - коли, як часто і з яким пріоритетом його сканувати Googlebot.

Джерела до глави

1. Google Search Central. In-Depth Guide to How Google Search Works. https://developers.google.com/search/docs/fundamentals/how-search-works Офіційне розділення URL discovery, crawling та indexing; посилання й sitemap як джерела нових URL.

2. Google Search Central. Link best practices for Google. https://developers.google.com/search/docs/crawling-indexing/links-crawlable Google використовує посилання для пошуку нових сторінок; <a href> як базова доступна для краулера конструкція.

3. Google Search Central. What Is a Sitemap. https://developers.google.com/search/docs/crawling-indexing/sitemaps/overview Коли sitemap особливо корисний; Google зазвичай може знайти добре пов'язані сторінки через внутрішні посилання.

4. Google Search Central. Build and Submit a Sitemap. https://developers.google.com/search/docs/crawling-indexing/sitemaps/build-sitemap 50 000 URL / 50 MB; абсолютні URL; sitemap як підказка; точний lastmod; Google ігнорує priority та changefreq.

5. Google Search Central. Troubleshoot Google Search Crawling Errors. https://developers.google.com/search/docs/crawling-indexing/troubleshoot-crawling-errors Поради щодо доступних для краулера посилань, мобільної версії, sitemap і логів.

6. Google Search Central. Introduction to robots.txt. https://developers.google.com/search/docs/crawling-indexing/robots/intro robots.txt керує доступом краулера, але URL може лишатися відомим і навіть з'явитися в результатах без контенту.

7. Google Search Console Help. Page indexing report / Discovered - currently not indexed. https://support.google.com/webmasters/answer/7440203?hl=en Статус означає, що сторінку знайдено, але ще не просканово.

8. Google Search Console Help. URL Inspection tool. https://support.google.com/webmasters/answer/9012289?hl=en Discovery, Sitemaps, Referring page і застереження щодо неповноти цих полів.

9. Google Search Console API. UrlInspectionResult. https://developers.google.com/webmaster-tools/v1/urlInspection.index/UrlInspectionResult Поля sitemap[] і referringUrls[]; lastCrawlTime та інші дані індексної версії URL.

10. Bing Webmaster Guidelines. https://www.bing.com/webmasters/help/webmaster-guidelines-30fba23a Bing прямо називає IndexNow, XML sitemap, внутрішні та зовнішні посилання каналами discovery.

11. Bing Webmaster Tools. IndexNow.
https://www.bing.com/webmasters/help/indexnow-0z209wby Активне повідомлення про нові, змінені або видалені URL; submission не гарантує indexing.

12. Google Search Central. Manage Your Sitemaps With Sitemap Index Files.
https://developers.google.com/search/docs/crawling-indexing/sitemaps/large-sitemaps До 50 000 loc у sitemap index; до 500 sitemap index files на сайт у Search Console.

13. Google Search Central. Redirects and Google Search.
https://developers.google.com/search/docs/crawling-indexing/301-redirects Googlebot переходить за редиректами; постійний редирект є сигналом канонічної цілі.

14. Google Search Central. Ask Google to Recrawl Your Website.
https://developers.google.com/search/docs/crawling-indexing/ask-google-to-recrawl URL Inspection для кількох URL, sitemap для великих наборів; повторні запити не пришвидшують crawl гарантовано.

ГЛАВА 6

Сканування: черга, частота, ресурси й пріоритети

Що насправді означає бюджет сканування, коли він стає проблемою, як Googlebot реагує на сервер, що показує Crawl Stats і як перейти від агрегатів Search Console до подієвого аналізу логів.

Стан даних і джерел: 12 вересня 2026 року

Бюджет сканування часто обговорюють так, наче кожен сайт має фіксований денний ліміт Googlebot, який можна “витратити” на хороші або погані сторінки. Це зручна метафора, але технічно вона неточна. У документації Google від 22 липня 2026 року модель описана інакше: фактичний обсяг сканування формується перетином двох речей - межі спроможності сканування хоста та попиту на сканування конкретних URL.[1]

Перша величина відповідає на питання “скільки Google може забрати, не створюючи проблем серверу”. Друга - “скільки Google взагалі хоче забрати”. Якщо сервер швидкий і стабільний, але попит низький, Google не зобов’язаний збільшувати частоту. Якщо попит високий, але хост починає відповідати повільніше, давати 5xx або 429, система знижує навантаження. Тому бюджет сканування - не одна ручка і не один лічильник.

Схема автора за Google Crawling Infrastructure, 2026

Рис. 6.1. Модель бюджету сканування у Google: фактичне сканування виникає на перетині спроможності хоста і попиту на сканування.

Схема автора за Google Crawling Infrastructure. Це концептуальна модель, а не формула множення двох показників.

6.1. Коли бюджет сканування узагалі вартий окремої уваги

Для більшості невеликих сайтів бюджет сканування не є окремою SEO-проблемою. Google прямо пише: якщо сторінок небагато, вони не змінюються дуже швидко або нові матеріали скануються того ж дня, достатньо підтримувати sitemap і регулярно перевіряти звіт Page Indexing. Окремий розширений посібник Google орієнтує насамперед на великі та дуже динамічні сайти.[1]

Офіційні орієнтири приблизні, а не жорсткі пороги: 1 млн і більше унікальних URL із помірними змінами приблизно раз на тиждень; 10 тис. і більше унікальних URL із дуже швидкими щоденними змінами; або сайти, де значна частина URL у Search Console має статус «Виявлено, але наразі не проіндексовано».[1] Сам Google окремо попереджає, що ці числа потрібні для класифікації, а не для математичного рішення “9 999 URL - проблеми немає, 10 000 - є”.

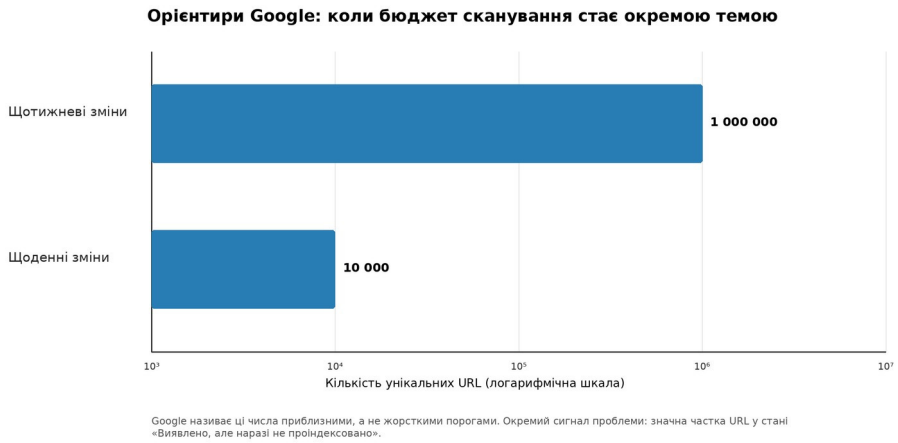


Рис. 6.2. Орієнтири Google для сайтів, де оптимізація бюджету сканування стає окремою технічною задачею.

Джерело: Google Crawling Infrastructure, 2026. Обидва пороги Google називає приблизними.

Ситуація	Чи починати з бюджету сканування	Що перевірити першим
800 статичних сторінок, нові URL швидко скануються	Зазвичай ні	звіт Page Indexing, sitemap, внутрішні посилання, технічну доступність.
50 тис. товарів, ціни й наявність змінюються щодня	Так, уже має сенс	Логи, Crawl Stats, частоту повторного сканування, 304, час відповіді, зайві параметричні URL.
2 млн URL каталогу, частина шаблонів майже не змінюється	Так	Інвентар URL, дублікати, фасети, soft 404, попит на повторне сканування.
20 тис. сторінок, але тисячі «Виявлено, але наразі не проіндексовано»	Так, як одна з гіпотез	Чи проблема у виявленні нових URL, попиті, якості, інвентарі URL або місткості хоста.
Сайт на 5 тис. URL після міграції	Тимчасово може бути важливо	Редиректи, sitemap, старі/нові URL, статуси, логи та хвилю повторного сканування.

Таблиця 6.1. Бюджет сканування - не універсальна причина повільної індексації. Спочатку визначаємо, чи масштаб і динаміка сайту взагалі роблять його релевантним.

6.2. Межа спроможності: Google дивиться не тільки на кількість запитів

Google визначає межу спроможності сканування через навантаження на хост. Враховується не лише кількість паралельних з'єднань, а й те, скільки часу сервер тримає їх відкритими. Якщо час відповіді та затримка стабільні або покращуються, межа може зрости. Якщо затримка або TTFB погіршуються, з'являються 5xx чи 429, межа знижується.[1]

Це одна з причин, чому я не використовую "запитів за секунду" як самодостатній KPI. Два сервери можуть обробляти однакові 5 запитів на секунду зовсім по-різному. Один віддає HTML за 80 мс і майже не помічає Googlebot. Інший тримає з'єднання 2 секунди, упирається в CPU або базу і починає деградувати. Для пошукового робота це різні хости, навіть якщо число запитів однакове.

ДАНІ Google не публікує універсальну "нормальну швидкість Googlebot" для SEO-аудиту. Документація пояснює адаптивну модель: якщо сайт здоровий і є попит, система може використати більше з'єднань; якщо сервер сповільнюється або повертає 5xx/429, сканування зменшується.[1][5]

Сигнал з боку хоста	Що бачить Google	Очікувана реакція
Стабільний або кращий час відповіді	Хост справляється з навантаженням	Межа спроможності може зрости, якщо є попит.
Зростає затримка / TTFB	Кожне з'єднання займає більше часу	Google може знизити інтенсивність сканування.
429 Too Many Requests	Сервер прямо сигналізує перевантаження	Google сповільнює сканування.
5xx на значній кількості URL	Серверна помилка / недоступність	Google тимчасово зменшує частоту сканування.
404 / 410 для видалених URL	Контент відсутній постійно	Частота повторних запитів поступово падає; це не обмеження частоти.
304 Not Modified	Контент не змінився	Google може повторно використати попередню версію без повторної передачі контенту.

Таблиця 6.2. HTTP-відповідь впливає не лише на індексацію, а й на економіку повторного сканування.

6.3. Попит на сканування: швидкий сервер ще не означає, що Google хоче більше URL

Другий бік моделі - попит на сканування. Для Googlebot він залежить від розміру сайту, частоти оновлення, якості та релевантності сторінок відносно інших сайтів. У поточній документації Google окремо виділяє відомий інвентар URL, популярність і застарівання. Найбільш керований фактор для власника - інвентар URL: якщо система знає мільйони дублів, варіантів сортування, застарілих адрес або інших URL, які не потрібні в пошуку, частина ресурсу йде на них.[1]

Популярність тут не треба перетворювати на примітив “більше зворотних посилань = більше сканування”. Google формулює обережніше: популярні URL в інтернеті мають тенденцію скануватися частіше, щоб зберігатися свіжими. Застарівання означає, що система намагається не тримати документи застарілими. Міграція сайту або інша подія масштабу всього сайту також може тимчасово підняти попит на повторне сканування.[1]

Фактор попиту	Що він означає практично	Що може зробити SEO / розробка
Інвентар URL	Скільки адрес Google знає і вважає потенційними кандидатами	Прибрати нескінченні простори, консолідувати дублікати, чистити sitemap, віддавати 404/410 там, де URL справді видалені.
Частота змін	Наскільки швидко застаріває документ	Коректний lastmod, кешування, окремий контроль динамічних шаблонів.
Якість і релевантність	Наскільки сторінки варті повторної обробки для продукту Google	Не масштабувати слабкі URL тільки тому, що технічно можна їх згенерувати.
Популярність	Наскільки URL помітний у вебі	Не використовувати як штучний KPI бюджету сканування; аналізувати як частину загальної цінності сторінок.
Події сайту	Міграція, масова зміна URL або контенту	Очікувати тимчасової хвилі повторного сканування і контролювати серверну місткість.

Таблиця 6.3. Попит на сканування - це не черга, яку можна вручну розставити по позиціях. Ми керуємо інвентарем і сигналами, а не внутрішнім планувальником Google.

6.4. Crawl Stats: що Search Console реально показує

Звіт Crawl Stats - найкраща перша точка для швидкого макроаналізу. Він показує загальну кількість запитів сканування, завантажений обсяг, середній час відповіді, стан хоста, розподіл за HTTP-відповідями, типами файлів, метою сканування і типом Googlebot.[2] Для властивості домену можна також дивитися хости, але інтерфейс показує тільки 20 дочірніх доменів, які отримували трафік сканування за останні 90 днів.[2]

Звіт має важливе обмеження: список прикладів URL не є повним логом. Відсутність конкретного URL у прикладах не доводить, що Google його не запитував. Сам Google описує ці URL як репрезентативну вибірку. Тому Crawl Stats добре відповідає на “що відбувається з хостом у цілому”, але слабше - на “що сталося з кожним URL”.[2]

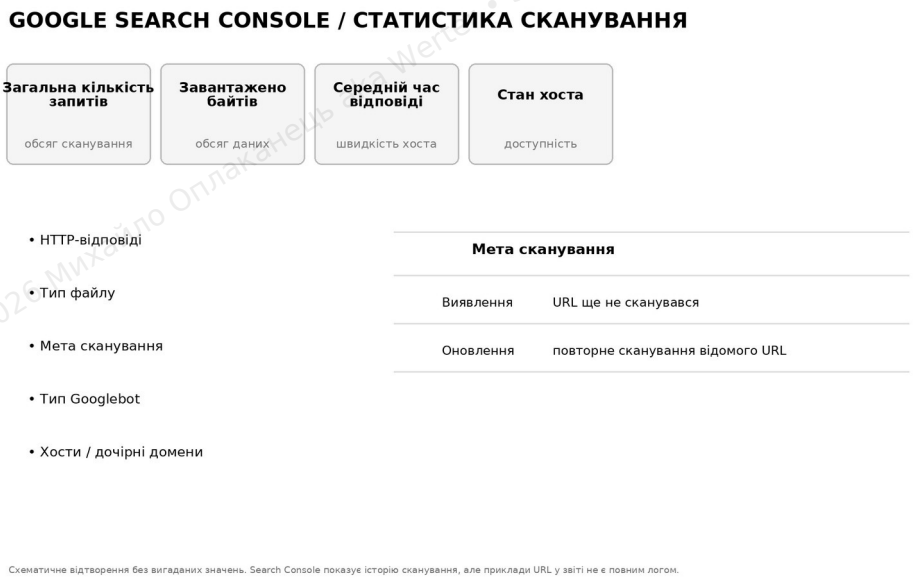


Рис. 6.3. Які зрізи дає Crawl Stats у Search Console. Значення на схемі навмисно не вигадані.

Схематичне відтворення офіційного звіту. Search Console окремо показує мету сканування: виявлення URL, який Google раніше не сканував, та оновлення - повторне сканування відомого URL.

Поле Crawl Stats	Що можна встановити	Чого не можна довести
Загальна кількість запитів сканування	Обсяг запитів Google до URL і ресурсів вашого хоста	Скільки унікальних сторінок було повністю переоцінено.
Завантажений обсяг	Скільки байтів Google завантажив за період	Що більший обсяг сам по собі є проблемою.
Середній час відповіді	Середній час відповіді для запитаних ресурсів	TTFB конкретного важливого шаблону без сегментації.
Стан хоста	Чи були значні проблеми robots.txt, DNS або мережевого з'єднання	Корінну причину бізнесового падіння без додаткової діагностики.
Мета сканування	Discovery або Refresh	Повний пріоритет URL усередині планувальника.
Тип Googlebot	Smartphone, Desktop, Image, Video, ресурси сторінки тощо	Що кожен запит мав однакову SEO-цінність.
Приклади URL	Приклади запитів у вибраному сегменті	Повний список усіх запитів Googlebot.

Таблиця 6.4. Search Console дає агрегований стан системи; серверний лог потрібен для подієвого рівня.

6.5. Discovery і Refresh: виявлення нових URL та повторне сканування

У Crawl Stats Google розділяє мету сканування на Discovery і Refresh. Discovery означає, що URL раніше не сканувався Google. Refresh - повторний запит уже відомої сторінки.[2] Це корисне розділення для релізів. Якщо ви щойно опублікували великий набір нових сторінок або подали sitemap, логічно очікувати підйом сканування нових URL. Якщо проблема в тому, що ціни, наявність товарів або новини застарівають у видачі, дивитися треба не лише на виявлення нових URL, а й на частоту повторного сканування.

Я не ставлю KPI "Discovery має бути X% від усіх запитів". Такий відсоток без контексту безглуздий. Для новинного сайту, каталогу після великого імпорту й зрілого енциклопедичного сайту нормальний профіль буде різним. Правильне питання - чи відповідає частота сканування бізнесовій швидкості змін конкретного типу сторінок.

6.6. Серверні логи: коли агрегатів уже мало

Як тільки питання переходить на рівень “які саме URL Googlebot сканує, як часто й з якими статусами”, я переходжу до серверних логів. У них є те, чого немає в Search Console: конкретний час, конкретний URL, HTTP-код, обсяг відповіді, User-Agent, IP і, якщо інфраструктура логгує, час відповіді. Це дозволяє групувати запити за шаблонами, параметрами, статусами та часом.

User-Agent недостатньо для верифікації. Google прямо попереджає, що його можна підробити. Для важливого аудиту потрібно перевіряти IP через зворотний DNS-запит або зіставляти з офіційними діапазонами Googlebot.[3] У книзі далі я використовую термін “Googlebot у логах” тільки для трафіку, який уже пройшов таку перевірку.

Рис. 6.4. Робочий шлях від сирого серверного логу до SEO-рішення.

Лог без класифікації дає мільйони рядків. Цінність з’являється після верифікації бота, нормалізації URL і прив’язки до типів сторінок.

Поле логу	Навіщо воно SEO	Типова помилка
Час (timestamp)	Рахувати перше сканування, інтервал повторного сканування, піки й затримки	Змішувати часові зони або округлювати до дня надто рано.
URL / path / query	Будувати шаблони та знаходити параметричні простори	Аналізувати кожен URL окремо й втрачати структуру.

Поле логу	Навіщо воно SEO	Типова помилка
HTTP-код (status)	Розділяти 2xx, редиректи, 404/410, 429, 5xx	Дивитися лише на кількість 200.
Обсяг відповіді (bytes sent)	Оцінювати мережевий трафік і важкі відповіді	Плутати стиснені й незжаті байти різних рівнів логування.
час відповіді	Бачити деградацію саме на Googlebot запити	Порівнювати різні сервери без нормалізації.
User-Agent + IP	Верифікувати тип пошукового робота	Довіряти тільки рядку Googlebot у UA.
Джерело переходу (referrer)	Іноді корисно для інших ботів/запитів	Очікувати, що Googlebot завжди передасть джерело виявлення нових URL.

Таблиця 6.5. Мінімальний набір полів для аналізу сканування з серверних логів.

6.7. Метрики, які я рахую з логів

Універсальної “правильної частоти сканування” немає, тому я не порівнюю сайт з абстрактним галузевим відсотком. Я будую базову лінію за типами сторінок, а потім дивлюся, як вона змінюється після релізів, міграцій, перебудови фасетів або оптимізації сервера.

Частка корисного сканування = запити до цільових canonical 2xx HTML / усі підтверджені запити Googlebot

Формула 6.1. Частка корисного сканування.

Це внутрішня операційна метрика. “Цільовий canonical” визначається вашим інвентарем, а не зовнішнім стандартом.

Частка небажаного сканування = запити до нецільових шаблонів URL / усі підтверджені запити Googlebot

Формула 6.2. Частка небажаного сканування.

До нецільових можна віднести фасети, внутрішній пошук, застарілі параметри, нескінченні календарі та інші URL, які ви свідомо не хочете підтримувати як пошукові сторінки.

Затримка першого сканування = час першого підтвердженого запиту Googlebot - час публікації

Формула 6.3. Затримка першого сканування.

Рахуйте медіану та перцентилі окремо за типами сторінок. Середнє легко спотворюють поодинокі дуже довгі хвости.

```
Інтервал повторного сканування = поточний час сканування - попередній час сканування того самого URL
```

Формула 6.4. Інтервал повторного сканування.

Агрегуйте за типом сторінки, а не одним числом по всьому сайту.

Метрика	Рівень аналізу	Що я хочу побачити після виправлення
Частка корисного сканування	Хост → каталог → шаблон	Більша частка запитів іде на цільові URL, якщо ми справді прибрали марні простори.
Частка небажаного сканування	Параметр / фасет / внутрішній пошук	Падіння запитів до адрес, які не мають пошукової цінності.
Затримка першого сканування	Нові URL одного типу	Менший медіанний час від публікації до першого підтвердженого сканування.
Медіанний інтервал повторного сканування	URL, що часто змінюються	Частота краще відповідає реальній швидкості змін.
Частка 5xx/429	Хост і окремі шаблони	Падіння помилок і відсутність кореляції піків сканування з деградацією сервера.
Частка 304	Стабільні URL	Коректна перевірка актуальності для незмінених документів без зайвої передачі контенту.

Таблиця 6.6. Логові метрики повинні бути прив’язані до типу URL і конкретної технічної гіпотези.

6.8. 304 Not Modified: економія без вигаданого “підсилення ранжування”

Google прямо рекомендує підтримувати HTTP-кешування і 304 Not Modified. Якщо документ не змінився з попереднього сканування, 304 повідомляє пошуковому роботу, що можна використати кешовану версію без повторної передачі повного контенту. Це економить мережевий трафік і серверні ресурси.[1][5]

Водночас 304 не треба продавати як “фактор ранжування” або спосіб примусити Google сканувати більше. Коректний висновок

скромніший: для великих сайтів із великою часткою незмінених повторних запитів перевірка актуальності може зробити повторне завантаження дешевшим. Чи перетвориться ця економія на більший обсяг сканування саме ваших важливих URL, залежить від попиту й спроможності, а не від одного заголовка.

МІФ / ДАНІ / ВИСНОВОК Міф: “Якщо ввімкнути 304, Google гарантовано віддасть зекономлений бюджет сканування іншим сторінкам”. Дані: Google підтверджує економію мережевого трафіку і ресурсів. Але документація не обіцяє прямого перерозподілу кожного зекономленого запиту. Висновок: 304 - інфраструктурна оптимізація, не кредитна картка бюджету сканування.

6.9. Фасетна навігація: найтиповіший спосіб створити URL більше, ніж бізнесу потрібно

Google називає фасетну навігацію однією з найчастіших причин надмірного сканування, з якими власники сайтів звертаються до команди Search. Проблема не у фільтрах як таких, а в комбінаторному просторі URL: кожна властивість і кожен порядок параметрів можуть породжувати нову адресу, яку пошуковий робот не може оцінити до запиту.[4]

Якщо такі сторінки не потрібні в пошуку, Google рекомендує не давати їх сканувати, наприклад через robots.txt, або проектувати фільтри так, щоб вони не створювали URL, доступні для сканування. Якщо ж частина фасетів потрібна як посадкові сторінки, завдання складніше: треба чітко розділити комбінації, що мають індексуватися і технічний “довгий хвіст” параметрів, підтримувати стабільний порядок параметрів, віддавати 404 для порожніх комбінацій і не створювати нескінченні простори.[4]

Патерн	Чому він дорогий	Що перевірити
Фасети й сортування	Кількість комбінацій росте швидше за корисний інвентар	Які комбінації мають пошуковий попит; чи решта доступні для сканування; порядок параметрів; canonical; robots.

Патерн	Чому він дорогий	Що перевірити
Внутрішній пошук	Кожен запит може створити новий URL	Чи є посилання, доступні для сканування на результати пошуку; чи такі URL мають бути в індексі.
Календарі	Можуть генерувати нескінченний рух у майбутнє/минуле	Границі дат, nofollow/robots, 404 для неіснуючих інтервалів.
Ідентифікатори сесій / параметри відстеження	Один документ отримує багато технічних адрес	Джерело параметрів, внутрішні посилання, canonical, нормалізація на сервері.
Дублікати протоколу/хоста/шляхів	Інвентар розмножується без нової цінності	Редиректи, canonical, внутрішні URL, sitemap, налаштування хоста.
Soft 404	URL повертає 200, але фактично не має корисного контенту	Правильний 404/410 або реальний контент замість шаблонної пустої сторінки.

Таблиця 6.7. Надмірне сканування майже завжди починається з неконтрольованого інвентарю URL, а не з “надто активного Googlebot”.

6.10. HTTP-статуси: не намагайтеся керувати швидкістю неправильними кодами

Google прямо застерігає не використовувати 401, 403 або 404 для обмеження частоти. Усі 4xx, крім 429, не є сигналом “сервер перевантажений”. 429 якраз означає Too Many Requests і враховується як сигнал перевантаження сервера. 5xx також знижують інтенсивність сканування.[5]

Для аварійної ситуації, коли потрібно на кілька годин або один-два дні різко зменшити трафік сканування, Google рекомендує повертати 500, 503 або 429. Це грубий інструмент: він впливає на весь хост, уповільнює виявлення нових URL і повторне сканування, а якщо помилки тримаються кілька днів, URL можуть почати випадати з індексу. Це не постійний механізм “оптимізації бюджету”.[5]

Рис. 6.5. Найважливіші HTTP-відповіді з точки зору сканування.

Схема узагальнює офіційну документацію Google. Для повної обробки кожного статусу дивіться джерело 5.

6.11. robots.txt - критична інфраструктура, а не файл, який можна віддавати “як вийде”

robots.txt сам теж треба сканувати. Google зазвичай кешує його до 24 годин, але може тримати кеш довше, якщо оновлення недоступне. Особливо небезпечний сценарій - 5xx на robots.txt. В офіційній специфікації поведінка описана по часу: перші 12 годин Google припиняє сканування сайту, але продовжує намагатися отримати robots.txt; далі до 30 днів може використовувати останню доступну версію, якщо вона є; після 30 днів поведінка залежить від загальної доступності сайту.[5]

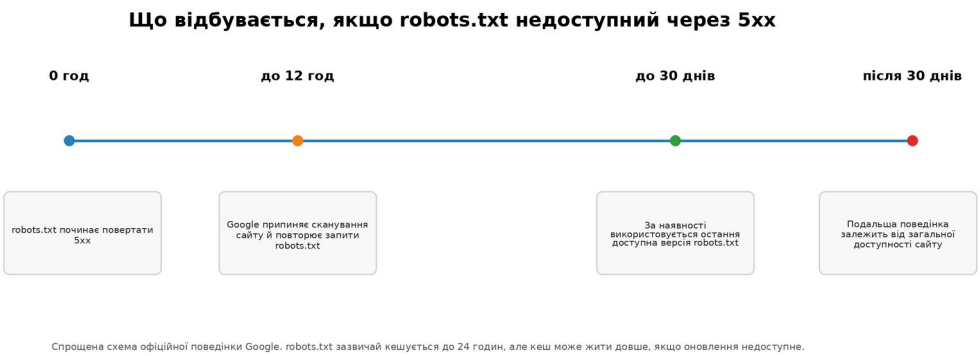


Рис. 6.6. Спрощена шкала поведінки Google при 5xx на robots.txt.

Джерело: Google Crawling Infrastructure. Схема не замінює повну специфікацію для DNS, тайм-аутів та інших мережевих помилок.

Практичний висновок: доступність robots.txt треба контролювати як критичну для доступності адресу. Якщо CDN, WAF або розгортання випадково починає віддавати 5xx на /robots.txt, наслідок не обмежується самим файлом. Це може зупинити або суттєво змінити сканування всього хоста.

6.12. Ліміт байтів: “сторінка відкривається в браузері” ще не означає, що Googlebot забрав її всю

У березні 2026 року Google окремо пояснив обмеження інфраструктури завантаження. Для Googlebot загального веб-пошуку завантажується до 2 МБ одного підтримуваного ресурсу в незжатому вигляді, включно з HTTP-заголовками; для PDF - до 64 МБ. Коли межа досягнута, пошуковий робот припиняє завантаження і передає далі лише вже отриману частину. Кожен CSS або JavaScript ресурс, на який посилається HTML, запитується окремо і має власну межу.[5]

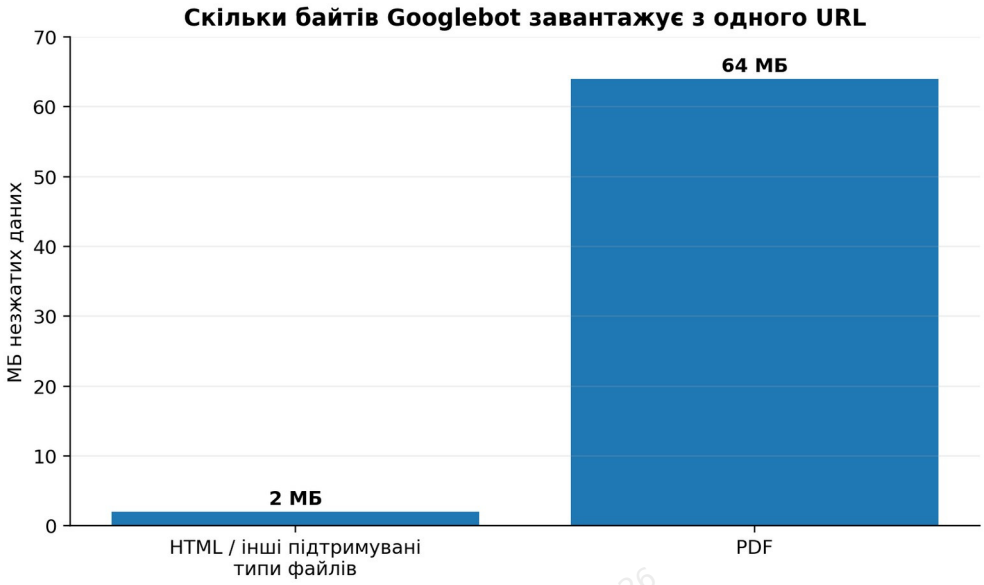


Рис. 6.7. Офіційні межі обсягу одного завантаження для Googlebot у 2026 році.

2 МБ для підтримуваних типів файлів загального веб-пошуку; 64 МБ для PDF. Інші пошукові роботи Google можуть мати інші межі.

Для більшості HTML-сторінок 2 МБ незжатого HTML - дуже великий обсяг. Але на масивних шаблонах масової генерації, довгих каталогах або сторінках із вбудованими даними це вже не теоретична межа. Я перевіряю не тільки переданий обсяг після gzip/brotli, а й фактичне незжате тіло відповіді, бо саме його обмеження описує Google.

6.13. Python: мінімальний аналіз підтверджених Googlebot-логів

Нижче я свідомо не намагаюся розбирати будь-який можливий формат nginx/Apache. У різних командах логи вже потрапляють у S3, BigQuery, ClickHouse або CSV з різними полями. Скрипт очікує нормалізований CSV після верифікації Googlebot з колонками timestamp, url, status, bytes, response_time. Далі задача SEO - перетворити запити на зрозумілі шаблони.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ

Я завантажу verified_googlebot.csv з колонками timestamp, url, status, response_time і bytes. Перевір типи даних і пропуски. Сегментуй URL за правилами, які я дам для product/category/search/other, порахуй кількість запитів і унікальних URL, медіанний response time, обсяг переданих байтів, частки 2xx/3xx/4xx/5xx та медіанний інтервал повторного сканування. Покажи аномалії, але не роби причинних висновків без додаткових даних.

АЛЬТЕРНАТИВА БЕЗ ШІ

Якщо CSV уже нормалізований, те саме можна зробити у Excel/Sheets: додати колонку типу сторінки, сімейство статусу, потім Pivot Table. Для інтервалу повторного сканування відсортуйте URL + timestamp і порахуйте різницю з попереднім запитом; Power Query зручніший на великих файлах.

QR: КОД

Скануйте камерою: QR містить компактну виконувану версію коду без коментарів. Коментований код наведено нижче.

ДЛЯ АВТОМАТИЗАЦІЇ: PYTHON ДЛЯ АНАЛІЗУ ЛОГІВ

```
import pandas as pd
from urllib.parse import urlsplit

# Очікувані поля: timestamp, url, status, response_time, bytes.
# CSV має містити вже верифіковані запити Googlebot.
df = pd.read_csv(
    "verified_googlebot.csv",
    parse_dates=["timestamp"]
)

# 1) Визначаємо тип сторінки за шляхом URL.
# Замініть правила нижче на структуру свого сайту.
def page_type(url):
    path = urlsplit(url).path
    if path.startswith("/product/"):
        return "product"
    if path.startswith("/category/"):
        return "category"
    if path.startswith("/search/"):
        return "internal_search"
    return "other"

# 2) Додаємо тип сторінки й сімейство HTTP-статусу.
df["page_type"] = df["url"].map(page_type)
df["status_family"] = (df["status"] // 100).astype(str) + "xx"
```

```
# 3) Зведення: скільки запитів, скільки URL,
# медіанний час відповіді та передані байти.
summary = (
   df.groupby("page_type")
     .agg(
        requests=("url", "size"),
        unique_urls=("url", "nunique"),
        median_response_time=("response_time", "median"),
        downloaded_bytes=("bytes", "sum"),
     )
     .sort_values("requests", ascending=False)
)

# 4) Частка 2xx/3xx/4xx/5xx усередині кожного типу сторінок.
status = pd.crosstab(
   df["page_type"],
   df["status_family"],
   normalize="index"
)

# 5) Медіанний інтервал між повторними запитами до того самого URL.
ordered = df.sort_values(["url", "timestamp"])
ordered["prev"] = ordered.groupby("url")["timestamp"].shift(1)
ordered["recrawl_hours"] = (
   ordered["timestamp"] - ordered["prev"]
).dt.total_seconds() / 3600
recrawl = ordered.groupby("page_type")["recrawl_hours"].median()

print(summary)
print(status)
print(recrawl)
```

Код 6.1. Базова сегментація підтверджених запитів Googlebot за типом сторінок, статусами й медіанним інтервалом повторного сканування.

Після цього я додаю бізнесові поля: цільовий canonical, можливість індексації, версію шаблону, час останньої зміни, дату публікації, стан інвентарю. Тоді лог перестає бути “технічним файлом” і перетворюється на систему, де можна відповісти, наприклад: скільки годин у медіані Googlebot потребує, щоб уперше просканувати новий товар; який відсоток його запитів до /search/; чи впала частка 5xx після зміни інфраструктури; чи скоротився інтервал повторного сканування для сторінок, де ціна змінюється щодня.

6.14. Діагностика: що робити, якщо Google сканує “не те”

Симптом	Найсильніша перша гіпотеза	Дані для перевірки	Дія
Багато запитів сканування до фасетів	Інвентар URL роздутий параметрами	Логи + сканування сайту + параметри + robots	Визначити набір фасетів, які мають індексуватися; решту прибрати з доступного для сканування графа або блокувати відповідно до стратегії.
Нові URL довго без першого сканування	Низький попит, слабке виявлення нових URL або перевантаження хоста	Час публікації, перший запит Googlebot, sitemap, внутрішні посилання, Crawl Stats	Розділити проблему виявлення нових URL і серверної спроможності; не лікувати все кнопкою запитом на індексацію.
Піки сканування збігаються з 5xx	Хост не тримає навантаження	Час відповіді, 5xx/429 за хвилинами, метрики CPU/DB/CDN	Підсилити серверну спроможність; не закривати Googlebot 403/404.
Google постійно сканує видалені URL	Старий інвентар продовжує отримувати сигнали або неправильні відповіді	Історія статусів, внутрішні/зовнішні посилання, sitemap, редиректи	404/410 для реально видалених; прибрати посилання і URL із sitemap; уникати soft 404.
Стабільні сторінки віддають повний 200 при кожному повторному скануванні	Немає ефективної перевірки актуальності	ETag/Last-Modified/If-Modified-Since, логи	Налаштувати коректне HTTP-кешування і 304 там, де документ справді не змінився.
Crawl Stats показує повільнішу відповідь	Інфраструктура або шаблон став дорожчим	Розбивка за типом файлу / хостом + серверні метрики	Знайти конкретний тип ресурсів або кінцеву точку; не робити висновок лише з середнього значення по всьому сайту.

Таблиця 6.8. Симптом сканування не дорівнює причині. Для кожної гіпотези потрібне окреме джерело даних.

6.15. Порядок аудиту бюджету сканування, який я використовую

Я починаю з інвентарю, а не з логу. Якщо команда сама не знає, які типи URL мають існувати в пошуку, неможливо визначити, що в трафіку сканування є “марним”. Далі дивлюся Crawl Stats, щоб зрозуміти масштаб, стан хоста, профіль часу відповіді і співвідношення виявлення/повторного сканування. Тільки після

цього йду в логи, бо без контексту мільйони рядків легко перетворити на красиві, але беззмістовні графіки.

Етап	Питання	Артефакт на виході
1. Інвентар URL	Які URL бізнес хоче підтримувати як пошукові сторінки?	Матриця: тип сторінки → цільовий для індексації / нецільовий / доступні для сканування / заблокований.
2. Архітектура	Звідки з'являються зайві адреси?	Карта параметрів, фасетів, сортувань, внутрішній пошук, календарні простори.
3. Crawl Stats	Чи є проблема масштабу або здоров'я хоста?	Тренд запитів, байтів, час відповіді, 5xx/429, Discovery/Refresh.
4. Логи	Які URL реально запитує Googlebot?	Розподіл запитів за шаблонами, статусами, інтервалами та затримкою першого сканування.
5. Сервер	Чи Googlebot упирається у серверну спроможність?	Кореляція трафіку сканування із затримкою, 5xx/429, CPU, DB, CDN.
6. Гіпотеза	Що саме ми змінюємо і який ефект очікуємо?	До/після за конкретною метрикою, а не "Googlebot став кращим".
7. Контроль	Чи не погіршили виявлення нових URL / деіндексацію / UX?	Паралельний контроль: звіт Page Indexing, GSC, sitemap та органічна видимість.

Таблиця 6.9. Аудит сканування як послідовність перевірок, а не список із 100 технічних прапорців.

6.16. Що точно не варто робити

Не потрібно блокувати все підряд у robots.txt лише тому, що Googlebot часто туди ходить. Якщо URL треба спочатку прибрати з індексу через noindex, пошуковий робот має мати можливість отримати цю директиву. Не потрібно робити випадкові щоденні зміни robots.txt "щоб перерозподіляти бюджет": Google прямо не рекомендує використовувати файл як тимчасовий механізм перерозподілу. Не потрібно відповідати 403 або 404, щоб "пригальмувати бота". І не потрібно вимірювати успіх просто падінням загальної кількості запитів сканування - менше запитів може означати як кращу ефективність, так і втрату попиту.[1][5]

ПОПЕРЕДЖЕННЯ Найнебезпечніша оптимізація бюджету сканування - та, яка успішно зменшила трафік сканування, але разом із ним зменшила виявлення нових URL або повторне сканування важливих URL. Мета не “менше Googlebot”. Мета - достатньо швидке сканування цільового інвентарю без марного навантаження на сервер.

6.17. Міфи, які я б викинув із технічного SEO

Міф	Що підтверджено	Практичний висновок
“У кожного сайту є фіксований денний бюджет сканування”	Google описує адаптивну модель спроможності та попиту.	Не шукайте одне число, якого Search Console все одно не показує.
“Будь-який сайт треба оптимізувати під бюджет сканування”	Google окремо орієнтує розширений посібник на великі/динамічні сайти або значну частку URL зі статусом «Виявлено, але наразі не проіндексовано».	Для малого сайту частіше важливі виявлення нових URL, можливість індексації, canonical і якість.
“Швидший сервер автоматично дає більше сканування”	Краща серверна спроможність може підняти межу спроможності, але потрібен ще попит на сканування.	Оптимізація сервера необхідна коли сервер є вузьким місцем, але не створює попит сама.
“404 економить бюджет сканування як обмеження частоти”	4xx, крім 429, не знижують частота сканування як сигнал перевантаження.	404/410 використовуйте за семантикою видаленого контенту, не для обмеження швидкості.
“Crawl Stats - це повний лог Googlebot”	Приклади URL у звіті репрезентативні, але не повні.	Для аналізу на рівні URL потрібні серверні логи.
“noindex економить сканування”	Google має отримати сторінку, щоб прочитати noindex.	Не плутайте керування індексацією з керуванням скануванням.
“Падіння кількості запитів сканування після змін завжди добре”	Менше запитів може означати і менший попит.	Оцінюйте затримку важливих URL, покриття і повторне сканування, а не загальну кількість запитів окремо.

Таблиця 6.10. Бюджет сканування без міфології: що реально впливає з поточної документації Google.

6.18. Що ми знаємо точно, а що лишається гіпотезою

Точно знаємо: Google визначає бюджет сканування через спроможність і попит; великий зайвий інвентар URL може зменшувати ефективність; час відповіді, 5xx і 429 впливають на спроможність сканування; популярність і застарівання впливають на попит; 304 може заощадити мережевий трафік; Search Console Crawl

Stats є агрегованим, а приклади URL не є повним логом; фасетну навігацію може створювати практично нескінченний простір URL; Googlebot має документовані межі байтів на одне завантаження.[1][2] [4][5]

Не знаємо публічно: точну формулу планувальника, ваги кожного сигналу, персональну “вартість” конкретного URL у черзі сканування, гарантований обсяг додаткового сканування після прискорення сервера або блокування фасетів. Якщо SEO-спеціаліст показує “Google виділив нам 187 432 умовних кредитів сканування на день” без внутрішніх даних Google, це не факт, а модель або вигадка.

6.19. Практичний висновок

Бюджет сканування має сенс лише тоді, коли ми перестаємо сприймати його як магічний ліміт. У великому сайті є інвентар URL, є серверна місткість, є попит пошукової системи, є історія змін і є конкретні HTTP-відповіді. SEO може дуже сильно впливати на перше, частково - на друге, опосередковано - на третє і напряму контролювати коректність четвертого та п’ятого.

Мій робочий критерій простий: важливі нові URL мають швидко отримувати перше сканування, важливі змінені URL - повторне сканування із частотою, що відповідає їхній бізнесовій динаміці, а сервер не повинен витрачати значну частину ресурсу на адреси, які продукт не збирається підтримувати в пошуку. Усе інше - засоби вимірювання цієї системи.

Джерела до Глави 6

1. Google Crawling Infrastructure: Optimize your crawl budget.
https://developers.google.com/crawling/docs/crawl-budget Оновлено 22.07.2026.
2. Google Search Console Help: Crawl Stats report.
https://support.google.com/webmasters/answer/9679690?hl=en
3. Google Search Central: Googlebot.
https://developers.google.com/search/docs/crawling-indexing/googlebot Оновлено 03.02.2026.
4. Google Crawling Infrastructure: Managing crawling of faceted navigation URLs.
https://developers.google.com/crawling/docs/faceted-navigation
5. Google Search Central / Crawling Infrastructure: technical crawling references.
https://developers.google.com/search/blog/2026/03/crawler-blog-post ;
https://developers.google.com/crawling/docs/crawlers-fetchers/reduce-crawl-rate ;
https://developers.google.com/crawling/docs/troubleshooting/http-status-codes ;
https://developers.google.com/crawling/docs/robots-txt/robots-txt-spec Межі завантаження Googlebot, аварійне зменшення частоти сканування, HTTP-коди та поведінка robots.txt.

ГЛАВА 7

Рендеринг: JavaScript, DOM і те, що реально бачить Google

Як відрізнити HTTP-відповідь від DOM після JavaScript, коли рендеринг реально ламає SEO, чому скриншот не дорівнює індексованому HTML і як перевіряти JavaScript-сайти без ритуалів та міфів.

Стан даних і джерел: 14 вересня 2026 року

Рендеринг часто плутають одразу з трьома різними речами: завантаженням HTML, виконанням JavaScript і тим, що врешті потрапило в індекс. Через це діагностика JavaScript-сайтів швидко перетворюється на магію: сторінка відкривається в Chrome, значить Google її бачить; або навпаки - контент з’являється через JavaScript, значить його обов’язково треба переносити на сервер. Обидва висновки занадто грубі.

У 2026 році Google прямо нагадує, що сам факт використання JavaScript більше не є коректним скороченням для “Google важче обробити сайт”. У березні Google прибрав із документації застарілий блок, пояснивши, що Search уже багато років рендерить JavaScript.[6] Але з цього не випливає, що архітектура не має значення. Вона визначає, на якому етапі з’являються контент, посилання, canonical, robots, структуровані дані та інші сигнали; скільки зовнішніх ресурсів треба отримати; чи потрібна взаємодія користувача; чи може API працювати без cookie, localStorage або дозволу.

ДАНІ Google документує три основні фази для JavaScript-сторінки: сканування, рендеринг та індексацію. URL із кодом 200 ставиться в чергу рендерингу, якщо сторінка не заборонена для індексації; очікування може тривати секунди або довше. Після рендерингу Google знову аналізує HTML і витягує посилання.[1] Це означає, що “Googlebot отримав 200” і “Google обробив контент після JavaScript” - не одна подія.

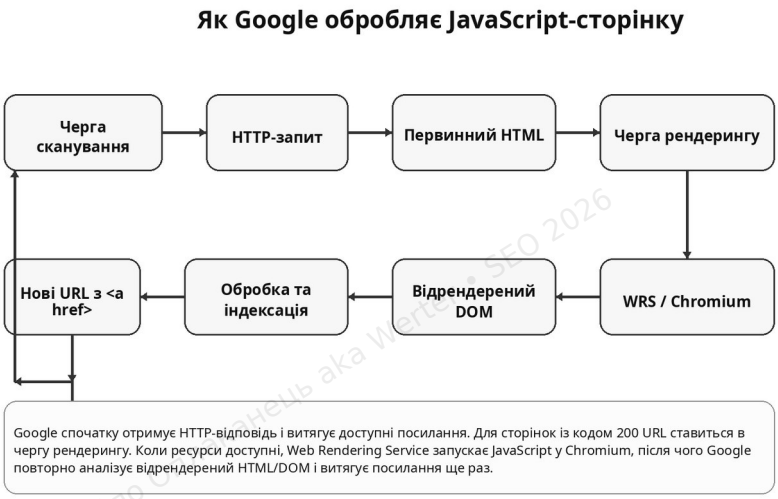


Рис. 7.1. Спрощений шлях JavaScript-сторінки від черги сканування до відрендереного DOM.

Схема автора за Google Search Central. Google не публікує точну внутрішню формулу пріоритизації черги рендерингу.

7.1. Рендеринг - це не “другий crawl” і не гарантія індексації

Перший корисний поділ - мережевий і браузерний. Під час HTTP-запиту сервер повертає статус, заголовки та тіло відповіді. На класичній серверній сторінці основний текст, посилання й SEO-метадані вже є в цьому HTML. На клієнтському застосунку відповідь

може містити лише оболонку, JavaScript-бандли та контейнер, у який контент буде вставлено пізніше.

Googlebot спочатку отримує саме HTTP-відповідь і вже з неї може витягнути доступні <a href> посилання. Далі сторінка з 200 може потрапити до Web Rendering Service, де Chromium без графічного інтерфейсу виконає JavaScript. Після цього Google отримує інше представлення документа - DOM після виконання сценаріїв - і ще раз шукає контент та посилання.[1]

Важлива деталь: не-200 відповідь може взагалі не пройти повний рендеринг.[1] Тому JavaScript не повинен бути способом “виправити” базову семантику HTTP. Якщо сторінки немає, краще повернути коректний 404/410; якщо вона переїхала - серверний редирект; якщо доступ закритий - відповідний 401/403. JavaScript-редиректи Google підтримує, але сам Google радить використовувати їх лише коли серверний або meta refresh варіант неможливий, бо JavaScript-редирект залежить від успішного рендерингу.[10]

Рис. 7.2. Для однієї URL треба розрізняти HTTP-відповідь, DOM після JavaScript і візуальний кадр.

Скриншот корисний для діагностики, але головний технічний доказ наявності тексту або посилання - відрендерений HTML/DOM, а не пікселі.

Таблиця 7.1. Чотири джерела доказів для одного URL не можна підміняти одне одним.

Шар	Що перевіряємо	Що він доводить	Чого не доводить
HTTP-відповідь	статус, HTTP-заголовки, вихідний HTML, robots, canonical, посилання	що сервер реально повернув до виконання JavaScript	що Google успішно виконав JavaScript і використав результат
DOM після JavaScript	текст, href, title, meta, JSON-LD, елементи після API-викликів	що браузер зміг сформувати після виконання сценаріїв	що саме цей стан збережений в індексі
жива перевірка в Search Console	відрендерений HTML, скриншот, ресурси, консоль JavaScript	що поточна версія сторінки доступна інструменту перевірки зараз	canonical, обраний Google, або те, що саме цю версію вже проіндексовано
Індексована версія	дані URL Inspection про останню відому Google версію	що Google зберіг після реального циклу сканування та індексації	що поточна версія сьогодні ідентична

7.2. Усі сторінки з 200 потрапляють у чергу рендерингу - але це не аргумент робити все через JavaScript

У поточній документації Google є формулювання, яке часто дивує навіть досвідчених SEO: усі сторінки з HTTP 200 ставляться в чергу рендерингу незалежно від того, чи є на них JavaScript.[1] Це не означає, що архітектура SSR і CSR для Search однакова. Різниця в тому, скільки корисної інформації Google має вже на першому етапі та скільки критичних елементів залежить від наступного виконання ресурсів.

Якщо основний текст, заголовок, навігація, canonical і JSON-LD є у первинному HTML, рендеринг радше доповнює документ. Якщо первинний HTML - порожня оболонка, успішний рендеринг стає умовою появи майже всього, що SEO хоче від сторінки. У другому випадку збільшується кількість точок відмови: JS-бандл, API, мережеві правила, блокування ресурсів, помилки виконання, стан клієнта, авторизація, експерименти та залежність від зовнішніх сервісів.

МІФ “Google тепер рендерить JavaScript, отже серверний HTML більше не має значення”. Google справді рендерить JavaScript, але водночас прямо рекомендує серверний або попередній рендеринг як хорошу практику; не всі боти виконують JavaScript, а серверний HTML зменшує кількість залежностей.[1][4]

7.3. SSR, SSG, hydration і CSR: для SEO важливе не модне скорочення, а місце появи критичного контенту

Я не оцінюю фреймворк за назвою. Next.js можна налаштувати так, що продуктова сторінка повертає повний HTML, а можна винести головний каталог у клієнтські запити. React сам по собі не робить сайт ні придатним для SEO, ні непридатним для SEO. Для аудиту важливіше відповісти: що є в HTTP HTML; що з’являється після JavaScript; що змінюється під час гідратації; що залежить від стану користувача.

web.dev у матеріалі, оновленому 5 січня 2026 року, визначає SSR як формування HTML на сервері, CSR - створення DOM у браузері JavaScript-ом, попередній рендеринг - підготовку статичного HTML під час збірки, а гідратацію - додавання клієнтського стану та інтерактивності до вже відрендереного HTML.[8] Для SEO ці визначення корисні не як рейтинг технологій, а як карта залежностей.

Рис. 7.3. Де формується основний контент у різних моделях рендерингу.

Динамічний рендеринг у схемі показаний окремо, бо Google вважає його тимчасовим обхідним рішенням, а не рекомендованою довгостроковою архітектурою.[4]

Таблиця 7.2. Архітектура сама по собі не є фактором ранжування; вона визначає точки технічної залежності.

Архітектура	Основний контент у HTTP HTML	Залежність від JavaScript для SEO	Практичний коментар
SSG / статична генерація	так	низька	сильна база для контентних і каталогових сторінок; JavaScript може залишатися для інтерактивності

Архітектура	Основний контент у HTTP HTML	Залежність від JavaScript для SEO	Практичний коментар
SSR	так	низька або середня	SEO-критичні дані доступні відразу, але серверний час відповіді та кешування все одно важливі
SSR + hydration	так	середня	HTML є, але після hydration треба перевірити, чи DOM не змінює canonical, контент або посилання некоректно
CSR	часто частково або ні	висока	основний контент залежить від JS, API та WRS; потребує окремої перевірки рендерингу
Динамічний рендеринг	версії для ботів і людей різні	штучно знижена для бота	Google називає це обхідним рішенням; краще виправляти архітектуру через SSR, статичний рендеринг або гідратацію

7.4. Які SEO-елементи я не хочу робити залежними від “пощастить після JS”

Технічно Google може побачити багато речей, доданих JavaScript. Документація прямо дозволяє змінювати title і метаопис, інжектити JSON-LD і навіть canonical.[1] Але “може обробити” і “це найнадійніша архітектура для критичного сигналу” - різні питання.

Для шаблонів, де SEO є важливою частиною бізнесу, я намагаюся мати у первинному HTML щонайменше основний контент сторінки, логічний H1, доступні для сканування посилання, базові robots/canonical сигнали та ті структуровані дані, які описують уже видимий контент. Це не вимога Google “все має бути SSR”; це інженерна стратегія зменшення кількості залежностей.

Таблиця 7.3. “Google підтримує через JavaScript” не означає “варто робити критичний елемент залежним від JavaScript”.

Елемент	Google може отримати після JS?	Що я перевіряю	Безпечніший дефолт
Основний текст	так	чи є в HTML після рендерингу без кліку/логіну/scroll-event	мати ключовий текст у HTML або гарантовано завантажувати без взаємодії
Внутрішні посилання	так, якщо це <a href> у DOM	наявність реального href після рендерингу	ключова навігація в HTML
title / meta description	так	чи не перезаписуються шаблонним або порожнім значенням	формувати коректно на сервері

Елемент	Google може отримати після JS?	Що я перевіряю	Безпечніший дефолт
canonical	так	конфлікт між HTTP HTML і DOM після рендерингу	одне стабільне значення; краще в HTML
robots meta	так, але є важлива пастка noindex	чи noindex не присутній у первинному HTML помилково	не ставити noindex у початковий HTML сторінки, яку треба індексувати
JSON-LD	так	чи описує видимий контент і присутній у HTML після рендерингу	серверний або стабільний клієнтський output

7.5. Пастка noindex: JavaScript не завжди встигне “передумати”

Один із найнебезпечніших шаблонів - віддавати noindex у первинному HTML, а потім знімати його JavaScript-ом, коли API підтвердить, що сторінка валідна. Google прямо попереджає: коли він зустрічає noindex, рендеринг і виконання JavaScript можуть бути пропущені. Тому логіка “спочатку noindex, потім JavaScript зробить index” ненадійна.[1]

Зворотний сценарій допустиміший: сторінка стартує без noindex, а JavaScript додає noindex, якщо API повернув стан, який не повинен індексуватися. Але для масових шаблонів я все одно віддаю перевагу правильному HTTP-статусу або серверній логіці. Особливо для товарів, вакансій і сторінок сутностей, де “існує / не існує” бізнес уже знає на сервері.

ПОПЕРЕДЖЕННЯ Якщо сторінка має індексуватися, не використовуйте noindex у первинному HTML як тимчасову заглушку з надією прибрати його JavaScript-ом. Це офіційно описана зона ризику. [1]

7.6. Контент після натискання, прокручування або дозволу користувача - це вже інший клас ризику

Google Search не взаємодіє зі сторінкою як людина. В офіційній документації про відкладене завантаження прямо сказано: реалізація не повинна покладатися на дії користувача на кшталт прокручування або кліку.[3] У посібнику Google з усунення проблем окремо зауважує, що функції, які вимагають дозволів, можуть бути безглуздими для Googlebot: він не надасть сайту камеру лише для того, щоб отримати основний контент.[2]

Тому важливий розподіл не “JavaScript / без JavaScript”, а “автоматично формується під час рендерингу / з’являється лише після окремої взаємодії”. Якщо текст відкривається після обробника натискання, список підвантажується лише на подію прокручування, API вимагає cookie з попередньої сесії або контент сидить за API дозволів - звичайна перевірка користувачем у браузері може вводити SEO в оману.

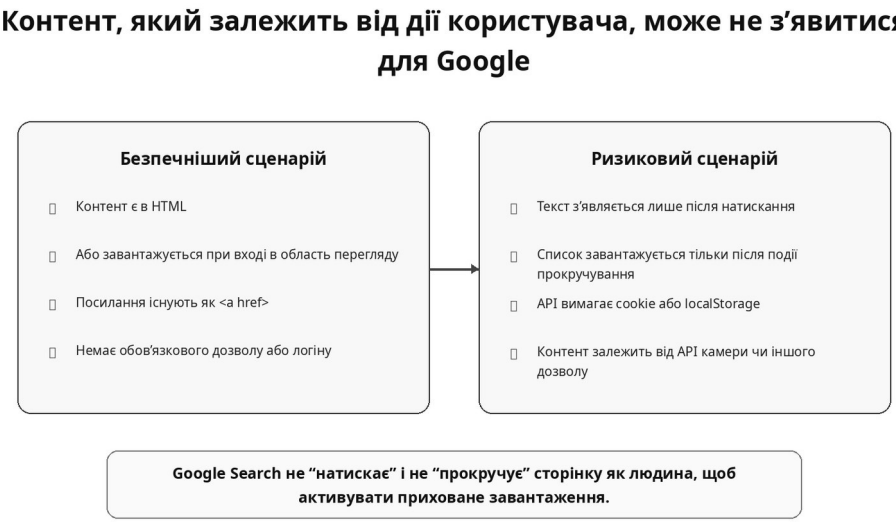


Рис. 7.4. Основний SEO-контент не повинен залежати від обов’язкової дії користувача.

Google Search не “проклікує” інтерфейс, щоб відкрити прихований основний контент. Для lazy-loading потрібна автоматична поява контенту, коли він стає видимим в області перегляду.[3]

Таблиця 7.4. Взаємодія користувача може бути UX-функцією, але не повинна бути єдиним шляхом до критичного SEO-контенту.

Залежність	Що бачить людина	Ризик для Google	Що робити
натискання “Показати ще”	контент після кліку	основний контент може не з’явитися під час рендерингу	не робити критичний текст залежним лише від обробника натискання

Залежність	Що бачить людина	Ризик для Google	Що робити
подія прокручування	елементи після ручної прокрутки	Google не взаємодіє зі сторінкою як людина	IntersectionObserver або вбудоване відкладене завантаження браузера; тестувати HTML після рендерингу
cookie / localStorage	контент після попередньої сесії	WRS не зберігає стан між завантаженнями сторінок[2]	мати працездатний початковий стан без збереженого стану браузера
login	персональний контент	Googlebot не проходить авторизацію як звичайний користувач	публічний SEO-контент має бути доступний без логіну
Camera/API дозволів	контент після дозволу	Googlebot не надає обов’язковий користувацький дозвіл[2]	не робити дозвіл умовою для основного індексованого контенту

7.7. Відкладене завантаження й нескінченна прокрутка: проблема не в техніці, а в недоступному HTML

Відкладене завантаження саме по собі не є SEO-помилкою. Google рекомендує його для невидимих або некритичних ресурсів, але з умовою: релевантний контент має завантажуватися, коли входить в область перегляду, без ручного натискання або прокручування як обов’язкового тригера.[3] Для зображень і iframe найпростіший

варіант - вбудоване відкладене завантаження браузера; для інших сценаріїв Google наводить IntersectionObserver.

Нескінченна прокрутка має окрему умову. Кожен логічний сегмент повинен мати постійну унікальну URL, однаковий контент при повторному відкритті й послідовні доступні для сканування посилання між сторінками. Коли під час прокручування новий сегмент стає основним, URL можна оновлювати History API.[3] Це вже не косметична техніка: без самостійних URL і пагінації пошуковий робот отримує лише один нескінченний інтерфейс без стабільної адресації.

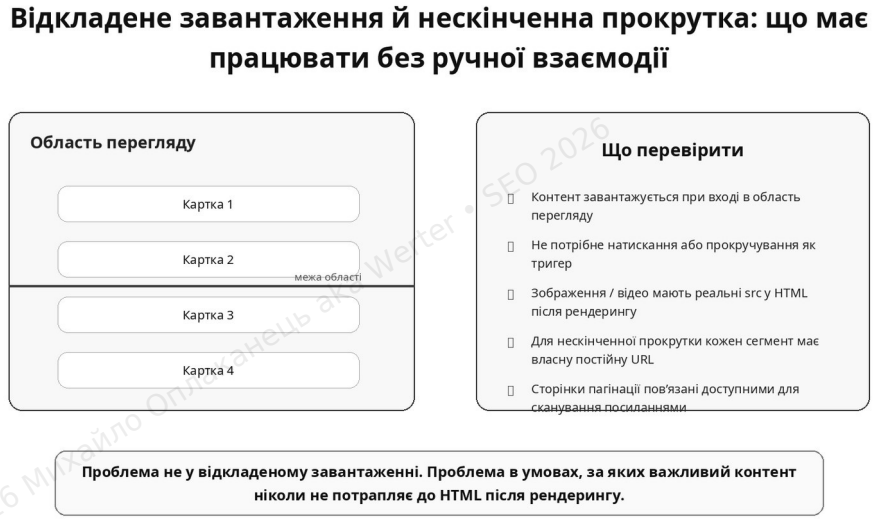


Рис. 7.5. Що перевіряти у відкладеному завантаженні та нескінченній прокрутці.

Критерій не “чи бачить це людина після прокрутки”, а “чи потрапляє контент і його URL у відрендерений HTML без обов’язкової ручної взаємодії”.

7.8. Ресурси JavaScript і CSS: рендеринг може ламатися не на HTML

Коли сторінка повертає правильний 200 і достатній HTML, але жива перевірка показує неповний DOM, я дивлюся на ресурси. WRS може не отримати JavaScript-бандл через robots.txt, CDN/WAF,

помилки хоста, відмінну поведінку для ідентифікатора бота або зовнішній API. Google також пише, що WRS аналізує ресурси й може не завантажувати ті, які не потрібні для основного контенту, наприклад частину аналітики або запитів звітування про помилки.[2]

Ще одна неочевидна проблема - кеш. Googlebot кешує ресурси агресивно, а WRS може ігнорувати заголовки кешування. Якщо файл main.js змінюється, але URL лишається тим самим, Google може певний час використати старішу версію. Google рекомендує версіювання імені файлу за вмістом: ім’я файлу змінюється разом із його вмістом, наприклад main.2bb85551.js.[1]

Таблиця 7.5. Проблема рендерингу часто знаходиться у ресурсному графі, а не в HTML документа.

Симптом	**Перша перевірка**	**Типова причина**	**Доказ**
HTML є, DOM порожній	мережеві запити / ресурси сторінки	JavaScript-бандл не завантажився або повернув 4xx/5xx	конкретний ресурс із помилкою
частина даних порожня	консоль JavaScript + API-запити	CORS, авторизація, cookie, обмеження частоти запитів, заблокований endpoint	помилка в консолі або мережевому запиті
Google бачить стару версію UI	URL ресурсів і версіювання за вмістом	той самий URL JS/CSS при зміні контенту	версії JavaScript-бандла у мережевих логах
локально все працює, жива перевірка ні	WAF/CDN/robots для ресурсів	відмінна політика для IP/UA краулера	ресурси в живій перевірці + логи сервера/CDN

Симптом	Перша перевірка	Типова причина	Доказ
аналітичний запит відсутній	чи він потрібен для контенту	WRS може не брати неістотні запити	не вважати це дефектом рендерингу без втрати контенту

7.9. Canonical, title і структуровані дані після JavaScript: технічно можна, але конфлікти треба ловити

Google підтримує JavaScript-зміни title та meta description, може прочитати JSON-LD, доданий через JavaScript і canonical.[1] Але canonical має окреме застереження: якщо canonical є в початковому HTML, JavaScript не повинен змінювати його на іншу URL. Якщо сервер не може поставити canonical, його можна додати JavaScript-ом, але не створювати конфлікт двох значень.[1]

У практичному аудиті я порівнюю не лише “canonical є / немає”, а три стани: HTTP HTML, DOM після JavaScript і те, що Search Console показує як canonical, заявлений сайтом / canonical, обраний Google. Якщо перші два вже конфліктують, немає сенсу шукати складну алгоритмічну причину, чому Google обрав не ту URL.

Таблиця 7.6. Порівняння до/після JavaScript швидше знаходить конфлікти, ніж перевірка одного стану.

Сигнал	HTTP HTML	DOM після JS	SEO-висновок
canonical A	A	A	стабільна реалізація
canonical A	B	B	конфлікт реалізації; JS перезаписує сигнал

Сигнал	HTTP HTML	DOM після JS	SEO-висновок
немає	A	A	Google може обробити, але сигнал залежить від рендерингу
title шаблонний	title унікальний	унікальний	може працювати, але помилка рендерингу повертає слабкий запасний варіант
JSON-LD немає	JSON-LD є	валідний	підтримується; тестувати результат після рендерингу

7.10. Shadow DOM і веб-компоненти: Google бачить не те саме, що “View Source”

Для веб-компонентів Google документує підтримку: під час рендерингу він “сплющує” light DOM і shadow DOM у HTML після рендерингу. Практичний критерій простий: якщо потрібний контент не видимий у HTML після рендерингу, Google не зможе його індексувати.[1] Тому перевіряти лише View Source для сайту з веб-компонентами недостатньо.

Це хороший приклад ширшої проблеми: View Source показує отриманий від сервера HTML, а Elements у DevTools - поточний DOM браузера. Для класичної сторінки вони схожі; для складного клієнтського застосунку можуть відрізнятися радикально. В аудиті треба знати, який саме шар ви дивитесь.

7.11. URL Inspection: найкорисніша точка перевірки для конкретної URL, але не оракул

URL Inspection поєднує два джерела даних: інформацію з індексу Google і дані живої перевірки. Індексована версія показує, що Google знає про сторінку після свого циклу сканування та індексації. Жива перевірка генерується окремо в момент перевірки і корисна для діагностики поточного стану, але не передбачає, наприклад, який canonical Google реально обере при індексації.[5]

У живій перевірці можна подивитися скриншот, HTML, HTTP-заголовки та консоль JavaScript і завантажені ресурси.[5] Для аудиту рендерингу це сильний набір. Але я не роблю висновок “все добре” лише зі скриншота: скриншот - візуальний кадр. Текст може бути нижче, елемент може бути присутній у HTML і не потрапити у видимий кадр, або навпаки інтерфейс може виглядати схожим, але критичне посилання не мати доступного для сканування href.

Рис. 7.6. Порядок перевірки рендерингу конкретної URL.

Жива перевірка та індексовані дані відповідають на різні питання. Результат живої перевірки Google не використовує як індексовану версію сторінки.[5]

Таблиця 7.7. Жоден інструмент окремо не дає повної картини рендерингу.

Інструмент	Що дивлюся	Сильна сторона	Обмеження
curl / raw response	статус, HTTP-заголовки, HTML	точно показує серверну відповідь	не виконує JavaScript
Chrome DevTools	DOM, мережеві запити, консоль	швидко локалізує помилки JavaScript, API або ресурсів	це звичайне браузерне середовище, а не Google
URL Inspection жива перевірка	HTML після рендерингу, скриншот, ресурси, консоль	перевірка поточної URL інструментом Google	не є даними індексу і не прогнозує canonical, який обере Google
URL Inspection indexed data	останнє сканування, стан індексації, canonical	реальний стан у системі Google	може бути старішим за поточний реліз
логи сервера/CDN	фактичні запити ресурсів	подієвий доказ доступу	не показують, що DOM сформувався правильно

7.12. Як я перевіряю один URL за 10-15 хвилин

Я починаю не з фрази “сайт на React”. Спершу фіксую конкретний симптом: відсутній текст, посилання, canonical, структуровані дані, зображення, пагінація або весь шаблон. Далі йду від найнижчого шару до найвищого.

Спочатку дивлюся HTTP-статус та первинний HTML. Якщо елемент уже відсутній там - це ще не дефект, але я вже знаю, що він залежить від наступного етапу. Потім відкриваю DevTools: чи з’явився елемент у DOM, які запити його створили, чи є помилки в консолі. Після цього запускаю живу перевірку в Search Console і перевіряю HTML після рендерингу, ресурси й скриншот. І лише потім порівнюю індексовану

версію, щоб не плутати поточну версію сайту зі станом, який Google зберіг раніше.

Рис. 7.7. Діагностичне дерево для симптома “у браузері є, у Google немає”.

Починати треба з шару, на якому зникає елемент: HTTP HTML, DOM після JS, жива перевірка або вже індексована версія.

7.13. Масштабний аудит: порівнюйте шаблони, а не 100 000 URL вручну

Для великого сайту перевіряти рендеринг по одній URL безглуздо. Я спочатку будую типологію: товар, категорія, стаття, посадкова сторінка, пагінація, фільтр, профіль тощо. На кожний тип беру кілька типових URL з різним станом: нова, стара, з малою кількістю контенту, з великим, з нульовим результатом, з пагінацією. Після цього перевіряю, чи однакова розбіжність рендерингу повторюється на шаблоні.

На рівні масового сканування корисно порівняти режими Text Only і JavaScript Rendering. Screaming Frog дозволяє запускати обидва режими й експортувати title, meta robots, canonical, H1, кількість посилань, структуровані дані та користувацьке вилучення. Тоді

питання “у нас JavaScript проблема?” перетворюється на таблицю “які саме поля змінюються після рендерингу і на яких типах URL”.

Залежність від рендерингу = критичні елементи, відсутні в HTTP HTML, але наявні після JS / усі перевірені критичні елементи

Формула 7.1. Коефіцієнт залежності SEO-елементів від рендерингу.

Це внутрішній діагностичний показник автора, а не офіційна метрика Google. Її сенс - порівнювати шаблони та релізи, а не “оптимізувати число заради числа”.

Частка розбіжностей = URL з ≥1 критичною розбіжністю / усі протестовані URL у шаблоні

Формула 7.2. Частка URL із критичною розбіжністю рендерингу.

Критичною розбіжністю може бути відсутній основний текст, href, canonical, robots, title або структуровані дані - набір треба зафіксувати до тесту.

Таблиця 7.8. Масштабний аудит рендерингу має бути шаблонним і порівняльним, а не ручним списком скриншотів.

Зріз	Приклад перевірки	Що сегментувати
тип сторінки	товар / категорія / стаття	окремі шаблони
HTTP vs JS	title, canonical, robots, H1, текст, internal links	поле та тип URL
помилки ресурсів	4xx/5xx JS/CSS/API	хост, JavaScript-бандл, реліз
час рендерингу / симптоми тайм-ауту	повний vs неповний DOM	шаблон, пристрій, реліз
індексована версія vs поточна	старий DOM / новий DOM	дата сканування, дата релізу

7.14. Автоматизація: порівняння первинного HTML і DOM після JavaScript

Нижче - мінімальний локальний тест. Він не емулює Googlebot і не замінює URL Inspection. Його завдання простіше: показати, які базові SEO-елементи змінюються між HTTP-відповіддю та браузерним DOM після JavaScript. Для масового аудиту в робочому середовищі я б додавав список URL, контроль User-Agent, тайм-аути, логування помилок ресурсів, власні селектори та збереження різниці у CSV.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ Я надам два HTML-файли однієї URL: raw.html з HTTP-відповіді та rendered.html після JavaScript. Порівняй тільки факти у файлах. Витягни title, meta robots, canonical, H1, усі внутрішні <a href>, JSON-LD, довжину видимого тексту та вказані мною CSS-селектори. Побудуй таблицю: елемент, raw, rendered, тип зміни, SEO-ризик. Окремо виділи елементи, що з'являються лише після JavaScript, зникають після JavaScript або конфліктують. Не роби висновків про індексацію, яких ці два HTML не доводять.	**QR: КОД** Скануйте камерою. QR містить компактну виконувану версію коду.

Альтернатива без Python: зробіть два сканування одного набору URL у Screaming Frog - Text Only і JavaScript rendering - та зіставте експорти за URL. Для одиничної URL достатньо HTTP-відповіді + Chrome DevTools + живої перевірки в Search Console. Python потрібен, коли порівняння треба повторювати регулярно або на великій вибірці.

```
import sys
import requests
from bs4 import BeautifulSoup
from playwright.sync_api import sync_playwright

# Запуск: python render_compare.py https://example.com/page
url = sys.argv[1]

# 1) Отримуємо початковий HTML без виконання JavaScript.
```

```
raw_html = requests.get(
    url,
    headers={"User-Agent": "Mozilla/5.0"},
    timeout=30,
).text

# 2) Відкриваємо ту саму URL у Chromium і чекаємо завершення мережевих запитів.
# Це локальний браузерний тест, а не емуляція Googlebot.
with sync_playwright() as p:
    browser = p.chromium.launch()
    page = browser.new_page()
    page.goto(url, wait_until="networkidle", timeout=60_000)
    rendered_html = page.content()
    browser.close()

# 3) Витягуємо мінімальний набір SEO-елементів з кожної версії HTML.
def extract(html):
    soup = BeautifulSoup(html, "html.parser")
    canonical = soup.find("link", rel="canonical")
    robots = soup.find("meta", attrs={"name": "robots"})
    return {
        "title": soup.title.string.strip() if soup.title and soup.title.string
else "",
        "h1": [x.get_text(" ", strip=True) for x in soup.find_all("h1")],
        "canonical": canonical.get("href") if canonical else "",
        "robots": robots.get("content") if robots else "",
        "links": len(soup.find_all("a", href=True)),
        "text_chars": len(soup.get_text(" ", strip=True)),
    }

print("RAW:", extract(raw_html))
print("RENDERED:", extract(rendered_html))
```

Код 7.1. Мінімальне порівняння HTTP HTML і браузерного DOM після JavaScript. Потрібні requests, beautifulsoup4 та playwright.

7.15. Динамічний рендеринг: не плутайте тимчасовий обхід із сучасною рекомендацією

У старих JavaScript SEO-гайдах динамічний рендеринг часто виглядає як нормальний шлях: ботам віддаємо попередньо відрендерений HTML, користувачам - клієнтський застосунок. Поточна позиція Google інша: динамічний рендеринг - тимчасовий обхід, а не довгострокове рішення. Google рекомендує серверний рендеринг, статичний рендеринг або гідратацію.[4]

Причина не лише у політиці Google. Дві версії одного документа треба синхронізувати; з’являються різні кеші, різні помилки, різний результат canonical і контенту. Якщо команда вимушено використовує динамічний рендеринг для застарілої системи, я ставлю йому строк життя та контроль відповідності між версією для

ботів і версією для користувачів, а не перетворюю його на постійну архітектуру.

7.16. Міфи JavaScript SEO, які варто прибрати з аудиту

Таблиця 7.9. JavaScript SEO без ритуалів: кожен висновок має прив'язуватися до конкретного шару обробки.

Міф	Що підтверджено	Практичний висновок
"JavaScript сам по собі робить SEO складнішим для Google"	У березні 2026 Google прибрав застаріле формулювання: Search рендерить JS багато років.[6]	Шукайте конкретну залежність або сценарій відмови, а не звинувачуйте технологію цілком.
"Якщо сторінка відкривається у Chrome, Google бачить те саме"	WRS має інший контекст, не зберігає стан між завантаженнями сторінок і не надає обов'язкові API-дозволи.[2]	Перевіряйте живу перевірку та HTML після рендерингу.
"Скриншот у Search Console доводить, що все індексується"	Скриншот показує візуальний результат живої перевірки.[5]	Для тексту, href, canonical і meta перевіряйте HTML; для стану індексу - індексовані дані.
"View Source = те, що бачить Google"	Google може виконати JS і проаналізувати HTML після рендерингу.[1]	Порівнюйте первинний HTML і DOM після JS.
"Динамічний рендеринг - рекомендований спосіб для SEO"	Google називає його обхідним рішенням і рекомендує SSR, статичний рендеринг або гідратацію.[4]	Не закладайте дві версії сайту як стандартну нову архітектуру.

Міф	Що підтверджено	Практичний висновок
“noindex можна безпечно прибрати JavaScript-ом”	Google може пропустити rendering, коли noindex є в первинному HTML.[1]	Не віддавайте noindex на сторінці, яку треба індексувати, з надією зняти його JS.

7.17. Що ми знаємо точно, а що лишається невідомим

Точно знаємо: Google сканує, ставить 200-сторінки в чергу рендерингу, виконує JavaScript у сучасному Chromium, повторно аналізує HTML після рендерингу та витягує посилання; підтримує JavaScript-зміни title, метаопис, canonical і структуровані дані; Search не взаємодіє зі сторінкою як користувач для відкладеного завантаження; WRS не зберігає localStorage/sessionStorage між завантаженнями сторінок; динамічний рендеринг не є рекомендованим довгостроковим рішенням.[1][2][3][4]

Не знаємо публічно: точний пріоритет конкретної URL у черзі рендерингу, гарантований максимальний час очікування, точну внутрішню модель планування ресурсів WRS, які саме ресурси система вирішить пропустити на конкретному рендері, і повну формулу, за якою документ після рендерингу впливає на подальшу індексацію. Тому будь-які “ліміти рендерингу Google” у секундах або магічна допустима кількість JS, якщо вони не мають актуального офіційного джерела, я не використовую як факт.

7.18. Практичний висновок

JavaScript SEO у 2026 році - не окрема магічна дисципліна. Це контроль переходу документа між станами. Сервер повернув одну версію. Браузер після JavaScript сформував другу. Жива перевірка Google може показати третю поточну версію, а в індексі зберігатися попереднє сканування. Завдання SEO - знайти, на якому саме переході губиться критичний елемент.

Мій робочий стандарт простий: усе, що критично для змісту сторінки, навігації й базових індексаційних сигналів, має бути або в первинному HTML, або гарантовано з'являтися в HTML після рендерингу без кліку, прокрутки, логіну, дозволів та збереженого стану. Аудит починається з HTTP і DOM, а не з назви фреймворку. Якщо елемент є у первинному HTML - рендеринг не є його першою проблемою. Якщо його немає у первинному HTML, але є після JavaScript - далі перевіряємо живу перевірку Google. Якщо є й там - переходимо до canonical, індексації та якості, а не продовжуємо звинувачувати JavaScript.

Джерела до Глави 7

1. Google Search Central. Understand JavaScript SEO Basics. https://developers.google.com/search/docs/crawling-indexing/javascript/javascript-seo-basics Основи crawling, черги рендерингу, WRS, canonical, noindex, кешування та веб-компонентів.
2. Google Search Central. Fix Search-Related JavaScript Problems. https://developers.google.com/search/docs/crawling-indexing/javascript/fix-search-javascript Обмеження WRS, дозволи, localStorage/sessionStorage, ресурси та діагностика.
3. Google Search Central. Fix Lazy-Loaded Website Content. https://developers.google.com/search/docs/crawling-indexing/javascript/lazy-loading Відкладене завантаження без ручної взаємодії та індексована нескінченна прокрутка.
4. Google Search Central. Dynamic Rendering as a workaround. https://developers.google.com/search/docs/crawling-indexing/javascript/dynamic-rendering Динамічний рендеринг як тимчасовий обхід; рекомендовані серверний/статичний рендеринг або гідратація.
5. Google Search Console Help. URL Inspection tool. https://support.google.com/webmasters/answer/9012289?hl=en Жива перевірка, скриншот, HTML після рендерингу, ресурси та консоль JavaScript.
6. Google Search Central. Latest Google Search Documentation Updates. https://developers.google.com/search/updates 04.03.2026: Google прибрав застарілий блок, наголосивши, що Search рендерить JavaScript багато років.
7. Google Search Central. In-Depth Guide to How Google Search Works. https://developers.google.com/search/docs/fundamentals/how-search-works Рендеринг JavaScript під час сканування та наступна індексація.
8. web.dev. Rendering on the Web. https://web.dev/articles/rendering-on-the-web Оновлено 05.01.2026. SSR, CSR, попередній рендеринг, гідратація та продуктивність.
9. web.dev. Client-side rendering of HTML and interactivity. https://web.dev/articles/client-side-rendering-of-html-and-interactivity Вплив клієнтського рендерингу на DOM і головний потік браузера.
10. Google Search Central. Redirects and Google Search. https://developers.google.com/search/docs/crawling-indexing/301-redirects JavaScript-редиректи через WRS; пріоритет серверного редиректу.
11. Google Search Central. Get started with Search: a developer's guide. https://developers.google.com/search/docs/fundamentals/get-started-developers JavaScript, semantic HTML, текст у DOM та URL для SPA.

ЧАСТИНА II / ЯК ПОШУКОВА СИСТЕМА БАЧИТЬ ВЕБ

ГЛАВА 8

Канонікалізація: дублікати, кластери й вибір головної версії

Як читати canonical не як тег, а як систему узгодження сигналів: чому дублікати нормальні, коли Google ігнорує вашу підказку, як діагностувати конфлікти і як не втрачати дані між URL-варіантами.

Стан даних і джерел: 14 вересня 2026 року

Canonical часто перевіряють як булеве поле: є тег - добре, немає - погано. Це слабка модель. Канонікалізація відбувається не в HTML-тегу, а в системі, яка групує однакові або дуже схожі документи і вибирає представника кластера. rel="canonical" лише один із сигналів. Google прямо описує canonical як представницьку URL, яку система обирає серед дублікатів; redirect і rel="canonical" є сильними сигналами, sitemap - слабшим, а остаточний вибір залишається за Google.[1][2]

Практично це означає, що проблема canonical рідко вирішується додаванням ще одного тегу. Якщо внутрішні посилання ведуть на одну версію, sitemap містить другу, redirect веде на третю, hreflang вказує четверту, а canonical - п’яту, ви самі створили систему голосування без більшості. Технічний аудит повинен перевіряти узгодження сигналів на рівні кластера URL, а не на рівні одного документа.

Рис. 8.1. Канонікалізація працює з кластером URL, а не з одним тегом на сторінці.

Схема автора за документацією Google Search Central про canonicalization і consolidation duplicate URLs.

8.1. Дублікат не означає порушення

Дублікатність усередині сайту сама по собі не є спамом. Параметри сортування, UTM, HTTP/HTTPS, www/non-www, альтернативні регіональні URL, printer-friendly сторінки або CMS-варіанти можуть породжувати однаковий основний контент. Завдання системи - не “покарати дубль”, а не зберігати і не показувати десяток еквівалентних представників без потреби.[1]

Для SEO наслідок інший: якщо багато URL представляють той самий документ, сигнали та звітність можуть фрагментуватися. Search Console звітує значну частину даних на canonical URL, тому аналітик, який дивиться лише на сирі URL з GA4 або crawl, може бачити іншу картину, ніж Google. Саме тому canonical-кластер варто розглядати як окрему одиницю даних.

8.2. Сила сигналів: redirect, canonical, sitemap і внутрішні посилання

Таблиця 8.1. Сигнали canonicalization і їх практичне значення.

Сигнал	Що він говорить системі	Типова помилка
301/308 redirect	Стара URL постійно переміщена на нову. Google називає redirect сильним canonical-сигналом.	Redirect веде на A, а canonical нової сторінки повертає на B.
rel="canonical"	Яка URL є бажаним представником дубліката. Сильний сигнал, але не директива.	Canonical на нееквівалентну сторінку або ланцюг canonical.
Sitemap	Які URL власник сайту вважає основними. Слабший сигнал.	У sitemap системно лежать параметричні або redirect-URL.
Внутрішні посилання	Яку версію сайт реально використовує в графі.	Меню, breadcrumbs і контент лінкують на неканонічні варіанти.
hreflang	Зв’язок між локалізованими варіантами. Має узгоджуватися з canonical.	hreflang вказує URL, яка canonical-иться в інший кластер.

Google окремо радить посилатися всередині сайту саме на canonical URL. Це не означає, що одна випадкова внутрішня ссылка “перепише” canonical; це означає, що граф сайту повинен бути узгодженим. Якщо CMS десятиліттями генерує параметричні URL, canonical не перетворює цю архітектуру на чисту - він лише допомагає системі зрозуміти ваш намір.

8.3. Self-canonical корисний як декларація, але не магічний захист

Self-referencing canonical я ставлю на основні HTML-сторінки за замовчуванням, якщо немає спеціальної причини робити інакше. Це робить намір явним і зменшує залежність від того, як CMS згенерує параметри або альтернативні URL. Але self-canonical не “закріплює” сторінку назавжди. Якщо система бачить сильніший набір суперечливих сигналів або вважає іншу URL кращим представником дубліката, вона може вибрати іншу canonical.[1][3]

НОТАТКА З ПРАКТИКИ
У масштабному аудиті я не рахую “% сторінок із canonical”. Я рахую щонайменше: % URL, де declared canonical = final 200 URL; % canonical, які ведуть через redirect; % canonical, що виходять за host; % indexable URL з canonical на noindex/404; % внутрішніх посилань на неканонічні URL.

8.4. Canonical має вести на еквівалентний документ

Найгірша звичка - використовувати canonical як сміттєпровід. Наприклад, усі фільтровані категорії canonical-ити на базову категорію, навіть коли фільтр має інший набір товарів і самостійний пошуковий попит; або всі варіанти товару canonical-ити на один SKU, хоча контент і пропозиція суттєво відрізняються. Якщо сторінки не є дублями або дуже близькими варіантами, canonical не замінює нормальне рішення про індексацію, архітектуру чи редирект.

Окремо небезпечні cross-domain canonical. Вони технічно підтримуються, але це сильне твердження: “основна версія цього матеріалу знаходиться на іншому домені”. Для синдикації це може бути доречно; для випадково скопійованого шаблону - катастрофа. Я завжди включаю зовнішні canonical у окремий звіт.

8.5. Canonical і hreflang: спочатку кластер, потім локаль

Для регіональних сторінок однією мовою Google рекомендує поєднувати canonical і hreflang. Практична модель: кожна

локалізована URL має залишатися canonical у власному логічному кластері, а hreflang пов’язує альтернативи. Якщо es-ES canonical-иться на глобальну англійську URL, а hreflang одночасно оголошує es-ES як окремий варіант, система отримує конфлікт намірів.[4][5]

8.6. Як читати “Google-selected canonical”

Коли URL Inspection показує, що user-declared canonical і Google-selected canonical не збігаються, це не діагноз. Це симптом. Далі треба порівняти контент, статуси, redirects, внутрішні посилання, sitemap, hreflang, мобільні/десктопні варіанти та історію URL. Я шукаю не “чому Google не слухає тег”, а “який інший набір сигналів виглядає для системи послідовнішим”.

Таблиця 8.2. Діагностика розбіжності declared і selected canonical.

Симптом	Що перевірити першочергово	Типове рішення
Google вибирає параметричну URL	Внутрішні inlinks, sitemap, backlinks, response chain	Уніфікувати всі внутрішні сигнали на clean URL.
Google вибирає інший домен/host	Redirects, cross-domain canonical, дублікати, історія міграцій	Прибрати випадкові зовнішні сигнали, перевірити 301.
Google вибирає іншу мовну URL	hreflang, мова основного контенту, canonical кластер	Розвести локалі й зробити взаємні hreflang.
Google не приймає canonical на базову категорію	Схожість контенту фільтра і базової сторінки	Вирішити, чи фільтр реально дубль; canonical не використовувати як noindex.

8.7. Метрики canonicalization для великого сайту

На великому сайті canonical-аудит я зводжу до матриці “URL → final status → declared canonical → canonical final status → indexability → inlinks → sitemap → selected canonical”. Після цього проблеми

групуються шаблонами. Якщо 80 тисяч URL одного facet template canonical-яться через redirect, це одна системна помилка, а не 80 тисяч задач.

Корисний показник - Canonical Agreement Rate: частка indexable URL, для яких declared canonical збігається з очікуваною основною URL за правилами бізнесу. Це внутрішня метрика, не Google metric. Її цінність у регресійному контролі: після релізу можна побачити, що частка узгоджених URL впала з 99,4% до 92,1%, ще до того, як зміна проявиться у видимості.

8.8. Що ми знаємо точно

Google групує дублікати й обирає представника; redirects і rel="canonical" є сильними сигналами, sitemap - слабшим; сигнали можна комбінувати; canonical є підказкою, а не директивою; дублікати самі по собі не є спамом; регіональні варіанти однією мовою потребують узгодження canonical і hreflang.[1][2][4]

Не знаємо публічно: точну вагу кожного canonical-сигналу в конкретному кластері та формулу, за якою Google вирішує конфлікт. Тому “підсилити canonical ще трьома сигналами” - практична інженерна стратегія, а не відтворення закритого алгоритму.

Джерела до Глави 8

1. Google Search Central. What is canonicalization. https://developers.google.com/search/docs/crawling-indexing/canonicalization

2. Google Search Central. How to specify a canonical URL with rel="canonical" and other methods. https://developers.google.com/search/docs/crawling-indexing/consolidate-duplicate-urls

3. Google Search Central. Fix canonicalization issues. https://developers.google.com/search/docs/crawling-indexing/canonicalization-troubleshooting

4. Google Search Central. Managing multi-regional and multilingual sites. https://developers.google.com/search/docs/specialty/international/managing-multi-regional-sites

5. Google Search Central. Localized versions. https://developers.google.com/search/docs/specialty/international/localized-versions

ЧАСТИНА II / ЯК ПОШУКОВА СИСТЕМА БАЧИТЬ ВЕБ

ГЛАВА 9

Індексація: чому сканування не гарантує індексацію

Як відділити “Googlebot був на URL” від “документ є в індексі”, чому noindex, дублікати, soft 404, якість і технічні правила дають різні класи проблем, і як будувати індексаційний аудит без ритуалу “Request indexing”.

Стан даних і джерел: 14 вересня 2026 року

Індексація - найчастіше місце, де SEO плутає факт із наміром. Googlebot отримав 200 - значить сторінка має бути в індексі. URL є в sitemap - значить вона має бути в індексі. Ми натиснули Request indexing - значить система “зобов’язана” її додати. Жодне з цих тверджень не відповідає тому, як Google описує Search. Навіть повна відповідність технічним вимогам не гарантує crawl, index або serving. [1]

Рис. 9.1. Сканування є передумовою для багатьох індексаційних рішень, але не гарантією індексу.

Схема автора за Google Search Central How Search Works і Crawling & Indexing documentation.

9.1. Індекс - це не архів усього, що Googlebot побачив

Пошуковій системі немає сенсу тримати в активному індексі кожну технічно доступну URL. Варіанти параметрів, дублікати, порожні фільтри, тестові сторінки, soft 404, малокорисні комбінації programmatic templates і безкінечні календарі можуть бути доступні для сканування, але не створювати окремої пошукової цінності. Тому “Crawled - currently not indexed” треба трактувати як клас стану, а не як один дефект із одним фіксом.

9.2. noindex працює тільки тоді, коли crawler може його побачити

Google прямо вказує: noindex має бути доступний crawler. Якщо сторінку одночасно заборонено в robots.txt, Googlebot не отримує HTML і не бачить noindex; URL усе ще може бути відома з посилань та інколи з’являтися в результатах без нормального snippet.[2] Це класичний приклад конфлікту між керуванням crawl і керуванням index.

Таблиця 9.1. Чотири різні способи “прибрати URL” вирішують різні задачі.

Механізм	Що реально робить	Коли застосовувати
robots.txt	Обмежує запити crawler до URL; не є надійним способом видалення з індексу.	Коли не хочете витрачати crawl на простір URL або ресурси, які не треба отримувати.
noindex	Після crawl наказує не тримати документ у Search results.	Для доступних користувачу сторінок, які не повинні бути в пошуку.

Механізм	Що реально робить	Коли застосовувати
404/410	Повідомляє, що ресурсу немає. З часом URL випадає з індексу.	Сторінка реально видалена без релевантної заміни.
301/308	Перенаправляє користувача і дає сильний canonical-сигнал на нову URL.	Є еквівалентна нова адреса або міграція.

9.3. “Discovered” і “Crawled” - це різні вузькі місця

Якщо URL у стані Discovered - currently not indexed, перше питання - чи отримував її Googlebot. Якщо ні, ми ще на етапі discovery/crawl scheduling: внутрішні посилання, sitemap, серверна спроможність, crawl demand, надмірний URL-space. Якщо URL у Crawled - currently not indexed, crawler уже дійшов до документа, і бездумно “підсилювати sitemap” менш логічно. Треба перевіряти canonical cluster, soft 404, дублікати, якість шаблону, унікальність фактичних даних і те, чи сторінка взагалі заслуговує на окремий індексований документ.

ПОПЕРЕДЖЕННЯ
Не використовуйте кількість Indexed pages як KPI сама по собі. Сайт із 100 000 корисних індексованих URL може бути здоровішим за сайт із 3 млн URL, якщо решта 2,9 млн - фільтри, порожні комбінації та дублікати. KPI має бути “частка цінного URL-space, який проіндексований”, а не “індексувати якомога більше”.

9.4. Soft 404 - це не тільки сторінка з текстом “не знайдено”

Soft 404 виникає, коли HTTP-відповідь формально успішна, але зміст виглядає як відсутній або непридатний ресурс. Типові приклади: порожня категорія з 200, товар, що назавжди недоступний і не має альтернативи, сторінка внутрішнього пошуку без результатів, шаблон із одним заголовком і нульовим основним контентом. Важлива причина проблеми - системі доводиться окремо

інтерпретувати семантичний стан замість отримати коректний HTTP-статус.

9.5. Дублікат може бути “не проіндексований” тому, що індексується його представник

В Indexing reports багато URL, які SEO називає “випали з індексу”, насправді є alternate pages або дублікати. Це не обов’язково проблема. Якщо параметричні URL, print-версія і clean URL утворюють один canonical cluster, індексація одного представника є очікуваною поведінкою. Тут треба оцінювати кластер, а не добиватися індексації кожної URL окремо.

9.6. Якість у індексації: не перетворюйте невідоме на “ranking factor”

Google не публікує повну формулу індексаційного відбору. Тому я не пишу “сторінка не індексується через E-E-A-T” або “потрібно 800 слів”. Коректніше: Search Essentials прямо не гарантує index навіть для технічно валідної сторінки; Google оцінює контент і дублікати на етапах обробки; якщо цілі класи тонких або майже взаємозамінних сторінок системно не індексуються, це сильний сигнал переглянути саму цінність шаблону, а не шукати чарівний технічний прапорець. [1][3]

9.7. Індексаційна воронка: правильний знаменник

Таблиця 9.2. Які знаменники потрібні для нормального індексаційного звіту.

Метрика	**Чисельник**	**Правильний знаменник**
Indexable coverage	Indexable URL, що є в індексі	Усі URL, які бізнес дійсно хоче індексувати

Метрика	Чисельник	Правильний знаменник
Crawl-to-index rate	Проіндексовані URL із просканованої вибірки	Просканові indexable URL того самого шаблону/періоду
Sitemap index rate	Проіндексовані canonical URL із sitemap	Валідні 200 canonical URL у sitemap
Template index rate	Проіндексовані URL шаблону	Усі indexable URL цього шаблону

Я навмисно не ділю “indexed URLs” на “усі URLs, які знайшов crawler”. Якщо crawler знайшов 12 млн параметрів, а бізнес очікує 400 тисяч індексованих сторінок, такий знаменник нічого не говорить про якість індексації.

9.8. Sampling замість ручного URL Inspection

URL Inspection корисний для доказу стану конкретної сторінки, але 100 тисяч URL вручну не перевіряються. Я будую стратифіковану вибірку: шаблон, рівень внутрішніх посилань, нові/старі URL, sitemap/non-sitemap, canonical/non-canonical, країна/мова. Потім беру 50-200 URL із кожного значущого сегмента. Мета - не отримати “загальний % індексації”, а знайти клас, де ймовірність індексації різко відрізняється.

9.9. Request indexing - інструмент перевірки й прискорення одиничних змін, не API для масового індексу

Ручний запит на індексацію я використовую для важливої нової URL або після виправлення конкретної технічної помилки, коли хочу швидше запустити наступний цикл. Це не масштабна стратегія. Google прямо пише, що crawl та indexing займають час і не гарантуються; sitemap і нормальний internal graph - системні канали, ручна кнопка - допоміжний інструмент.[4]

9.10. Практичний порядок діагностики

Мій порядок: 1) чи URL повинна існувати як окремий searchable document; 2) final HTTP status; 3) robots/noindex; 4) canonical і дублікати; 5) rendered content; 6) внутрішні посилання та sitemap; 7) URL Inspection; 8) серверний лог; 9) порівняння шаблону з індексованими аналогами. Це майже завжди швидше, ніж починати з “додамо більше тексту” або “купимо посилання, щоб Google захотів індексувати”.

9.11. Що ми знаємо точно

Технічна доступність не гарантує index; noindex має бути доступний crawler; sitemap допомагає discovery, але не гарантує index або ranking; canonicalization може означати, що альтернативна URL не представлена окремо; Google не дає строків або гарантій для crawl/index.[1][2][4]

Джерела до Глави 9

1. Google Search Central. Search Essentials. https://developers.google.com/search/docs/essentials

2. Google Search Central. Block Search indexing with noindex. https://developers.google.com/search/docs/crawling-indexing/block-indexing

3. Google Search Central. In-depth guide to how Google Search works. https://developers.google.com/search/docs/fundamentals/how-search-works

4. Google Search Central. Crawling and indexing FAQ. https://developers.google.com/search/help/crawling-index-faq

5. Google Search Central. Crawling and indexing overview. https://developers.google.com/search/docs/crawling-indexing

ЧАСТИНА II / ЯК ПОШУКОВА СИСТЕМА БАЧИТЬ ВЕБ

ГЛАВА 10

Відбір, ранжування і показ: що відбувається після індексу

Чому індексація лише робить документ кандидатом, а не дає позицію; як розділяти relevance, ranking, SERP features, локаль і персональний контекст; і чому позиція - результат запиту, а не постійний атрибут URL.

Стан даних і джерел: 14 вересня 2026 року

Після індексації починається частина Search, яку SEO найчастіше стискає до слова “ранжування”. Насправді для практичної діагностики корисно розділяти щонайменше три рішення: які документи витягнути з індексу як кандидатів, як упорядкувати кандидатів і що саме показати користувачу в конкретному інтерфейсі. Google описує ranking systems як автоматизовані системи, що аналізують багато факторів і сигналів серед величезного індексу, щоб знайти релевантні й корисні результати.[1]

Після індексу: відбір, ранжування і показ - три різні рішення

Позиція - не властивість URL сама по собі. Це результат конкретного запиту, набору конкурентів, контексту показу та систем, що спрацювали в цей момент.

Рис. 10.1. Відбір кандидатів, ранжування й показ - різні етапи одного запиту.

Схема автора за Google Search ranking systems guide та How Search Works.

10.1. Позиція не зберігається всередині сторінки

Фраза “ця сторінка має позицію 3” зручна, але неточна. Вона має позицію для конкретного query, country, device, search surface, часу й набору конкурентів. Змінився query interpretation - змінився candidate set. З’явився local pack, shopping block або AI Overview - змінилася геометрія SERP. Змінилася країна - змінився intent і набір джерел. Тому rank tracker дає спостереження, а не внутрішній score документа.

10.2. Retrieval before ranking: спочатку треба потрапити в набір кандидатів

Якщо документ не витягується для конкретної інформаційної потреби, говорити про “підвищення позиції” рано. Для класичного пошуку цей шар менш видимий зовні, але в AI Search Google уже прямо описує RAG та query fan-out: системи виконують пов’язані пошуки, отримують релевантні сторінки з індексу, а потім використовують їх як джерела.[2] Механіка generative Search зробила стару проблему очевиднішою: ranking і retrieval не тотожні.

10.3. “Фактор ранжування” - занадто груба одиниця для аналізу

Google пише про many factors and signals, але це не ліцензія перетворювати будь-яку кореляцію на “ranking factor”. Page speed, links, freshness, language, location, meaning of query, content relevance - різні системи можуть бути важливими в різних класах запитів. Практична робота Senior SEO не полягає в пошуку глобальної ваги сигналу X. Вона полягає у визначенні, який клас проблем обмежує конкретний набір сторінок у конкретній SERP.

Таблиця 10.1. П’ять рівнів, які не варто зводити до одного слова “ranking”.

Рівень	Питання	Приклад помилки в аналізі
Eligibility	Чи може документ узагалі бути показаний?	noindex/robots/ manual action трактують як “просіли позиції”.
Retrieval	Чи потрапляє документ до кандидатів для intent?	Пишуть ще 500 слів, хоча сторінка відповідає іншій задачі.
Ranking	Як документ оцінюється відносно інших кандидатів?	DR замінює аналіз самої SERP.
Serving	Що показати в конкретній країні, пристрої та форматі?	Один desktop rank tracker видають за весь Search.
Presentation	Як результат виглядає і який CTR отримує?	Позиція стабільна, але CTR падає через новий SERP feature.

10.4. Query interpretation: “ключове слово” не дорівнює наміру

Одна й та сама фраза може мати кілька можливих задач. Google може інтерпретувати “jaguar” як авто, тварину або бренд залежно від контексту; у комерційній SERP “best X” може означати огляд, список, категорію або marketplace. Тому я не починаю оптимізацію з TF-IDF або “додамо exact match”. Спочатку фіксую склад SERP: типи сторінок, домінуючі intents, features, бренди, свіжість, локальність, комерційність і те, чи стабільний цей склад у часі.

10.5. Page-level і site-wide сигнали

У ranking systems guide Google прямо уточнює: системи працюють переважно на рівні сторінок, але є також site-wide signals і classifiers. Це важлива протиотрута від двох крайнощів: “Google ранжує домен, сторінка не важлива” і “кожна URL повністю незалежна”. Практично я аналізую обидва рівні: конкретний документ і властивості

класу/сайту, але не приписую третім метрикам типу DR роль внутрішнього site score Google.[1]

10.6. Relevance без достатньої якості не гарантує top positions; quality без relevance теж

Сторінка може бути “дуже якісною” за редакційними мірками, але відповідати не на той intent. І навпаки, exact-match сторінка може бути релевантною словесно, але не давати користувачу достатньої інформації, доказів або функціональності. У практичному конкурентному аналізі я розкладаю gap на дві осі: semantic/task fit і evidence/value. Це дає кращі рішення, ніж “конкурент має 2400 слів, у нас 1800”.

10.7. SERP features змінюють видимість без зміни rank

Класична position metric не вимірює площу екрана. Position 1 під AI Overview, двома ads, shopping carousel і local pack має іншу економіку, ніж position 1 у простій видачі. Саме тому в попередніх главах ми розділили visibility і click yield. На цьому етапі важливо додати ще один принцип: serving system визначає не тільки порядок blue links, а й форму відповіді.

10.8. Локаль, мова й пристрій - частина serving, не “шум”

Для міжнародного SEO я ніколи не зводжу позиції різних країн у одну середню. Google документує окремі механізми локалізації й hreflang, а mobile/desktop можуть мати різну видачу. Середня позиція 4,8 може складатися з position 2 в Іспанії й position 18 у Мексиці. Для бізнесу це дві різні задачі, а не одна “позиція 4,8”.

10.9. Як я порівнюю конкурентів у SERP

Таблиця 10.2. Конкурентний SERP-аудит на рівні документа.

Вісь	Що фіксую	Навіщо
Тип сторінки	категорія, review, tool, product, forum, publisher	Не конкурувати іншим форматом без причини.
Задача	купити, порівняти, перевірити, навчитися, знайти бренд	Визначити dominant і secondary intents.
Доказовість	дані, джерела, screenshots, власні тести, експерти	Знайти інформаційну перевагу, а не обсяг тексту.
Authority context	посилання, бренд, згадки, тематичний граф	Зрозуміти конкурентне середовище без магічного DR threshold.
SERP features	AI, snippets, video, local, shopping, PAA	Оцінити реальну click opportunity.
Стабільність	які домени/типи сторінок тримаються 4-8 тижнів	Відділити структурний патерн від випадкового snapshot.

10.10. Кореляційні дослідження: корисні для гіпотез, небезпечні як рецепт

Якщо top-3 pages у середньому мають більше links, швидші Core Web Vitals або довший текст, це не показує, що різниця в цій метриці спричинила ranking. Рейтингові вибірки за визначенням змішані з брендом, типом сайту, intent, віком URL і десятками інших змінних. Я використовую такі studies для генерації гіпотез, а рішення підтверджую своїми даними, контрольованими змінами або хоча б сегментацією.

10.11. Модель діагностики: eligibility → retrieval → ranking → serving → click

Ця послідовність економить години. Якщо page excluded by noindex - не аналізуємо anchors. Якщо indexed, але ніде не з'являється навіть за вузьким task-fit query - перевіряємо retrieval/relevance. Якщо стабільно position 8-12 для правильного intent - дивимося competitive gap. Якщо position 2, але clicks падають - дивимося SERP composition і CTR. Одна цифра "organic traffic -30%" не містить діагнозу.

10.12. Що ми знаємо точно

Google використовує автоматизовані ranking systems із багатьма signals; page-level і site-wide signals співіснують; Search Essentials не гарантує serving навіть для технічно валідної сторінки; у generative features retrieval і query fan-out описані публічно; Search results динамічні й залежать від запиту та контексту.[1][2][3]

Не знаємо: повну формулу rank score, ваги сигналів, точний candidate set і всі умови, за яких окрема система активується. Тому професійна модель - не "вивчити 200 ranking factors", а відділити спостережувані стани системи й тестувати гіпотези на власних даних.

Джерела до Глави 10

1. Google Search Central. A guide to Google Search ranking systems. https://developers.google.com/search/docs/appearance/ranking-systems-guide

2. Google Search Central. Optimizing for generative AI features on Google Search. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

3. Google Search Central. Search Essentials. https://developers.google.com/search/docs/essentials

4. Google Search Central. In-depth guide to how Google Search works. https://developers.google.com/search/docs/fundamentals/how-search-works

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 11

Інформаційна архітектура: як перетворити сайт на зрозумілий граф

Чому меню й папки URL не дорівнюють архітектурі, як моделювати зв’язки між money pages і supporting content, як вимірювати глибину та сирітські сторінки і чому кожна важлива URL повинна мати зрозумілий шлях із графа.

Стан даних і джерел: 14 вересня 2026 року

Інформаційну архітектуру часто малюють як дерево: Home → Category → Subcategory → Product. Це корисна презентаційна схема, але технічно сайт є графом. Одна сторінка може мати посилання з меню, breadcrumbs, тематичного гайда, comparison page, footer і блоку related products. Google прямо пояснює, що аналізує зв’язки між сторінками і може використовувати кількість кроків до сторінки та кількість внутрішніх посилань як сигнали відносної важливості.[1]

Рис. 11.1. Практична архітектура сайту є графом зв’язків, а не лише деревом директорій.

Схема автора на основі Google Search Central ecommerce site structure та link best practices.

11.1. Архітектура починається не з URL, а з набору задач

Я спочатку виписую типи документів, які бізнес реально повинен мати: category, product/service, comparison, review, how-to, research, legal/support, locale variants. Потім визначаю, як користувач переходить між ними. URL structure є реалізацією, але не первинною моделлю. Google окремо зазначає, що не покладається на структуру URL як на головний спосіб визначити структуру ecommerce-сайту, а аналізує links between pages.[1]

11.2. Кожна важлива URL повинна мати хоча б один нормальний внутрішній шлях

Google у link best practices прямо радить, щоб на кожну важливу URL посилалася принаймні одна інша сторінка сайту.[2] Для мене це мінімум, а не оптимум. Money page, яку можна знайти лише в sitemap, технічно існує, але архітектурно майже не інтегрована. На великому сайті я рахую не тільки orphan pages, а й кількість незалежних шляхів до важливого шаблону.

Таблиця 11.1. Архітектурні метрики, які мають сенс.

Метрика	Що вимірює	Як читати
Click depth	мінімальну кількість переходів від вибраних входів	Порівнювати всередині типів сторінок, а не шукати магічний максимум.
Internal inlinks	скільки crawlable посилань веде на URL	Висока кількість не гарантує ranking, але показує пріоритет у власному графі.
Unique source templates	з яких типів сторінок приходять посилання	Один sitewide footer-link ≠ кілька контекстних шляхів.

Метрика	Що вимірює	Як читати
Orphan rate	частку потрібних URL без внутрішніх inlinks	Рахувати від expected indexable inventory, а не від усіх URL crawl.
Hub coverage	частку money pages, пов’язаних із логічним hub	Допомагає знаходити контент, який існує, але не має тематичного контексту.

11.3. Глибина не має універсального порога

Порада “усе має бути не далі трьох кліків” зручна, але не універсальна. Для сайту на 500 URL це може бути природно; для каталогу на 5 млн SKU - ні. Я дивлюся на distribution: яка median depth у money pages, які URL формують довгий хвіст, чи збігаються deep pages із poor crawl/index performance, чи глибина виникла через бізнес-логіку або через поламану навігацію.

11.4. Хаби - не “SEO category pages”, а точки стиснення графа

Хаб потрібен, коли він скорочує шлях між пов’язаними документами і дає користувачу зрозумілий огляд простору. Хороший hub може бути category, comparison landing, thematic guide або brand page. Поганий hub - сторінка, створена тільки щоб перелінкувати 100 статей без власної задачі. Якщо сторінка не допомагає навігації чи прийняттю рішення, вона додає URL-space, а не архітектуру.

11.5. Фасетна навігація - архітектура, помножена комбінаторикою

Фільтри можуть створити корисні landing pages для стабільного попиту, але кожна додаткова dimension множить URL-space. Google називає faceted navigation найпоширенішим джерелом overcrawl-проблем і радить або блокувати непотрібні комбінації, або дуже

дисципліновано керувати тими, які мають сканuватися.[3] Тому рішення “index/noindex” повинно виходити з inventory попиту й бізнес-цінності, а не з того, що CMS може згенерувати URL.

11.6. Архітектура для affiliate: money pages не повинні висіти окремими островами

У affiliate-проєктах типова крайність - 90% внутрішньої ваги спрямувати на review/bonus pages, а supporting content існує лише як трафіковий long-tail. Я будую двосторонні зв’язки: informational → commercial для переходу до рішення, commercial → explanatory для доказів і зняття заперечень, brand hub → конкретні products/offers, comparative pages → individual reviews. Це корисно користувачу й одночасно робить граф семантично зрозумілим.

11.7. Практичний аудит архітектури

Я беру crawl із inlinks/outlinks, sitemap inventory, GSC clicks/impressions і business value. Для кожної indexable URL додаю type, parent concept, click depth, internal inlinks, unique linking templates, traffic, conversions і status. Далі шукаю чотири класи: важливі URL без graph support; надмірно сильні utility pages; дубльовані хаби; глибокі класи URL із слабким crawl/index coverage. Це перетворює “покращити перелінковку” на конкретну чергу робіт.

НОТАТКА З ПРАКТИКИ
Не оптимізуйте архітектуру лише під crawler. Якщо після “SEO-перелінковки” користувачеві стало важче зрозуміти, куди натискати, ви, ймовірно, створили граф для робота, а не інформаційну архітектуру.

11.8. Що ми знаємо точно

Google використовує links для discovery і як сигнал релевантності; crawler нормально витягує стандартні <a href>; Google аналізує зв’язки сторінок і відносну важливість; для ecommerce рекомендує навігаційні шляхи category → subcategory → products; на кожну

важливу URL має посилатися інша сторінка.[1][2] Точну внутрішню вагу кожного internal link Google не публікує.

Джерела до Глави 11

1. Google Search Central. Help Google understand your ecommerce site structure. https://developers.google.com/search/docs/specialty/ecommerce/help-google-understand-your-ecommerce-site-structure

2. Google Search Central. Link best practices. https://developers.google.com/search/docs/crawling-indexing/links-crawlable

3. Google Search Central. Faceted navigation guidance. https://developers.google.com/search/blog/2024/12/crawling-december-faceted-nav

4. Google Search Central. URL structure best practices. https://developers.google.com/search/docs/crawling-indexing/url-structure

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 12

Попит і аналіз ключових слів: від списку фраз до простору запитів

Як перестати збирати 100 000 keywords “про запас”, відділити формулювання від задачі, поєднати GSC, SERP, third-party volume та бізнес-дані і вирішити, які запити заслуговують на окремі сторінки.

Стан даних і джерел: 14 вересня 2026 року

Keyword research у слабкій реалізації - це експорт із сервісу, фільтр KD < 10 і кластеризація за словами. У сильній реалізації це модель попиту. Ключове слово лише один спосіб сформулювати задачу. Одна потреба породжує десятки формулювань, а одна фраза може мати кілька intent. Тому результат research - не CSV із volume, а карта query space → task → page type → business value → evidence.

Рис. 12.1. Робоча одиниця keyword research - не фраза, а стабільна задача та її місце в архітектурі.

12.1. Джерела попиту мають різні систематичні викривлення

Таблиця 12.1. Джерело keyword data визначає, що ви взагалі здатні побачити.

Джерело	Сильна сторона	Сліпа зона
GSC	реальні impressions/clicks вашого сайту	не показує попит, де сайт не мав видимості; частина queries анонімізована
Keyword tools	широкий discovery і сторонні volume/keywords	оцінки, власні бази й моделі; не census реальних запитів
SERP	показує фактичний склад результатів та intent	snapshot; залежить від locale/time/personalization
Internal search	мову вже наявних користувачів і проблеми навігації	лише ваша аудиторія
Sales/support	реальні заперечення й формулювання покупців	малий і нерівномірний sample
Google Trends	відносний тренд і сезонність	не дає абсолютного search volume

Search Essentials прямо радить використовувати слова, якими люди шукають ваш контент, у помітних місцях сторінки.[1] Але це не означає “створити URL під кожну фразу”. Сучасні системи розуміють релевантність без exact-match для кожної варіації, а Google окремо попереджає не створювати сторінки під кожний fan-out query лише для маніпуляції Search.[2]

12.2. Нормалізація починається з модифікаторів

Для комерційного кластера я розкладаю запити на base entity і modifiers: brand, product class, location, price, bonus, promo, app, review, comparison, safety, payment method, withdrawal, registration, alternative, problem. Це дозволяє побачити, які dimension є системними, а які

випадковими. Якщо “brand + review”, “brand + bonus” і “brand + app” мають різний SERP і різну задачу, це кандидати на окремі page types. Якщо SERP майже ідентичний і користувач очікує одну комплексну сторінку, розмноження URL створить cannibalization.

12.3. Кластеризація за SERP overlap корисна, але не є істиною

SERP-overlap clustering відповідає на практичне питання: чи Google зараз часто показує ті самі URL для двох queries. Але він залежить від country, device, date і нестабільності SERP. Я використовую overlap як evidence, а не як єдиний rule. Для high-value cluster перевіряю вручну: чи однакова задача, чи однаковий тип сторінки, чи є окремий conversion path.

12.4. Volume без click opportunity переоцінює informational head terms

Після AI Overviews і багатьох SERP features 10 000 monthly searches не дорівнюють 10 000 потенційних organic visits. Я додаю до keyword model SERP click opportunity: ads, local, shopping, featured snippet, AI, video, brand dominance. Це не точний CTR forecast, а корекція здорового глузду. Для affiliate query із volume 700 і сильним commercial intent реальна цінність може бути вищою, ніж для informational query на 20 000, який закривається у SERP.

12.5. Business value - окрема вісь, а не колонка CPC

CPC корисний як сторонній ринковий сигнал, але не замінює власну економіку. Я оцінюю conversion probability, revenue per conversion, approval rate, LTV або affiliate EPC, залежно від моделі. Потім пріоритизую не keywords, а clusters. Просте expected value: search opportunity × expected CTR band × conversion rate × value per conversion. У кожного множника має бути діапазон, а не псевдоточне число.

12.6. Новий сайт і діючий сайт потребують різного research

На новому проєкті я будую зовнішню карту попиту: конкуренти, tools, SERP, forums, product taxonomy. На діючому сайті пріоритет зміщується до first-party data: GSC queries/pages, internal search, landing conversions, lost impressions, pages із position 5-20, clusters із великою різницею між impressions і clicks. Діючий сайт уже має власний "датчик попиту", ігнорувати його заради чужого volume - помилка.

12.7. Page decision: коли запит заслуговує на окрему URL

Таблиця 12.2. Моя перевірка перед створенням нової SEO-сторінки.

Питання	**Так - аргумент за окрему URL**	**Ні - аргумент за консолідацію**
Є окрема задача користувача?	Окремий decision / workflow / intent	Лише інше формулювання
SERP суттєво відрізняється?	Інший тип результатів або конкуренти	Ті самі URL і формат
Є окремі дані/контент?	Сторінка має власну інформаційну цінність	Шаблон просто міняє keyword
Є бізнес-цінність?	Окремий conversion path або важлива visibility	Немає окремого результату
Сторінку можна нормально інтегрувати?	Є місце в IA та internal links	URL буде сиротою заради long-tail

12.8. "KD 0" не означає "легко"

Third-party KD - модель конкретного сервісу. Вона може недооцінювати brand SERP, regulatory vertical, fresh query або SERP з сильними доменами й малою кількістю links. Я ніколи не використовую KD як gate. На low-KD запитах дивлюся, чому SERP така:

чи це справді слабкі сторінки, чи інструмент просто не вимірює фактор, який визначає конкуренцію.

12.9. Вихідний артефакт research

Мій фінальний dataset має cluster_id, primary task, query examples, locale, page type, current URL, proposed URL, demand range, SERP features, top competitors, business value, evidence available, content gap, internal hub, priority і status. Це вже база для architecture, content operations і measurement. CSV “keyword-volume-KD” сам по собі не є SEO-стратегією.

Джерела до Глави 12

1. Google Search Central. Search Essentials.
https://developers.google.com/search/docs/essentials

2. Google Search Central. Optimizing for generative AI features.
https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

3. Google Search Console. Performance data filtering and limits.
https://developers.google.com/search/blog/2022/10/performance-data-deep-dive

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 13

Намір, тема та сутності: як моделювати реальну потребу користувача

Чому intent не зводиться до “informational/commercial”, entity не дорівнює списку NLP-слів, а topical authority не вимірюється кількістю статей. Практична модель для SERP, контенту й архітектури.

Стан даних і джерел: 14 вересня 2026 року

Три поняття - intent, topic і entity - часто змішують в одну “семантику”. У результаті SEO отримує списки сутностей, які треба “вставити”, класифікацію intent на чотири кольори і content hub із 80 статей, бо “треба покрити тему”. Я розділяю їх. Intent - дія або результат, якого хоче користувач. Topic - предметна область. Entity - конкретний об’єкт, який можна однозначно відрізнити від інших об’єктів.

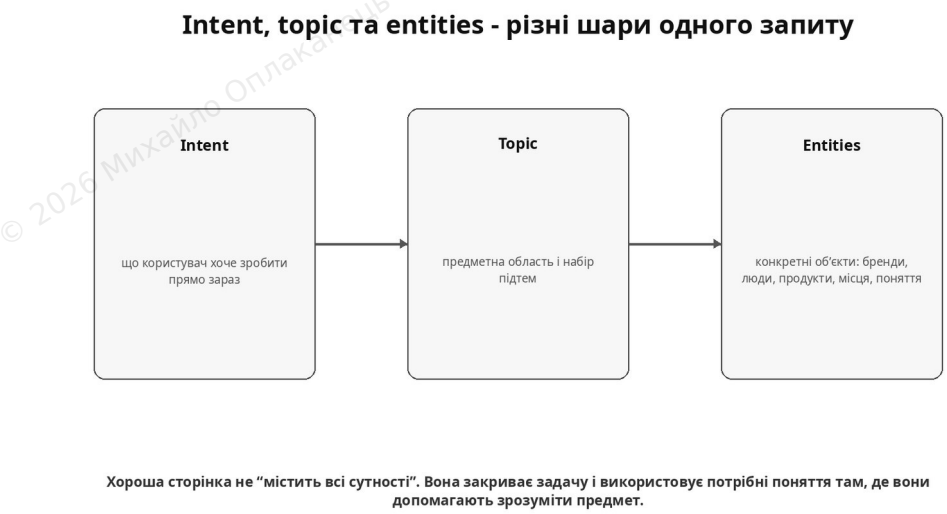


Рис. 13.1. Intent, topic і entities вирішують різні задачі аналізу.

13.1. Intent - не лейбл, а набір очікувань до відповіді

“Commercial investigation” нічого не каже редактору, якщо не розкласти очікування. Для “best online casino Spain” користувач може очікувати список операторів, критерії безпеки, бонуси, платежі, legal context і швидкий шлях до сайту. Для “casino X review” очікування інші: конкретний бренд, trust, withdrawal, KYC, bonus terms, app, support. Обидва commercial, але content object різний.

13.2. SERP - найкращий зовнішній датчик інтерпретації intent, але не абсолютна правда

Я фіксую dominant page types, repeated subtopics, brands/entities, features і те, що відсутнє. Але не копіюю SERP як технічне завдання. Поточний SERP може бути слабким, змішаним або сформованим через дефіцит хороших документів. Завдання Senior SEO - зрозуміти pattern і перевірити, де є opportunity дати кращу відповідь.

13.3. Entities потрібні для точності, а не для “density”

Якщо сторінка про iPhone 18 Pro, логічно коректно називати Apple, модель, chipset, camera system, iOS, storage variants, relevant predecessors. Але “entity optimization” не означає вставити 40 пов’язаних імен. Google structured data documentation описує entity-like markup як явні clues про значення сторінки, але не обіцяє ranking boost за кількість entities.[1]

13.4. Topic coverage не дорівнює кількості URL

Якщо про тему можна написати 200 статей, це не означає, що бізнес повинен мати 200 URL. Покриття я оцінюю за задачами: чи є відповідь на primary commercial tasks, support questions, comparison, trust, troubleshooting, post-purchase. Якщо 20 сторінок дають повне покриття, додаткові 180 можуть бути просто scaled sameness.

Таблиця 13.1. Як я розрізняю три семантичні шари.

Шар	Питання	SEO-рішення
Intent	Який результат потрібен користувачу?	тип сторінки, структура, CTA, рівень деталізації
Topic	Який предмет і межі матеріалу?	кластер, supporting content, coverage
Entities	Про які конкретні об’єкти й поняття йдеться?	точність фактів, structured data, disambiguation, internal links

13.5. Multi-intent SERP потребує рішення, а не автоматичного splitting

Якщо SERP містить product pages, guides і videos, є кілька стратегій. Створити окремі сторінки під intents; зробити один сильний mixed page; або сфокусуватися на intent із найвищою business value. Я дивлюся на стабільність сегментів і conversion. Якщо Google роками тримає два типи результатів, це може бути реальний multi-intent, а не “канібалізація”.

13.6. Query fan-out робить task modeling важливішим

У AI Search Google прямо описує query fan-out: модель запускає пов’язані queries для підтем.[2] Практичний висновок не “створимо сторінку під кожний fan-out query”. Навпаки: треба мати документи, які добре відповідають на логічні sub-tasks і містять достатньо конкретики, щоб бути retrieved як джерело. Це ще один аргумент мислити задачами, а не exact-match phrases.

13.7. Entity consistency для бренду

Для бренду я прагну однакових базових facts на first-party sources: name, legal/brand relation, URL, logo, contacts, location, product naming. Organization structured data може давати Google явні clues про організацію і використовуватися в knowledge panels та attribution

surfaces.[3] Але markup не виправить хаос, якщо на різних сторінках різні назви й контакти.

13.8. Практичний шаблон semantic brief

Таблиця 13.2. Brief без “вставити 27 keywords”.

Блок	Що має бути
Primary task	одне речення: що користувач хоче завершити
Secondary tasks	що треба перевірити/зрозуміти до рішення
Entities	обов’язкові конкретні об’єкти й терміни для точності
Evidence	які дані, джерела, screenshots, tools або tests додадуть перевагу
Out of scope	що свідомо не покриваємо, щоб сторінка не стала енциклопедією
Internal graph	звідки приходить користувач і куди логічно піде далі

13.9. Що ми знаємо точно

Google використовує системи для розуміння meaning і relevance, structured data може давати явні clues про зміст, а generative Search використовує query fan-out. Не знаємо публічно: універсального “entity score”, необхідної кількості сутностей або формули topical authority. Тому такі терміни я використовую як робочі моделі, а не офіційні metrics Google.

Джерела до Глави 13

1. Google Search Central. Introduction to structured data. https://developers.google.com/search/docs/appearance/structured-data/intro-structured-data

2. Google Search Central. Optimizing for generative AI features. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

3. Google Search Central. Organization structured data.
https://developers.google.com/search/blog/2023/11/introducing-organization-markup

4. Google Search Central. Ranking systems guide.
https://developers.google.com/search/docs/appearance/ranking-systems-guide

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 14

Контентне SEO: інформаційна перевага замість переписування конкурентів

Як будувати матеріал, який не є ще одним переказом top-10: evidence, information gain, first-party data, freshness, consolidation, шаблони й контентні операції у світі дешевого AI-тексту.

Стан даних і джерел: 14 вересня 2026 року

LLM зробили дешевим не “контент”, а виробництво граматично нормального тексту. Це велика різниця. Переписати 10 результатів, скласти список порад і додати вступ тепер можна майже без маржинальної вартості. Отже, саме цей тип output перестає бути конкурентною перевагою. Google у новому AI optimization guide окремо робить акцент на valuable, unique, non-commodity content.[1]

Рис. 14.1. Чим більше інформацію можна відтворити без доступу до вашого досвіду або даних, тим менша її захисна перевага.

14.1. Original information - не декоративний “експертний коментар”

Google helpful content guide прямо пропонує перевіряти, чи дає матеріал original information, reporting, research або analysis.[2] Для мене це практичний тест: якщо конкурент може скопіювати всю цінність сторінки, не маючи доступу до наших даних, процесів, експертів або продукту, moat низький. Не кожна сторінка повинна бути дослідженням, але найважливіші сторінки мають містити щось, крім переказу загальнодоступного.

14.2. П’ять джерел інформаційної переваги

Таблиця 14.1. Що створює non-commodity content на практиці.

Джерело	**Приклад**	**Чому це важко відтворити**
First-party data	власна вибірка GSC, sales, products, customer behavior	дані недоступні конкуренту
Original experiment	SEO test, benchmark, controlled comparison	потрібні час, методологія й повторення
Operational experience	реальна міграція, recovery, запуск GEO	знання failure modes, яких немає в документації
Tool / dataset	калькулятор, база, API, template	корисність існує поза текстом
Primary-source synthesis	точне порівняння документації, laws, specs	дорога перевірка фактів і підтримка актуальності

14.3. “Більше слів” не є інформаційною перевагою

Сторінка на 5000 слів може мати нижчу корисність, ніж таблиця на 800 слів, якщо решта - повторення. Я оцінюю coverage через questions/tasks, а не word count. Якщо користувач хоче порівняти три

продукти, найціннішим може бути decision table, test methodology і висновок з обмеженнями.

14.4. Контентний brief повинен визначати evidence

У brief я додаю поле “Evidence required”. Для review це можуть бути screenshots продукту, actual terms, payments, withdrawal test, support response. Для technical guide - code tested on sample, source documentation, before/after. Для research - sample, period, inclusion criteria, calculation. Це дисциплінує команду краще, ніж список “LSI keywords”.

14.5. AI-контент: оцінюємо production process і output

Google spam policies прямо говорять, що scaled content abuse визначається primary purpose і low value, незалежно від того, створено контент AI, людиною чи автоматизацією.[3] Тому “AI content penalty” як загальна теза некоректна. Ризик - масове виробництво взаємозамінних сторінок, factual errors, outdated facts і відсутність added value.

МІФ
“Щоб AI-текст не визначався, треба зробити його менш структурованим”. Ні. Пошукова цінність не виникає від імітації людських стилістичних випадковостей. Вона виникає від коректності, корисності, доказів і того, що сторінка реально робить для користувача.

14.6. Freshness - не автоматична перевага нової дати

Я розділяю factual freshness і cosmetic freshness. Якщо бонус, ціна, закон, API, product spec або supported feature змінилися, документ треба оновити. Якщо тема фундаментальна, дата “Updated today” без змін не додає інформації. Для large content inventory корисно мати freshness SLA за типом факту, а не за сторінкою в цілому.

14.7. Content decay треба діагностувати по components

Падіння сторінки може бути через demand shift, SERP composition, competitor improvement, outdated section, broken links, changed product або site-wide issue. Тому “оновити статтю” - не діагноз. Я порівнюю query mix, sections, competitors, links, conversion і freshness-sensitive facts. Іноді найкраще рішення - нічого не дописувати, а видалити 40% застарілого.

14.8. Consolidation сильніша за нескінченне оновлення слабких URL

Коли 4 статті відповідають на один intent, мають слабкий traffic і повторюють одна одну, я частіше консолідую їх у один сильний document, переношу унікальні fragments, роблю redirects і оновлюю internal links. Це не універсальне правило: якщо intents різні, merger зашкодить. Рішення повинне виходити з SERP і task model.

14.9. Контентна система повинна мати QA до публікації

Таблиця 14.2. Мінімальний QA для сторінки, де факти мають значення.

Шар	Перевірка
Факти	чи кожне time-sensitive твердження має джерело й дату
Entities	правильні бренди, моделі, країни, валюти, units
Intent	чи сторінка завершує primary task без зайвого пошуку
Evidence	що тут є такого, чого немає в commodity SERP
Internal links	чи сторінка вбудована в граф і має логічні next steps
Conversion	чи CTA відповідає intent, а не заважає відповіді
Updateability	які поля треба перевіряти через 30/90/180 днів

14.10. Що ми знаємо точно

Google рекомендує helpful, reliable, people-first content і прямо запитує про original information/research/analysis; 2026 AI guide підкреслює valuable, unique, non-commodity content; scaled content abuse залежить від мети і value, а не від факту використання AI.[1][2][3] Не існує підтвердженого Google metric “information gain score”, який SEO може виміряти зовнішнім сервісом.

Джерела до Глави 14

1. Google Search Central. Optimizing for generative AI features. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

2. Google Search Central. Creating helpful, reliable, people-first content. https://developers.google.com/search/docs/fundamentals/creating-helpful-content

3. Google Search Central. Spam policies - scaled content abuse. https://developers.google.com/search/docs/essentials/spam-policies

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 15

Внутрішні посилання: виявлення URL, PageRank і тематичні зв’язки

Як будувати internal linking не списком “додайте 5 посилань”, а керованим графом: crawlable href, anchors, хаби, orphan pages, sitewide links, пріоритизація money pages і вимірювання змін.

Стан даних і джерел: 14 вересня 2026 року

Внутрішні посилання одночасно допомагають discovery, передають відносну важливість у власному графі й дають контекст зв’язку між сторінками. Якщо зводити їх до “вставити keyword у anchor”, дві третини системи зникає. Google прямо пише, що uses links as a signal of relevance і для discovery, a crawlable link зазвичай має бути <a href>.[1]

Рис. 15.1. Внутрішнє посилання одночасно впливає на discovery, graph priority і контекст.

15.1. Починайте з expected graph, а не з автоматичного плагіна

Я спочатку описую правила: product → category, category → key products, guide → relevant commercial, brand hub → review/bonus/app, comparison → individual entities, article → deeper explainer. Після цього автоматизація може шукати відсутні edges. Якщо почати з “link every keyword mention”, сайт швидко перетвориться на сітку шуму.

15.2. Crawlable link - технічна умова, а не дизайнерське рішення

Кнопка з onclick, div із router event або текстова URL не гарантовано є посиланням для crawler. Google документує стандартний <a href> як надійний формат.[1] На SPA я перевіряю DOM після render, але для критичної навігації не хочу залежати від нестандартного JS-handler.

15.3. Anchor text повинен описувати destination

Хороший anchor зменшує невизначеність: “умови бонусу Locowin” кращий за “детальніше”, якщо саме це destination. Але exact-match у кожному місці не потрібен. Я прагну природного variation, який відповідає контексту. Anchor - semantic hint і UX element, а не поле для stuffing.

15.4. Sitewide links мають іншу роль, ніж contextual links

Таблиця 15.1. Не всі внутрішні посилання еквівалентні за функцією.

Тип	Сильна сторона	Ризик
Main navigation	стабільний discovery важливих hubs	обмежена кількість слотів, sitewide шум
Breadcrumbs	ієрархія й шлях назад	не створює lateral topical connections
Contextual	точний семантичний зв’язок	можна переспамити шаблонними anchors

Тип	Сильна сторона	Ризик
Related modules	масштабоване cross-linking	алгоритм може лінкувати нерелевантне
Footer	utility/global access	не варто перетворювати на keyword directory

15.5. Internal PageRank - модель, не видима Google-метрика

Класична модель PageRank показує важливу властивість: link graph перерозподіляє вагу між nodes. Внутрішній граф - частина цього web graph. Але SEO не бачить внутрішній Google PageRank. Тому власні “internal PR” у crawler - модель за певними припущеннями. Я використовую її порівняльно: які сторінки надмірно центральні, які майже відрізані, як змінюється distribution після нового меню.

15.6. Link sculpting через nofollow - погана заміна архітектури

Якщо сторінка не варта crawl або internal importance, правильніше переглянути сам link/URL, а не масово ставити nofollow для “збереження PageRank”. Nofollow є hint для qualifying links і має нормальні use cases для untrusted/sponsored зовнішніх links, але не є сучасним універсальним інструментом sculpting внутрішнього графа.

15.7. Як я знаходжу opportunities

Я join-ю GSC pages із crawl graph. Найцікавіші групи: position 5-20 + high business value + low contextual inlinks; pages із clicks, але без hub links; pages, що канібалізують один cluster і лінкуються однаково; new pages без входів із authority hubs. Потім перевіряю, чи link корисний користувачу. Якщо ні - не додаю його лише заради метрики.

15.8. Вимірювання після зміни internal linking

Таблиця 15.2. Що вимірювати після системної перелінковки.

Рівень	Метрика	Очікування
Crawl	Googlebot requests / discovery latency	важливі URL знаходяться й refresh-яться швидше
Graph	inlinks, depth, modeled internal PR	зростає підтримка target pages без хаосу
Search	impressions, ranking distribution, indexed coverage	ефект оцінюється сегментом, не одним URL
UX	click-through між сторінками	користувач реально використовує нові переходи
Business	assisted conversions	лінки допомагають руху до рішення, а не лише crawler

15.9. Масштабована автоматизація

Для великого сайту я генерую candidate links на основі entities/topics, але не публікую їх без правил. Фільтри: не link to same URL; не більше N links із блока; target indexable 200 canonical; relevance threshold; diversity targets; cap sitewide repetition. AI добре генерує candidates і anchors, але QA має перевіряти destination і context.

15.10. Що ми знаємо точно

Google використовує links для discovery і relevancy; crawlable <a href> є рекомендованим форматом; на кожну важливу URL має посилатися хоча б одна сторінка; ecommerce guidance прямо пов’язує internal links із relative importance.[1][2] Точну величину PageRank, що передається конкретним внутрішнім посиланням, ми не бачимо.

Джерела до Глави 15

1. Google Search Central. Link best practices. https://developers.google.com/search/docs/crawling-indexing/links-crawlable

2. Google Search Central. Ecommerce site structure. https://developers.google.com/search/docs/specialty/ecommerce/help-google-understand-your-ecommerce-site-structure

3. Google Search Essentials. https://developers.google.com/search/docs/essentials

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 16

Технічне SEO як система, а не чекліст

Як перейти від аудиту на 200 пунктів до причинно-наслідкової моделі: URL space → crawl → render → canonical → index → serve → monitor. Як пріоритизувати технічні дефекти за впливом, масштабом і вартістю виправлення.

Стан даних і джерел: 14 вересня 2026 року

Чекліст корисний як memory aid, але небезпечний як модель. Він ставить поруч “відсутній favicon”, “canonical на 404” і “robots блокує product pages”, хоча business impact відрізняється на порядки. Технічне SEO стає сильним, коли кожен дефект прив’язаний до етапу pipeline, affected URL set і observable outcome.

Рис. 16.1. Технічне SEO - ланцюг залежностей, а не список незалежних прапорців.

16.1. Upstream дефекти породжують downstream шум

Якщо template генерує мільйон параметричних URL, downstream ви побачите overcrawl, canonical conflicts, index exclusions, noisy logs,

GSC coverage clutter і rank dilution. Лікувати кожний symptom окремо - марно. Я шукаю найранішу точку, де система почала створювати зайвий URL-space.

16.2. Severity = impact × scale × confidence, а не “Critical” із crawler

Автоматичний audit tool не знає business value URL. 404 на старому test path може бути нульовою проблемою; 302 замість 301 на domain migration - великою. Я оцінюю: скільки valuable URL affected; який механізм Search порушено; наскільки впевнені в causality; що коштує fix; чи є risk rollout.

Таблиця 16.1. Моя проста модель пріоритету технічного дефекту.

Вісь	**0-1**	**2-3**	**4-5**
Business impact	майже немає	помірний сегмент	ключовий revenue / visibility
Scale	одиничні URL	тисячі / один template	велика частина сайту
Evidence	гіпотеза	сильна кореляція / docs	прямий технічний доказ
Fix cost	дорога перебудова	середній development	простий конфіг / template fix

Я не складаю ці числа у “науковий score”. Вони примушують команду явно проговорити припущення. Часто вже після цього “Critical SEO issue” падає на 30-й рядок roadmap.

16.3. Тестуйте шаблон, не URL

У site with 2M pages майже кожна технічна проблема є template problem. Вибірка по 20-50 URL на template дає більше, ніж random crawl 500k URL без класифікації. Я завжди додаю page_type до crawl/GSC/log datasets. Тоді “canonical mismatch 7%” перетворюється на “product variant template 63%”.

16.4. Release QA важливіший за річний “технічний аудит”

Найкраща технічна SEO-система ловить regression у день релізу. Для критичних шаблонів я автоматизую smoke tests: HTTP, robots, canonical, hreflang, structured data, main content selector, links count, title, sitemap inclusion. Раз на квартал audit потрібен для системних речей, але не повинен бути єдиним контролем.

16.5. Monitoring має бути symptom-based

Таблиця 16.2. Мінімальний технічний monitoring layer.

Сигнал	Що може зламатися	Поріг реакції
5xx rate Googlebot	інфраструктура / deploy	аномалія від baseline + affected valuable templates
Indexed coverage	noindex/canonical/ content regression	різка зміна specific template
Crawl requests	robots/server/url explosion	відхилення по host/page type
Sitemap valid URLs	generation pipeline	невідповідність expected inventory
Canonical mismatch	template / migration	зростання частки conflict cluster
hreflang errors	international release	broken reciprocity / wrong locale mapping

16.6. Technical debt має ціну opportunity

Fix, який економить crawler resources, але не змінює discovery/index/serving valuable pages, може бути nice-to-have. Натомість повільне створення sitemap для 100k daily inventory може мати реальну opportunity cost. Я завжди формулюю issue через expected outcome: “зменшити 3 млн непотрібних fetch/day і скоротити discovery lag new products”, а не “прибрати parameters”.

16.7. Інструменти - це датчики, не судді

Screaming Frog, Sitebulb, Ahrefs audit, Botify, Oncrawl бачать різні representations і застосовують власні rules. Я не імпортую їх severity у roadmap. Дані crawler + GSC + logs + server + browser + business inventory мають бути зведені в одну модель. Найважливіше питання: чи дефект існує для Googlebot і чи стосується цінної URL.

16.8. Технічний audit deliverable

Мій deliverable - не PDF на 150 сторінок. Це таблиця системних issues: mechanism, affected template, affected URL count, evidence, business risk, reproduction steps, recommended fix, owner, effort, validation query, rollback risk. До неї можна додати appendix із raw findings. Команда розробки повинна мати змогу відтворити проблему без SEO-перекладача.

16.9. Що ми знаємо точно

Search pipeline складається з discovery/crawl/render/index/serve; Search Essentials визначає мінімальні технічні вимоги; Google має окремі documented controls для robots, status codes, canonical, hreflang, structured data, migrations. Не існує офіційного Google “technical health score”. Будь-який 92/100 - метрика інструмента.

Джерела до Глави 16

1. Google Search Central. Crawling and indexing overview. https://developers.google.com/search/docs/crawling-indexing

2. Google Search Central. Search Essentials. https://developers.google.com/search/docs/essentials

3. Google Search Central. Latest documentation updates. https://developers.google.com/search/updates

ЧАСТИНА III / ПОБУДОВА SEO-СИСТЕМИ

ГЛАВА 17

Міжнародне SEO: hreflang, країни, мови та конфлікти сигналів

Як проєктувати locale architecture, коли потрібні окремі URL, як canonical взаємодіє з hreflang, чому auto-redirect по IP ламає crawl і як QA-ити великі міжнародні кластери без ручної перевірки кожної сторінки.

Стан даних і джерел: 14 вересня 2026 року

International SEO ламається не через складність самого hreflang. Воно ламається через те, що бізнес-модель, URL architecture, canonicalization, translations, geolocation і content operations приймають різні рішення про те, що є окремою сторінкою. hreflang не виправляє цю суперечність - він лише описує зв’язок уже існуючих locale URLs.

Рис. 17.1. Локалізовані URL мають утворювати узгоджений hreflang-кластер із self-canonical сторінками.

17.1. Multilingual і multi-regional - різні задачі

Google розділяє multilingual site - контент кількома мовами - і multi-regional site - контент для різних країн/регіонів.[1] es-ES і es-MX можуть бути однією мовою, але різними commercial contexts, currency, regulation, offers. en-US і fr-CA - і різна мова, і locale. Архітектура повинна починатися з product/legal/business differences, а не з того, скільки subfolders хоче SEO.

17.2. Окремі URL для мовних версій - базова умова

Google рекомендує різні URL для різних мовних версій, а не одну URL, яка змінює контент через cookie або browser language.[1] Причина практична: crawler має стабільну адресу документа. Locale-adaptive content може бути знайдений не повністю, оскільки Googlebot часто crawls from US і без Accept-Language header.

17.3. ccTLD, subdomain чи subfolder - це trade-off, не ranking hack

Таблиця 17.1. Архітектурні варіанти GEO.

Модель	Плюси	Мінуси
example.es	сильний country cue, локальний бренд	окремий authority/ops, дорожче масштабувати
es.example.com	ізоляція stack/teams	окремий host, складніша аналітика й deployment
example.com/es/	спільний домен і простіше link consolidation	потрібна дисципліна locale routing
динамічна одна URL	менше URL	ризик неповного crawl, складна кеш/геолокація, слабка прозорість

Я вибираю модель за операційною реальністю: чи є окремі teams, products, regulations, link acquisition, infrastructure. SEO не повинно змушувати бізнес заводити 20 ccTLD, якщо компанія не може їх якісно підтримувати.

17.4. hreflang вимагає взаємності

Localized versions повинні посилатися одна на одну, а locale codes бути валідними. Для великого набору сторінок зручніше sitemap hreflang, але логіка та сама. x-default корисний для global selector або fallback, але не є “міткою англійської мови”.[2]

17.5. Canonical і hreflang повинні говорити про різні речі

Canonical визначає representative у duplicate cluster; hreflang визначає locale alternatives. Для es-ES і es-MX зі схожим іспанським контентом я зазвичай хочу self-canonical на кожній locale URL плюс взаємний hreflang. Canonical es-MX → es-ES одночасно з hreflang es-MX створює суперечливе повідомлення: “це окрема локаль” і “це не основна версія документа”.[1][3]

17.6. Не redirect-те crawler за IP або Accept-Language без вибору

Google прямо радить уникати automatic redirect між language versions, оскільки це може завадити користувачам і crawler побачити всі версії.[1] Краща модель: стабільна URL, банер/selector із пропозицією локалі й crawlable links між варіантами. Geo-personalization можна робити всередині локалі, але він не повинен приховувати самі документи.

17.7. Переклад boilerplate не створює окрему мовну сторінку

Google визначає language за visible content, а не lang attribute або URL. Якщо перекладені menu/footer, а основний body лишився англійським, система може не сприймати сторінку як повноцінну локалізацію.[1] Це особливо важливо для UGC, marketplace і programmatic pages, де template перекладений, а value content - ні.

17.8. International keyword research робиться окремо по locale

Переклад keyword list із ES у MX або US у UK пропускає local vocabulary, brands, regulation, payment methods і SERP composition. Я використовую translation тільки як seed. Далі окремі GSC properties/filters, local SERPs, native-language review і business data. Навіть той самий англійський query може мати інший commercial intent у різних країнах.

17.9. QA великого hreflang-графа

Таблиця 17.2. Що перевіряти автоматично.

Перевірка	Умова
HTTP	кожен hreflang target повертає expected 200
Indexability	target не noindex і не blocked
Canonical	target self-canonical або узгоджений із locale strategy
Reciprocity	A посилається на B і B на A
Locale code	валідна language/region комбінація
Content	мова основного контенту відповідає locale
Sitemap parity	expected locale inventory повний

17.10. Що ми знаємо точно

Google рекомендує окремі URL для мовних версій, підтримує hreflang у HTML/headers/sitemaps, радить не робити примусові locale redirects і визначає мову за visible content. Для similar same-language regional pages рекомендує поєднувати canonical і hreflang.[1][2][3] Hreflang не є ranking boost; його задача - допомогти показати правильний locale variant.

Джерела до Глави 17

1. Google Search Central. Managing multi-regional and multilingual sites. https://developers.google.com/search/docs/specialty/international/managing-multi-regional-sites

2. Google Search Central. Tell Google about localized versions. https://developers.google.com/search/docs/specialty/international/localized-versions

3. Google Search Central. Canonicalization. https://developers.google.com/search/docs/crawling-indexing/canonicalization

4. Google Search Central. International and multilingual topics. https://developers.google.com/search/docs/specialty/international

ЧАСТИНА IV / АВТОРИТЕТ І КОНКУРЕНЦІЯ

ГЛАВА 18

PageRank і граф посилань без міфів

Що дає оригінальна модель PageRank, чого вона не пояснює в сучасному Google, як думати про графову важливість без “link juice” і чому внутрішні й зовнішні посилання - одна система з різними межами контролю.

Стан даних і джерел: 14 вересня 2026 року

PageRank важливий не тому, що SEO у 2026 році може порахувати “реальний PR” сторінки. Не може. Він важливий як модель: web - directed graph, де links створюють структуру рекомендацій/переходів, а importance рекурсивно залежить від важливості сторінок, що посилаються. Оригінальна робота Page, Brin, Motwani і Winograd описувала саме таку graph-based оцінку.[1] Сучасні ranking systems значно складніші, але Google і сьогодні прямо називає links одним із сигналів relevance.[2]

Рис. 18.1. Базова інтуїція PageRank: важливість сторінки залежить не лише від кількості посилань, а й від графу, з якого вони приходять.

18.1. Кількість links не дорівнює PageRank

100 links із майже ізольованих сторінок не обов'язково сильніші за одне посилання зі сторінки, на яку веде багато важливих шляхів. Саме рекурсивність відрізняє PageRank від простого indegree. У практиці це пояснює, чому sitewide junk links і meaningful editorial references не варто рахувати однією колонкою.

18.2. Random surfer - інтуїція, а не поточний Google formula

Оригінальна модель часто пояснюється random surfer: користувач переходить links і з певною ймовірністю "телепортується" на випадкову сторінку. Це хороша математика для розуміння steady-state importance. Але некоректно стверджувати, що сучасний Google ranking буквально використовує ту саму формулу з тим самим damping factor. Patent/paper ≠ current production weights.

18.3. Internal і external graph - одна логіка, різний контроль

Внутрішній граф ми можемо перепроєктувати. Зовнішній - лише частково впливати через PR, product, outreach, partnerships, citations. Це робить internal linking найконтрольованішим способом змінити topology навколо money pages. Але внутрішній link не створює зовнішню authority з нуля; він перерозподіляє те, що вже є в site graph.

Таблиця 18.1. Що PageRank-модель пояснює добре, а що не треба з неї виводити.

Добре пояснює	Не доводить
Чому links формують graph importance	що DR/DA дорівнює PageRank Google
Чому важливість донора має значення	що будь-яке посилання з "сильного домену" корисне
Чому internal links можуть перерозподіляти вагу	що можна точно порахувати Google PR crawler-ом
Чому orphan pages структурно слабкі	що PageRank є єдиним або головним ranking signal

18.4. Third-party authority metrics - корисні моделі, якщо пам’ятати їх природу

DR, DA, Authority Score, Trust Flow будуються на власних crawls, формулах і link indexes. Це не Google metrics. Я використовую їх як comparative features: фільтр донорів, trend, competitor gap. Але рішення про link acquisition ніколи не приймаю за одним score. Найважливіші питання - чи реальний сайт, чи індексується сторінка, чи має вона реальну аудиторію, чи релевантний context, чи link editorially plausible.

18.5. Redirects і canonical змінюють граф, але не як магічний “100% juice transfer”

Google документує permanent redirects як сильний canonical signal. [3] Це підстава для нормальних міграцій і консолідації. Але точний “% PageRank transfer” Google не публікує. Тому заяви “301 передає 90/95/100% ваги” без джерела - псевдоточність. Практично оцінюємо outcome: canonical selection, indexed target, rankings, crawl і backlinks.

18.6. Link decay і граф у часі

Зовнішній граф не статичний. Pages зникають, редиректяться, стають noindex, змінюють content, links випадають із templates. Тому важливі не тільки acquired links, а retained live links. Для великих проєктів я дивлюся lost referring pages, status distribution, redirect chains і те, чи залишився link у rendered HTML.

18.7. Практичний висновок

PageRank потрібен Senior SEO не як релігія про “juice”, а як дисципліна мислення про граф. Сильні pages повинні мати шляхи до тих документів, які бізнес вважає важливими. External links мають оцінюватися в контексті source page і web graph. Усе інше - моделі з обмеженнями, а не доступ до внутрішнього score Google.

Джерела до Глави 18

1. Page, Brin, Motwani, Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Stanford technical report. http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

2. Google Search Central. Link best practices. https://developers.google.com/search/docs/crawling-indexing/links-crawlable

3. Google Search Central. Redirects and Google Search. https://developers.google.com/search/docs/crawling-indexing/301-redirects

4. Brin, Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. https://infolab.stanford.edu/~backrub/google.html

ЧАСТИНА IV / АВТОРИТЕТ І КОНКУРЕНЦІЯ

ГЛАВА 19

Зовнішні посилання: оцінка, придбання, анкори та ризик

Практична оцінка донорів без культу DR: source/page/context/rel, link acquisition як портфель, анкори як розподіл ризику, paid links, мережі, drops і межа між механікою та Google policy.

Стан даних і джерел: 14 вересня 2026 року

Зовнішні links - одна з тем, де SEO найчастіше змішує три різні речі: механіку web graph, third-party метрики і правила Google. Я розділяю їх. Links можуть бути важливим сигналом для discovery/relevance; DR/DA - моделі сторонніх сервісів; buying links for ranking purposes прямо входить у Google link spam policy.[1] Ці твердження можуть бути істинними одночасно.

Рис. 19.1. Оцінка посилання має щонайменше чотири шари: source, page, link і policy/risk.

19.1. Донор починається з реальності сайту

Перед metrics я перевіряю: чи є органічний footprint, нормальна історія, реальні categories, authors, contact/brand, inbound traffic signals,

indexed pages, consistency теми. “Сайт” із 300 paid guest posts, нульовою аудиторією і випадковими topics може мати високий DR завдяки старому link graph, але це не робить placement сильним.

19.2. Page-level context важливіший за domain average

Link існує на конкретній page. Я дивлюся indexability, traffic/query relevance, internal inlinks, outbound link density, placement, surrounding text, freshness. Link із тематичної статті, яка сама має visibility й internal support, відрізняється від link на orphan post у “blog” сильного domain.

Таблиця 19.1. Швидкий аудит донорської сторінки.

Шар	**Перевірка**
HTTP/index	200, canonical self/expected, indexable
Visibility	queries/traffic або хоча б нормальний search footprint
Context	тема сторінки й paragraph реально пов’язані з target
Placement	body editorial > випадковий footer/sidebar
Outlinks	чи сторінка не є link farm
Persistence	чи publication/archive стабільні
Rel	sponsored/nofollow/ugc або plain link

19.3. Anchor text - не точка оптимізації, а distribution

Я не ставлю задачу “купити 30% exact anchors”. Спершу дивлюся natural language навколо entity/brand. Для brand/review pages нормальні brand, URL, partial, generic, compound anchors. Exact commercial anchors можуть існувати, але якщо portfolio виглядає як шаблон SEO-закупівлі, це додатковий risk. Anchor distribution треба читати разом із source types і timeline.

19.4. Link velocity без baseline беззмістовна

100 нових referring domains за місяць можуть бути аномалією для маленького локального сайту і нормою для бренду після PR launch. Я нормалізую темп на historical baseline, content/PR events, competitor growth і domain age. Сам по собі “швидкий приріст” не є documented penalty trigger.

19.5. Paid links: механіка може працювати, policy risk залишається

Google прямо класифікує buying/selling links for ranking purposes як link spam і рекомендує qualifying paid placements через rel="sponsored" або nofollow.[1][2] Це не заважає ринку paid links існувати. Але senior-level decision повинна включати probability × impact risk, а не тільки expected ranking upside.

19.6. PBN і networks: головний risk - pattern і централізована крихкість

PBN дає контроль над anchors, placements і timing, але одночасно створює common ownership / infrastructure / content / linking patterns і single-point-of-failure. Я оцінюю мережу як portfolio risk: footprint, cost to replace, deindex history, outbound saturation, host diversity, ownership leakage. “Немає footprint” - не факт, а гіпотеза, яку неможливо довести ззовні.

19.7. Expired domains і redirects

Expired domain може мати корисний history/link graph, але Google окремо має policy expired domain abuse: repurpose primarily to manipulate rankings with low-value content.[3] Redirect expired domain на unrelated money site - не “безкоштовний PageRank”, а risk strategy з невизначеним outcome. Перш ніж купувати drop, перевіряю topical history, anchors, snapshots, status continuity, spam, trademarks і чи можна створити legitimate user value.

ПОПЕРЕДЖЕННЯ
У high-risk vertical тактика може бути економічно раціональною навіть із policy risk. Але книга не повинна підміняти business decision міфом “Google цього не бачить”. Ризик треба називати ризиком.

19.8. Link acquisition як portfolio

Я розділяю assets: editorial PR, niche edits, guest posts, partnerships, directories/citations, UGC/community, paid placements, owned network. Для кожного type - cost, expected lifetime, traffic/referral value, policy risk, anchor control. Це дає портфель, де один channel не може знищити весь link profile.

Джерела до Глави 19

1. Google Search Central. Spam policies - Link spam.
https://developers.google.com/search/docs/essentials/spam-policies

2. Google Search Central. Qualify outbound links.
https://developers.google.com/search/docs/crawling-indexing/qualify-outbound-links

3. Google Search Central. Spam policies - Expired domain abuse.
https://developers.google.com/search/docs/essentials/spam-policies

4. Google Search Central. Link tagging and link spam update.
https://developers.google.com/search/blog/2021/07/link-tagging-and-link-spam-update

ЧАСТИНА IV / АВТОРИТЕТ І КОНКУРЕНЦІЯ

ГЛАВА 20

Бренд і сутності: як бренд стає пошуковим активом

Без фрази “Google любить бренди”: brand demand, entity consistency, first-party sources, Organization markup, reviews, publishers, co-occurrence і те, що реально можна виміряти.

Стан даних і джерел: 14 вересня 2026 року

Бренд у Search - не секретний ranking factor з одним score. Це сукупність спостережуваних речей: navigational demand, consistency entity facts, mentions, links, reviews, first-party pages, knowledge representations, publisher coverage. Сильний бренд зменшує невизначеність для користувача й системи, але ми не бачимо внутрішній “brand authority”.

Рис. 20.1. Бренд стає пошуковим активом через послідовну ідентичність, попит, незалежні згадки й представлення в Search/AI.

20.1. Brand queries - найпряміший first-party signal попиту

У GSC я сегментую exact brand, brand+modifier і non-brand. Динаміка brand impressions показує не “authority”, а search demand і visibility власної сутності. Brand+review, brand+login, brand+bonus, brand+support показують окремі user tasks навколо entity. Для нових брендів це сильний health metric, бо його важко підробити внутрішньою перелінковкою.

20.2. Entity consistency починається з first-party source of truth

Name, logo, official URL, contacts, legal company, product names, founders/owners where relevant, social profiles - мають бути узгоджені. Organization structured data може допомогти Google краще understand і disambiguate організацію, а деякі fields впливають на visual elements/knowledge panels.[1] Але markup не повинен описувати те, чого немає у видимому або реальному бізнесі.

20.3. Structured data - explicit clue, не кнопка “створити entity”

Organization/LocalBusiness schema корисна як machine-readable layer, але Google general guidelines вимагають, щоб structured data відповідала visible content і не була misleading.[2] Я додаю лише факти, які можу підтримувати. SameAs не повинен бути списком випадкових профілів, а logo - різним на кожній сторінці.

20.4. Independent sources сильніші за 100 власних сторінок

Згадка бренду на своєму сайті нічого не доводить про зовнішнє існування. Reviews, publishers, business directories, app stores, social profiles, partners, regulators, datasets - створюють external references. Я

не стверджую, що кожна mention без link є ranking signal. Я використовую їх як evidence entity footprint і джерело branded discovery.

Таблиця 20.1. Brand SEO без магії.

Шар	Метрика / перевірка
Demand	brand impressions, branded queries, direct demand trends
Identity	consistent name/logo/URL/contact/legal relation
SERP control	частка first-party/controlled results у brand SERP
Reputation	review volume, rating distribution, recurring complaints
Independent coverage	publisher/partner/regulator references
AI visibility	brand mentions, cited first-party URLs, factual consistency

20.5. Brand SERP - реальна операційна поверхня

Для navigational query я інвентаризую top results, sitelinks, knowledge panel, social, reviews, news, autocomplete. Мета не “зайняти всі 10 позицій”, а щоб користувач швидко отримав правильні official destinations і не бачив суперечливих facts. Для regulated affiliate brands окремо важливі support, terms, license/regulatory information.

20.6. AI робить factual consistency важливішою

Generative systems синтезують кілька sources. Якщо official page каже одне, app store друге, partner третє, шанс помилкового synthesis зростає. Я створюю canonical fact sheet на first-party site і підтримую однакові facts на high-authority profiles. Це не “LLM hack”, а data hygiene.

20.7. Co-occurrence і mentions: обережно з причинністю

SEO часто називає co-occurrence підтвердженим ranking factor. Публічних даних недостатньо для такої категоричності. Я розглядаю mention поруч із category/entity як спосіб external context і brand discovery, але не приписую йому невідому вагу. Якщо кампанія digital PR збільшує mentions, branded searches і referral traffic - це вже вимірюваний outcome.

20.8. Що ми знаємо точно

Google підтримує Organization/LocalBusiness structured data, використовує її для explicit organization details і може відображати ці facts у knowledge panels/visual elements; structured data має відповідати реальному content.[1][2][3] Не знаємо: універсального “brand score” або формули, що перетворює mentions у ranking.

Джерела до Глави 20

1. Google Search Central. Organization structured data. https://developers.google.com/search/docs/appearance/structured-data/organization

2. Google Search Central. General structured data guidelines. https://developers.google.com/search/docs/appearance/structured-data/sd-policies

3. Google Search Central. Establish your business details. https://developers.google.com/search/docs/appearance/establish-business-details

4. Google Search Central. LocalBusiness structured data. https://developers.google.com/search/docs/appearance/structured-data/local-business

ЧАСТИНА IV / АВТОРИТЕТ І КОНКУРЕНЦІЯ

ГЛАВА 21

Affiliate SEO: комерційна видача, money pages і ризик thin affiliation

Як робити affiliate-сторінки, які реально допомагають вибору: reviews, comparisons, bonus/promo, EMD, локалізація, програми партнерів і де Google прямо проводить межу thin affiliation.

Стан даних і джерел: 14 вересня 2026 року

Affiliate SEO не є маргінальною версією SEO. Це бізнес-модель, де органічна сторінка повинна одночасно відповісти на query, зменшити невизначеність перед вибором і перевести користувача до merchant. Проблема починається, коли affiliate layer не додає нічого між SERP і merchant. Google прямо називає thin affiliation практикою, де descriptions/reviews копіюються з merchant або network і не дають original value.[1]

Рис. 21.1. Сильна affiliate page додає власний шар оцінки й доказів між запитом та merchant.

21.1. Money page повинна мати власну задачу

Review, bonus, app, promo code, withdrawal, payment methods, alternatives - це не просто keyword-modifiers. Кожна page type має окрему decision task. Review допомагає оцінити продукт; bonus - зрозуміти value і terms; app - install/compatibility; withdrawal - time/methods/limits. Якщо всі сторінки мають 80% однакового тексту, структура створена під query variations, а не під користувача.

21.2. Review без evidence слабкий навіть якщо дуже довгий

Google high-quality reviews guidance радить демонструвати knowledge, власний evidence, quantitative measurements, плюси/мінуси й порівняння.[2] Для affiliate це практична перевага: screenshots account, actual onboarding, payment test, terms date, support response, product changes. Не треба вигадувати “ми протестували”, якщо не тестували. У такому випадку чесно описуйте desk research і джерела.

21.3. Comparison intent потребує критеріїв

Таблиця 21.1. Сильне порівняння - це decision model, а не список брендів.

Шар	Що потрібно
Критерії	що реально впливає на вибір у цій vertical
Метод	як зібрано/перевірено facts
Нормалізація	однакові units і time cut-off
Trade-offs	кому варіант підходить / не підходить
Evidence	terms, screenshots, source links, measurements
Disclosure	як affiliate model впливає на монетизацію

21.4. Bonus/promo pages швидко старіють

Комерційні terms - time-sensitive data. Якщо welcome bonus змінився, стара сторінка може створювати factual error і compliance risk. Я зберігаю fields структуровано: amount, match %, wagering, min deposit, max bet, expiry, code, eligible countries, last verified, source. Це дозволяє QA і масові updates без переписування всього тексту.

21.5. EMD не замінює бренд і content moat

Exact-match domain може бути зрозумілим для query і CTR, але Search не зобов'язаний ранжувати його через слова в domain. У портфельних стратегіях EMD також підвищує ризик doorway-like footprint, якщо десятки domains ведуть до одного merchant/offer через дуже схожі pages. Google doorway policy прямо згадує multiple sites/URLs, створені для охоплення схожих queries і funnel users до одного destination.[3]

21.6. Thin affiliation - policy, яку треба читати буквально

Google дає й позитивні приклади affiliate value: additional price information, original reviews, rigorous testing/ratings, navigation of products/categories, product comparisons.[1] Це корисніше за спробу вгадати "мінімальний % унікальності". Унікальний текст без унікальної value може лишатися thin.

21.7. Affiliate links і rel

Paid/affiliate relationships потрібно правильно qualifying. Google outbound link guidance рекомендує rel="sponsored" для advertisements/paid placements.[4] Це не заважає tracking parameters чи redirectors, але треба контролювати status, destination, geo behavior і те, щоб crawler/user не потрапляли в різні deceptive flows.

21.8. Multi-GEO affiliate - localization, а не переклад

Іспанія, Мексика й Аргентина можуть мати спільну мову, але різні operators, payments, regulation, currency і search behavior. Я не тиражую одну review page 1:1 через AI translation. Спільним може бути factual core продукту; локальними - offer, legal context, payment methods, terms, CTA, support availability і competitor set.

21.9. Що вимірювати

Окрім rankings/clicks: outbound CTR to merchant, click-to-registration, registration-to-FTD/lead approval, revenue/EPC by landing page, brand/non-brand query mix, freshness error rate, pages per merchant, time-to-update offer. SEO page, яка дає багато clicks і поганий downstream conversion, може бути неправильною answer, навіть якщо rank хороший.

Джерела до Глави 21

1. Google Search Central. Spam policies - Thin affiliation. https://developers.google.com/search/docs/essentials/spam-policies

2. Google Search Central. Write high quality reviews. https://developers.google.com/search/docs/specialty/ecommerce/write-high-quality-reviews

3. Google Search Central. Spam policies - Doorway abuse. https://developers.google.com/search/docs/essentials/spam-policies

4. Google Search Central. Qualify outbound links. https://developers.google.com/search/docs/crawling-indexing/qualify-outbound-links

5. Google Search Central. Reviews system. https://developers.google.com/search/docs/appearance/reviews-system

ЧАСТИНА V / AI-ПОШУК

ГЛАВА 22

RAG, розгортання запиту та опора на джерела: механіка генеративного пошуку

Технічна модель AI Search без магії: query interpretation, fan-out, retrieval, candidate passages, grounding, synthesis, citations. Що Google підтверджує, а що лишається закритою реалізацією.

Стан даних і джерел: 14 вересня 2026 року

Generative Search краще розуміти не як “LLM поверх Google”, а як pipeline, де класичні search capabilities з’єднані з генеративним synthesis. Google у 2026 році публічно описує retrieval-augmented generation (RAG/grounding) і query fan-out: модель запускає пов’язані пошуки, отримує релевантні web pages із Search index і використовує їх для формування відповіді з links.[1]

Рис. 22.1. Спрощена модель generative Search: fan-out → retrieval → grounding → synthesis.

22.1. Query fan-out змінює одиницю релевантності

Один user prompt може породити кілька retrieval queries. Наприклад, “best phone for travel under $600” може розкладатися на camera, battery, roaming/eSIM, durability, current prices. Це означає, що одна сторінка може бути relevant не до surface prompt, а до конкретного subtask. SEO не бачить повний fan-out у Google, тому не треба вигадувати точні hidden queries. Bing уже показує sample grounding queries - це ближче до спостережуваних даних.[2]

22.2. Retrieval ≠ final answer

Документ може потрапити у candidate set і не бути використаний. Інший може бути використаний для factual grounding, але не отримати видиму citation. Третій - отримати link як додаткове джерело. Саме тому “нас немає в AI answer” недостатньо: треба визначити, яку подію вимірює інструмент.

22.3. Passage-level usefulness важлива, але “chunk size 50 слів” не підтверджений rule

Generative systems можуть витягувати specific information із pages/passages, але Google не вимагає спеціально різати контент на маленькі chunks. 2026 AI guide прямо mythbust-ить ідею спеціального chunking як requirement.[1] Практичний висновок простіший: headings, tables, clear statements і локальна coherence допомагають і людям, і extraction, але не треба ламати нормальний текст заради псевдоформули.

22.4. Grounding цінує актуальність і перевірюваність

RAG потрібен саме для того, щоб модель спиралася на external current sources. Для SEO це підвищує цінність pages із clear facts, dates, sources, structured product/local data. Але “додати citations” не є ranking hack. Важливіше, щоб page містила correct, specific, updateable information.

Таблиця 22.1. Що ми можемо оптимізувати на різних етапах.

Етап	Під нашим контролем	Не під нашим контролем
Index eligibility	crawl, noindex, canonical, snippet controls	коли/чи Google index URL
Retrieval readiness	task fit, facts, structure, fresh data	точний hidden fan-out і candidate set
Grounding usefulness	конкретні claims, evidence, clear sections	які passages система вибере
Citation	stable URL, accessible source, clear attribution	чи буде видимий link у конкретній відповіді
Click	title/source trust, value beyond answer	позиція citation і UI конкретного experiment

22.5. Search index лишається фундаментом Google AI features

Google прямо пише: щоб page була eligible supporting link у AI Overviews/AI Mode, вона має бути indexed і eligible for snippet у normal Search.[3] Це важлива межа проти “LLM SEO без SEO”: якщо crawler не може отримати page або вона noindex, спеціальна AI schema її не врятує.

22.6. Controls для AI Search у Google - це ті самі Search controls

Для AI features у Search Google використовує Googlebot controls: noindex, nosnippet, data-nosnippet, max-snippet для того, що можна показувати.[3] Google-Extended стосується інших training/grounding contexts і не є control для AI Overviews/AI Mode Search. Це треба чітко розділяти.

22.7. Що ми не знаємо

Не знаємо публічно: точний fan-out, кількість retrieval rounds, thresholds, passage scoring, prompt rewriting, weights organic ranking vs other systems. Тому будь-який сервіс, що продає “точний citation score Google AI”, фактично моделює поверхню, а не читає internal system.

Джерела до Глави 22

1. Google Search Central. Optimizing for generative AI features. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

2. Bing Webmaster Blog. AI Performance in Bing Webmaster Tools. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview

3. Google Search Central. AI features and your website. https://developers.google.com/search/docs/appearance/ai-features

ЧАСТИНА V / AI-ПОШУК

ГЛАВА 23

Цитування: як сторінка стає джерелом AI-відповіді

Чому citation не дорівнює ranking, які властивості сторінки реально можна покращити, як працювати з source inclusion без “AI schema” і як відрізняти evidence від нової індустрії GEO-хаків.

Стан даних і джерел: 14 вересня 2026 року

Цитування - нова видима подія, але не нова магія. Для Google supporting link page повинна пройти звичайний index/eligibility layer, а AI systems потім знаходять релевантні sources через Search. У Bing citation вже стала окремою measurable metric: Total Citations, cited pages і grounding queries.[1] Це дозволяє аналізувати source inclusion окремо від organic rank.

Рис. 23.1. До видимого citation сторінка проходить кілька окремих фільтрів.

23.1. Ranking ≠ citation

Top-3 page має конкурентну перевагу relevance/authority, але AI answer може потребувати вузький fact із page, яка органічно нижче.

Навпаки, top page може бути listicle без конкретного fact для grounding. Тому citation optimization - не “підняти позицію” і не “додати FAQ schema”, а зробити source technically eligible і information-dense для task.

23.2. Clear claims краще за розмиті абзаци

Я не дроблю текст механічно, але прагну, щоб важливе твердження мало subject, value, unit, date/context і source. “Вивід швидкий” слабше, ніж “у нашій вибірці 50 withdrawals median time становив X годин, period Y”. Друге простіше перевірити, процитувати й оновити.

23.3. Source identity має бути стабільною

Stable canonical URL, author/byline where expected, publication/update dates, organization identity і links to primary sources допомагають читачу оцінити provenance. Google helpful content guidance окремо ставить питання Who/How і радить accurate authorship.[2] Це не “citation factor”, але правильна publishing hygiene.

23.4. Немає спеціальної AI schema для Google Search

2026 AI guide прямо каже, що special schema.org markup для generative AI Search не потрібна.[3] Використовуйте structured data, яка відповідає реальному content і supported Search features. Створювати вигаданий JSON-LD type “AIAnswer” немає сенсу.

Таблиця 23.1. Практичний source-readiness audit.

Шар	Питання
Technical	URL indexable? canonical stable? snippet allowed?
Task fit	чи є конкретна відповідь на subtask, а не тільки загальний текст?
Evidence	чи є primary sources/data/methodology?
Freshness	чи time-sensitive facts мають дату перевірки?

Шар	Питання
Identity	хто автор/організація і чи зрозуміла відповідальність?
Extractability	таблиці/списки/sections читаються без JS interaction?

23.5. Citation volatility - нормальна властивість generative answer

Відповідь може змінитися через model version, query wording, date, locale, conversation context. Тому один screenshot не доказ “ми цитуємось”. Я вимірюю repeated prompt set, кілька runs/period, platform, locale і фіксую source URLs. Metric - citation frequency у визначеній тестовій системі, а не абсолютна видимість усіх користувачів.

23.6. Не плутайте citation із traffic

Bing Total Citations не каже placement або click; Google generative impressions не дають dedicated click metric; third-party trackers бачать свої prompts. Бізнесу потрібні окремі layers: citation → referral traffic → assisted conversion → branded demand. Інакше “+40% citations” може не мати жодного economic outcome.

23.7. Що можна тестувати

Тести source inclusion мають бути змістовними: додали original dataset, чіткі comparison tables, primary sources, оновили stale facts, створили tool. Не тест “поставили 17 FAQ і 4 schema”. Вимірюємо citation frequency на fixed prompt set, organic performance і user metrics, бажано з control pages.

Джерела до Глави 23

1. Bing Webmaster Blog. AI Performance. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview

2. Google Search Central. Creating helpful, reliable, people-first content. https://developers.google.com/search/docs/fundamentals/creating-helpful-content

3. Google Search Central. Optimizing for generative AI features. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

4. Google Search Central. AI features and your website. https://developers.google.com/search/docs/appearance/ai-features

ЧАСТИНА V / AI-ПОШУК

ГЛАВА 24

GEO, AEO та LLMO: що реально нового, а що просто перейменували

Розкладаємо модні терміни на конкретні задачі: Search fundamentals, retrieval, citations, mentions, AI measurement і platform controls. Без “секретних GEO-факторів”.

Стан даних і джерел: 14 вересня 2026 року

Я використовую GEO/AEO/LLMO як ярлики, коли треба коротко назвати scope робіт. Але не приймаю тезу, що SEO “закінчилося” і його замінила нова дисципліна. Google 2026 формулює прямо: optimizing for generative AI Search is optimizing for the search experience and still SEO; foundational best practices лишаються релевантними.[1]

Рис. 24.1. GEO/AEO/LLMO додають нові поверхні й метрики поверх фундаментального SEO-пайплайна.

24.1. Що реально нове

Таблиця 24.1. Реальні нові задачі generative visibility.

Задача	Чим відрізняється від класичного SEO
Prompt-set monitoring	немає одного canonical keyword; відповіді стохастичні
Citation tracking	source inclusion окремо від organic position
Brand mention tracking	entity може бути згадана без URL
Grounding query analysis	у Bing доступний sample retrieval phrases
Platform crawler controls	OAI-SearchBot, GPTBot, Googlebot/Google-Extended мають різні ролі
AI referral attribution	нові referrers і delayed/assisted journeys

24.2. Що просто отримало нову назву

Technical accessibility, indexability, useful content, clear entities, original evidence, link graph, brand, freshness, internal architecture не стали іншими механізмами через слово GEO. Якщо consultant продає “GEO” і не перевіряє canonical/index/robots, він оптимізує поверхню без фундаменту.

24.3. Google прямо mythbust-ить частину GEO-ринку

У AI optimization guide Google розбирає популярні міфи: llms.txt не потрібен для ranking/visibility у Google Search; немає special AI schema; не потрібно штучно chunk-ити content; не треба створювати сторінку під кожен fan-out query.[1] Це дуже корисна межа між measurable work і ritual.

24.4. Інші платформи мають інші controls

ChatGPT Search використовує OAI-SearchBot для search discovery; OpenAI окремо описує GPTBot для training та інші agents.[2] Bing має IndexNow і AI Performance. Google Search використовує Googlebot controls для AI Overviews/Mode. Отже, platform-specific ops реальні, але це infrastructure, а не “новий content algorithm”.

24.5. Термінологія без вимірювання нічого не варта

Якщо агентство обіцяє “GEO visibility +300%”, перше питання: який denominator? Які prompts? Яка platform/model/date/locale? Mention чи citation? Які repetitions? Без protocol цифра не відтворювана. Власне measurement methodology важливіша за назву дисципліни.

24.6. Моя робоча модель

Я залишаю SEO як umbrella. Усередині: Search SEO, AI Search visibility, entity/brand visibility, agent accessibility. GEO/AEO/LLMO можна використовувати для комунікації, але в roadmap task називається конкретно: “підняти citation frequency у Bing Copilot для 120 fixed prompts” або “дозволити OAI-SearchBot і перевірити referral traffic”.

Джерела до Глави 24

1. Google Search Central. Optimizing for generative AI features. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

2. OpenAI Help Center. Publishers and developers FAQ. https://help.openai.com/uk-ua/articles/12627856-publishers-and-developers-faq

3. Bing Webmaster Blog. AI Performance. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview

ЧАСТИНА V / AI-ПОШУК

ГЛАВА 25

Контент, який AI не робить взаємозамінним

Як будувати інформаційну перевагу в середовищі, де базовий текст дешевий: proprietary data, experiments, tools, first-hand evidence, workflows, primary-source research і product utility.

Стан даних і джерел: 14 вересня 2026 року

Якщо модель може створити ваш матеріал, не маючи доступу до вашого сайту, даних або досвіду, ваша перевага невелика. Це не означає, що весь informational content помер. Це означає, що “правильний текст” стає baseline, а differentiator - data, evidence, utility і trust.

Рис. 25.1. Найстійкіша контентна перевага виникає з ресурсів, яких немає в загальнодоступному корпусі.

25.1. Proprietary data

Навіть невеликий dataset може бути сильним, якщо методологія зрозуміла. 2 млн GSC queries, 500 support tickets, 100 withdrawal tests,

price history, crawl logs - це матеріал, який competitor не може чесно відтворити без власного збору. Не треба завжди “велике дослідження”: іноді достатньо 50 observations із чіткими limitations.

25.2. Original experiments

Experiment - не “ми змінили title і traffic виріс”. Потрібні hypothesis, sample, control/comparison, timeframe, confounders, result, limitation. Навіть якщо test imperfect, transparent methodology створює більшу value, ніж категорична порада без evidence.

25.3. Tools і calculators

Інструмент часто закриває task краще за статтю. Mortgage calculator, hreflang validator, odds converter, bonus wagering calculator, SERP diff, log parser. Такі pages можуть отримувати links, repeat visits і citations, бо мають functional value. LLM може пояснити формулу, але користувач усе одно цінує готовий tool.

25.4. First-hand evidence

Photos, screenshots, screen recordings, measurements, support transcripts, code repos, before/after, invoices/terms snapshots - роблять claims перевірюваними. Google review guidance прямо радить evidence of own experience і quantitative measurements.[1]

Таблиця 25.1. Шкала “комодитизації” контенту.

Рівень	Приклад	Моя дія
Високий	визначення, загальні поради, переказ documentation	коротко, консолідувати або не створювати
Середній	порівняння з перевіреними актуальними facts	інвестувати в QA/freshness
Низький	власні дані, tool, experiment, firsthand evidence	робити flagship asset

Рівень	Приклад	Моя дія
Унікальний	продуктова функція або dataset, який існує лише у вас	будувати distribution і links навколо активу

25.5. Primary-source research

У regulated topics цінність часто не у “досвіді автора”, а в точному читанні законів, official docs, terms, APIs, regulator databases. Це теж non-commodity work: знайти current source, правильно інтерпретувати scope, зафіксувати cut-off і не вигадати те, чого документ не каже.

25.6. Workflow content

Senior audience цінує не “що таке canonical”, а “як за 30 хвилин відрізнити canonical conflict від index quality issue”. SOP, query, regex, spreadsheet model, decision tree - це executable knowledge. У цій книзі я навмисно додаю такі артефакти, бо їх відкривають під час роботи.

25.7. AI як production layer, не source of truth

AI добре прискорює clustering, extraction, draft, SQL, code, QA suggestions. Але proprietary value має приходити з реальних inputs. Якщо input - ті самі 10 top results, output лишається derivative незалежно від стилю.

Джерела до Глави 25

1. Google Search Central. Write high quality reviews. https://developers.google.com/search/docs/specialty/ecommerce/write-high-quality-reviews

2. Google Search Central. Creating helpful, reliable, people-first content. https://developers.google.com/search/docs/fundamentals/creating-helpful-content

3. Google Search Central. Optimizing for generative AI features. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

ЧАСТИНА V / AI-ПОШУК

ГЛАВА 26

Вимірювання AI-видимості: цитування, згадки, cited URLs і трафік

Практичний measurement stack: Google generative impressions, Bing citations/grounding queries, ChatGPT referrals, third-party prompt trackers, fixed prompt sets і головна проблема - різні знаменники.

Стан даних і джерел: 14 вересня 2026 року

AI visibility легко перетворити на новий DR: одна красива цифра, незрозумілий denominator і leaderboard конкурентів. Я проти цього. У 2026 вже є first-party telemetry: Google Search Console має окремі generative AI performance reports для AI Overviews/AI Mode, а Bing Webmaster Tools показує citations, cited pages і grounding queries.[1][2] Це не повна картина, але значно краще за pure scraping.

Рис. 26.1. Mention, citation, click і conversion - різні події з різними знаменниками.

26.1. Google: generative impressions

Google Search Console generative report показує impressions у generative features і dimensions pages/countries/dates/devices. Він не дає

query dimension, dedicated clicks/CTR/position для generative view.[1] Тому з нього можна чесно сказати “URL отримує exposure”, але не “цей prompt дав 14 clicks”.

26.2. Bing: citations і grounding queries

Bing AI Performance показує Total Citations, Average Cited Pages, page-level citation activity і sample grounding queries.[2] Це інша telemetry. Citation count не означає ranking або placement. Grounding queries - sample, не повний log. Але для content gap це унікальний first-party signal.

26.3. ChatGPT: referrals вимірюються краще за no-click mentions

OpenAI описує OAI-SearchBot і можливість public websites з’являтися в ChatGPT Search.[3] На своїй стороні можна виміряти referrals sessions/conversions із known referrers. Але no-click brand mention усередині відповідей платформа не дає власнику сайту як повний analytics feed, тому third-party prompt monitoring лишається вибіркою.

26.4. Fixed prompt set - лабораторія, не census

Я будую набір prompts за tasks: discovery, comparison, brand, category, support. Фіксую exact wording, language, locale, model/platform, date, fresh/new session, repetition count. Metric називаю “citation frequency у нашому test set”, а не “частка ринку AI”.

Таблиця 26.1. Мінімальний AI measurement dashboard.

Рівень	Метрика	Джерело
Google exposure	generative impressions, cited pages if available	GSC
Bing/Copilot	citations, cited URLs, grounding queries	Bing Webmaster Tools

Рівень	Метрика	Джерело
Prompt lab	mention/citation frequency by fixed prompts	власний tracker
Traffic	sessions by AI referrer, landing pages	GA4/logs
Business	conversion/revenue assisted by AI traffic	analytics/CRM
Brand	branded demand and direct traffic trend	GSC/GA4/Trends

26.5. Share of citations

Можна рахувати частку citations бренду серед competitor citations у фіксованому prompt set. Формула корисна для trend, але залежить від set composition. Якщо завтра додати 100 prompts, де конкурент сильний, metric зміниться без реальної зміни platform. Тому version prompt set - обов'язкове поле.

26.6. Volatility

Для кожного prompt я зберігаю не тільки current source, a history. Citation volatility = частка runs/періодів, де source set змінюється. High volatility означає, що один screenshot взагалі не репрезентативний; low volatility дає сильнішу основу для experiment.

26.7. Source overlap із organic SERP

Корисний research metric - overlap cited domains/URLs із top organic results. Але correlation не означає, що organic rank спричинив citation. Обидва можуть бути наслідком shared relevance/quality/authority systems. Google сам говорить, що AI features rooted in core ranking and quality systems.[4]

26.8. Attribution no-click

Якщо user бачить бренд у AI, потім через день робить branded search і конвертується, стандартний last-click не віддасть credit AI. Можна дивитися branded lift, geo/period experiments, post-view surveys, incrementality. Але без user-level exposure data причинність слабка. Я не записую весь ріст direct traffic в “AI-assisted”.

26.9. Що ми точно можемо сказати у 2026

Google дає direct generative impressions; Bing дає direct citations і sample grounding queries; website може виміряти referral traffic; third-party trackers дають controlled samples. Повного cross-platform census prompts/answers/clicks не існує для site owner. Хороший dashboard показує цю межу, а не приховує її одним score.

Джерела до Глави 26

1. Google Search Central. Search Generative AI performance reports. https://developers.google.com/search/blog/2026/06/gen-ai-performance-reports

2. Bing Webmaster Blog. AI Performance in Bing Webmaster Tools. https://blogs.bing.com/webmaster/February-2026/Introducing-AI-Performance-in-Bing-Webmaster-Tools-Public-Preview

3. OpenAI Help Center. Publishers and developers FAQ. https://help.openai.com/uk-ua/articles/12627856-publishers-and-developers-faq

4. Google Search Central. AI features and your website. https://developers.google.com/search/docs/appearance/ai-features

ЧАСТИНА VI / МАСШТАБ

ГЛАВА 27

Масштабне генерування сторінок: коли programmatic SEO має сенс

Programmatic SEO починається не з генератора тексту, а з правила існування URL: яка окрема потреба, який окремий набір даних і яка бізнес-дія виправдовують появу ще однієї сторінки.

Стан даних і джерел: 14 вересня 2026 року

Programmatic SEO часто помилково зводять до автоматичного створення тисяч сторінок за шаблоном. Це описує виробництво, але не стратегію. Для мене programmatic SEO — це система, у якій набір даних, правила комбінації сутностей, шаблон сторінки, внутрішні посилання, індексація та вимірювання працюють як один продукт. Якщо URL не має окремої причини існувати, автоматизація лише швидше масштабує зайві сторінки.

Критичне питання перед генерацією не “скільки сторінок ми можемо зробити?”, а “чому пошуковій системі й користувачу потрібна саме ця комбінація даних?”. Google прямо відносить до scaled content abuse масове створення неоригінальних сторінок, основною метою яких є маніпуляція ранжуванням, незалежно від того, створені вони генеративним AI, скриптом чи людьми.[1] Отже, автоматизація сама по собі не є проблемою. Проблема — відсутність окремої цінності URL.

Рис. 27.1. Programmatic SEO — це конвеєр даних, правил, шаблонів і контролю, а не просто масове написання текстів.

27.1. Правило існування URL

Я використовую простий gate: URL створюється лише тоді, коли він має окрему комбінацію наміру, даних і корисної дії. “Casino X bonus”, “Casino X withdrawal time” і “Casino X app” можуть бути різними сторінками, якщо SERP, очікування користувача і фактичні дані різні. Але створити по сторінці для кожної перестановки “best casino + місто + payment method + device” без окремої цінності — це вже комбінаторний шум.

Практично gate можна формалізувати чотирма питаннями: чи існує окремий попит; чи відрізняється набір фактів; чи можна зробити сторінку кориснішою за загальну категорію; чи має сторінка окремий шлях до конверсії. Якщо дві або більше відповідей “ні”, я не генерую URL автоматично.

Таблиця 27.1. Рішення про створення програмної сторінки.

Сигнал	Створювати	Не створювати
Попит	Є стабільні запити або зрозуміла задача користувача.	Комбінація існує лише тому, що її можна згенерувати.

Сигнал	Створювати	Не створювати
Дані	Є окремі first-party або структуровані факти.	Текст відрізняється тільки назвою міста/бренду.
SERP	Результати показують окремий тип сторінки.	За всі варіації ранжується одна загальна категорія.
Конверсія	Є окрема дія, фільтр, калькулятор, порівняння.	Сторінка лише перекидає користувача на іншу.
Підтримка	Дані можна оновлювати автоматично і контролювати.	Дані швидко старіють, джерело ненадійне.

27.2. Комбінаторний вибух

Найнебезпечніша властивість programmatic SEO — геометричне зростання можливих URL. 200 брендів × 40 способів оплати × 20 країн × 3 пристрої — це вже 480 000 комбінацій. Більшість із них може не мати окремого попиту, даних або сенсу. Саме тут потрібен eligibility layer до генератора сторінок.

Я спочатку рахую повний теоретичний простір комбінацій, а потім поетапно відсікаю: комбінації без даних, без попиту, без достатньої відмінності, із юридичними обмеженнями, дублікати наміру, комбінації з нульовим inventory. Те, що залишилося, — не “всі можливі URL”, а індексований продукт.

27.3. Дані важливіші за текст

Шаблон може бути один і той самий для 50 000 сторінок, якщо фактичне наповнення різне й корисне. І навпаки, 50 000 “унікальних” AI-текстів можуть бути семантично взаємозамінними. У сильних програмних проєктах основний актив — dataset: ціни, наявність, правила, коефіцієнти, характеристики, географія, реальні відгуки, вимірювання або інші структуровані факти.

Текстовий шар повинен пояснювати дані, а не маскувати їхню відсутність. Якщо джерело даних має дату, версію або SLA оновлення, це варто зберігати в моделі даних. Тоді можна автоматично

знаходити сторінки зі stale facts, а не чекати, поки помилку побачить користувач.

27.4. Шаблон як продукт

Шаблон — це не “H1 + 800 слів”. Це набір компонентів із правилами появи: summary, ключові факти, таблиця, порівняння, FAQ, калькулятор, карта, availability, CTA, докази, джерела. Для кожного компонента потрібні умови: показувати лише коли дані достатні; не підставляти порожні заглушки; мати fallback, якщо зовнішнє API недоступне.

На великих сайтах я тестую шаблони на крайніх випадках: найкоротший dataset, найдовші назви, нульове inventory, одна одиниця inventory, відсутня ціна, змішана валюта, невідомий бренд, кілька мов. Саме крайні випадки створюють тисячі thin або broken pages після масштабування.

27.5. Дублікати і канонікалізація

Programmatic SEO генерує дублікати не тільки на рівні тексту. Два URL можуть показувати той самий набір об’єктів через інший порядок фільтрів, різні параметри або альтернативні назви сутності. Перед запуском потрібен детермінований ключ сторінки: набір полів, який однозначно визначає інформаційну потребу.

Canonical не повинен бути методом приховування поганої URL-моделі. Якщо система може не створювати дубль, краще не створювати його. Canonical — сигнал для консолідації, але Google все одно обирає канонічну версію сам, використовуючи кілька сигналів. [2]

27.6. Внутрішні посилання і виявлення

Програмні сторінки мають бути частиною графа сайту. Якщо десятки тисяч URL існують лише в sitemap, це ознака слабкої інформаційної архітектури. Google прямо пояснює для ecommerce, що

зв’язки між сторінками й кількість переходів за посиланнями допомагають зрозуміти структуру і відносну важливість сторінок.[3]

Я будую hub pages за сутностями і атрибутами, а не випадкові “related pages”. Кожен URL повинен мати зрозуміле батьківське місце: країна → категорія → підкатегорія → об’єкт; бренд → продукт → варіант; ліга → команда → матч. Це одночасно полегшує навігацію, виявлення і контроль сиріт.

27.7. Sitemap як контрольна площина

На великому програмному сайті sitemap — не просто спосіб “відправити URL Google”. Я використовую sitemap-файли як керовані сегменти inventory: за типом сторінки, країною, мовою, датою запуску або статусом. Це дозволяє зіставляти submitted URLs з indexed URLs і швидше локалізувати проблему.

У sitemap треба класти канонічні URL, які сайт реально хоче бачити в індексі. Якщо у файлі змішані 200, 301, noindex, canonical-to-other і 404, сам sitemap перестає бути корисним як діагностичний набір.

27.8. Сканування та індексація як обмеження системи

Коли генерується 100 000 URL за день, питання “чи можемо ми згенерувати?” стає менш важливим за “чи може пошукова система нормально обробити inventory?”. Google окремо попереджає, що фасетна навігація може створювати практично безмежну кількість URL, спричиняти зайве сканування й уповільнювати виявлення важливого контенту.[4]

Перед великим запуском я роблю ramp-up: невеликий сегмент, перевірка crawl/index/traffic, потім наступна партія. Масовий реліз без контрольної групи ускладнює діагностику: коли половина URL не індексується, незрозуміло, проблема в шаблоні, попиті, якості даних, внутрішніх посиланнях чи просто в темпі запуску.

27.9. AI у виробництві

AI корисний для перетворення структурованих даних у читабельні пояснення, генерації чернеток, перевірки пропущених полів, нормалізації назв, класифікації intent. Але AI не повинен вигадувати відсутні атрибути. Якщо поле `withdrawal_time` порожнє, модель не має “знати типове значення”. Порожнє поле має викликати fallback або блокування компонента.

На production-рівні я розділяю factual layer і linguistic layer. Факти приходять із контрольованого джерела й передаються моделі як immutable inputs. Модель може пояснити їх, але не змінювати. Після генерації автоматичний QA перевіряє числа, валюти, бренди, URL, дати й допустимі значення.

27.10. Pruning — частина продукту

Програмний сайт не можна тільки нарощувати. Потрібна політика видалення або консолідації. Я відстежую сторінки без показів, без унікальних даних, з нульовим inventory, з постійним canonical conflict, дублікати intent, застарілі сутності. Але “0 кліків за 90 днів” не є автоматичною підставою видалити URL: сезонність, long-tail і роль сторінки в архітектурі можуть бути важливішими.

Pruning має бути сегментним експериментом. Видалити 50 000 URL і потім пояснювати загальну зміну трафіку — слабкий дизайн. Краще сформувати однорідну групу, чіткий критерій, контрольний сегмент і період спостереження.

27.11. Що я вимірюю

Для programmatic SEO я не дивлюся тільки на загальний organic traffic. Мінімальний dashboard має показувати inventory, discoverability, crawl, indexation, impressions, clicks, conversions і freshness за типом сторінки. Тоді “50 000 сторінок дали +20% трафіку” можна розкласти: скільки реально проіндексовано, які шаблони дають результат, де є overcrawl і скільки URL не створили цінності.

Таблиця 27.2. Мінімальний контроль programmatic SEO.

Рівень	Метрика	Питання
Inventory	eligible URLs / generated URLs	Скільки сторінок пройшли gate цінності?
Виявлення	internal linked / sitemap only / orphan	Чи є URL частиною графа?
Сканування	Googlebot hits, status mix	Чи витрачається ресурс на потрібні URL?
Індекс	indexed / submitted by template	Які шаблони системно не індексуються?
Попит	impressions / indexed URL	Чи є реальна пошукова видимість?
Бізнес	conversions, revenue, leads	Чи окупається підтримка набору URL?
Якість	stale facts, missing fields, errors	Чи не масштабуємо помилки?

МІФ. Programmatic SEO — це спосіб швидко зробити мільйон сторінок. Насправді сильна система спочатку максимально жорстко вирішує, які сторінки не створювати.

Джерела до Глави 27

1. Google Search Central. Spam policies for Google Web Search. https://developers.google.com/search/docs/essentials/spam-policies — Scaled content abuse і doorway abuse.

2. Google Search Central. Consolidate duplicate URLs. https://developers.google.com/search/docs/crawling-indexing/consolidate-duplicate-urls — Canonicalization і консолідація дублів.

3. Google Search Central. Help Google understand your ecommerce site structure. https://developers.google.com/search/docs/specialty/ecommerce/help-google-understand-your-ecommerce-site-structure — Внутрішня структура та виявлення товарів.

4. Google Search Central Blog. Crawling December: Faceted navigation. https://developers.google.com/search/blog/2024/12/crawling-december-faceted-nav — Комбінаторний простір URL і overcrawl.

ЧАСТИНА VI / МАСШТАБ

ГЛАВА 28

Великі сайти: сканування, індексація, шаблони та очищення

На сайті з сотнями тисяч або мільйонами URL SEO перестає бути набором перевірок сторінок. Одиницею управління стає тип сторінки, шаблон, сегмент inventory і потік змін.

Стан даних і джерел: 14 вересня 2026 року

Великий сайт ламається не тому, що в нього “занадто багато сторінок”. Він ламається, коли команда не знає, які URL існують, навіщо вони існують, хто їх створює, як вони зв’язані, які з них має обробляти Google і як швидко можна помітити масову регресію. На 300-сторінковому сайті можна відкрити сторінку руками. На 30-мільйонному — потрібні системи.

Моя базова одиниця аналізу — page type. Product, category, filter, brand, article, location, profile, result page, discontinued item, pagination. Для кожного типу я хочу знати inventory, crawl behavior, indexation, visibility, conversion і owner. Якщо dashboard не може відповісти на це за 5 хвилин, сайт ще не керується як великий.

Рис. 28.1. Операційна модель великого сайту: кожний рівень вимірюється за типами сторінок.

28.1. Інвентаризація URL

Перший артефакт великого сайту — не crawl export, а URL inventory. Я збираю всі джерела: внутрішній crawl, sitemap, CMS/database, Search Console, server logs, analytics, redirects. Потім нормалізую URL і класифікую за page type та бізнес-статусом.

Це дає категорії, яких не видно з одного джерела: URL існує в базі, але не має внутрішніх посилань; URL відвідує Googlebot, але його вже немає в sitemap; сторінка отримує трафік, але CMS вважає її discontinued; редирект живе роками, хоча жодне внутрішнє посилання на нього вже не веде.

28.2. Page type як контракт

Для кожного шаблону я документую контракт: який HTTP status очікуємо; чи indexable; canonical; robots directives; sitemap inclusion; structured data; внутрішні link sources; expected content components; target country/language; primary conversion event. Це дозволяє автоматично тестувати не “сайт”, а правила.

Якщо product page раптом починає віддавати `noindex` через помилку release, 200 випадкових URL у QA можуть не зловити регресію. Контрактний тест на representative sample кожного template ловить її до масштабування.

Таблиця 28.1. Приклад контракту шаблону.

Поле	Категорія	Товар	Фільтр без попиту
HTTP	200	200	200 або відсутній URL
Indexability	так	так	ні
Canonical	self	self	на категорію або URL не створюється
Sitemap	так	так	ні
Internal links	меню/батько	категорія/бренд	не створюємо crawl path
Schema	за потреби	Product	зазвичай ні

28.3. Фасетна навігація

На великих ecommerce/marketplace сайтах найбільш типова причина URL explosion — фільтри. Google прямо називає faceted navigation найпоширенішим джерелом overcrawl-скарг, які бачить команда Search.[1] Проблема не у фільтрах як UX, а в тому, що кожна комбінація може стати crawlable URL.

Я ділю фасети на три групи: indexable landing pages з доведеним попитом; crawlable, але heindexable технічні стани, якщо вони потрібні продукту; і states, які взагалі не повинні створювати окремий URL для краулера. Головна помилка — дозволити frontend генерувати повний combinatorial graph, а потім намагатися “виправити SEO” canonical-тегами.

28.4. Sitemap sharding

На великих сайтах sitemap треба використовувати як вимірювальні сегменти. Я розділяю файли за типом сторінки, країною, мовою, іноді

за lifecycle. Тоді Page indexing / API / власні моніторинги можна зіставляти з конкретним inventory.

Sitemap-файли не повинні ставати архівом усіх URL, які колись існували. Вони мають описувати поточний канонічний набір URL, який команда хоче бачити в пошуку. Інакше submitted metric змішує активні сторінки з редиректами, noindex і 404.

28.5. Логи: фактичне сканування

Search Console Crawl Stats дає корисний агрегат, але логи відповідають на питання на рівні URL: що саме запитував bot, коли, з яким статусом, скільки байтів і як довго відповідав сервер. На великому сайті це єдиний спосіб побачити, наприклад, що Googlebot витрачає 35% запитів на параметричні URL, які команда не вважає важливими.

Не можна сліпо довіряти User-Agent. Для серйозного аналізу bots треба верифікувати згідно з рекомендаціями Google, а потім уже рахувати crawl frequency. Інакше “Googlebot” у логах може бути стороннім scraper.

28.6. Індексні сегменти

Мене цікавить не “у Google 2,3 млн сторінок”, а частка eligible URL за типом. 90% indexation для продуктів і 15% для category landing pages — зовсім інша проблема, ніж рівномірні 60% для всього сайту.

Я будую funnel: inventory → eligible → discovered → crawled → indexed → impressions. Якщо просідання між discovered і crawled, дивлюся crawl demand/architecture. Якщо crawled → indexed, дивлюся canonical/duplication/quality/soft 404. Якщо indexed → impressions, проблема вже в retrieval/relevance/competition, а не в “індексації”.

28.7. Template regression monitoring

Великі сайти часто втрачають SEO не через “Google update”, а через реліз. React component прибрав `<a href>`, CMS додала noindex, canonical почав вести на staging domain, JSON-LD перестав містити

price, hreflang генерує 404. Один deployment може торкнутися мільйонів URL.

Тому я тримаю synthetic SEO tests у CI/CD або хоча б щогодинному моніторингу representative URLs. Мінімум: status, robots, canonical, hreflang, title, H1, body presence, ключові structured-data properties, внутрішні посилання, render result. У production додаю alert тільки на зміни, які мають масштаб.

28.8. Очищення inventory

Великий сайт природно накопичує discontinued products, порожні categories, старі campaigns, профілі без контенту, query pages, duplicates. Але масове “pruning” без карти залежностей може знести сторінки, які передають внутрішню вагу, мають backlinks або сезонний попит.

Я приймаю рішення по класах. Якщо товар знято назавжди й немає еквівалента — 404/410 може бути правильним. Якщо є чіткий successor — redirect. Якщо category тимчасово порожня, рішення залежить від likelihood повернення inventory і користі сторінки. Якщо URL не має попиту, даних, links і ролі в навігації — часто краще не підтримувати його.

28.9. Продуктивність команди

На великому сайті ручний SEO-аудит — це sampling. Він потрібен, але не є системою контролю. Я автоматизую те, що детерміноване: extraction, classification, comparisons, contract tests, anomalies. Людина залишається для діагностики, пріоритизації і рішень із неоднозначними trade-offs.

Ключова метрика SEO-команди на такому сайті — не кількість знайдених “issues”, а час від регресії до виявлення і від виявлення до виправлення. Incident, який 10 хвилин торкався 2 млн URL, майже завжди дешевший за “невеликий canonical bug”, що жив шість тижнів.

28.10. Дані як warehouse, а не набір CSV

Після певного масштабу Excel перестає бути джерелом правди. Я зводжу Search Console bulk export, analytics, logs, inventory і crawler output у warehouse. Google прямо рекомендує BigQuery exports для детального спільного аналізу Search Console і GA, коли потрібна максимальна деталізація й мінімум розбіжностей.[2]

Не потрібно будувати data platform “бо це модно”. Поріг простий: якщо одна й та сама таблиця збирається вручну щотижня, якщо історія втрачається, якщо різні люди рахують metric по-різному — пора формалізувати pipeline.

НОТАТКА З ПРАКТИКИ. Найцінніший dashboard великого сайту часто дуже нудний: page type × inventory × indexability × indexed × impressions × conversions × change WoW. Він корисніший за десятки “SEO health scores”.

Джерела до Глави 28

1. Google Search Central Blog. Crawling December: Faceted navigation. https://developers.google.com/search/blog/2024/12/crawling-december-faceted-nav — Ризик combinatorial URL space і overcrawl.

2. Google Search Central. Using Search Console and Google Analytics data for SEO. https://developers.google.com/search/docs/monitor-debug/google-analytics-search-console — Спільний аналіз і BigQuery exports.

3. Google Search Central. Ecommerce site structure. https://developers.google.com/search/docs/specialty/ecommerce/help-google-understand-your-ecommerce-site-structure — Граф внутрішніх посилань.

4. Google Search Central. Ecommerce URL structure. https://developers.google.com/search/docs/specialty/ecommerce/designing-a-url-structure-for-ecommerce-sites — URL, canonical, filters, pagination.

ЧАСТИНА VI / МАСШТАБ

ГЛАВА 29

Багатодоменне SEO: портфелі сайтів, GEO та операційний контроль

Декілька доменів можуть бути виправданою бізнес-архітектурою, а можуть бути дорогим способом розмножити один і той самий сайт. Рішення треба приймати через різницю продукту, бренду, країни й операційної спроможності, а не через віру в “більше доменів — більше SERP”.

Стан даних і джерел: 14 вересня 2026 року

Multi-domain SEO особливо поширене в affiliate, marketplaces, SaaS із різними брендами, regulated verticals і міжнародних проєктах. На рівні презентації все виглядає просто: один домен на бренд або країну. На рівні операцій кожний додатковий домен множить DNS, hosting, Search Console, analytics, monitoring, legal pages, content updates, links, migrations, incidents і QA.

Google spam policies прямо описують doorway abuse як створення кількох схожих сайтів або сторінок для захоплення близьких запитів і перенаправлення користувача до фактичної цілі.[1] Тому “портфель” не є автоматично нейтральною структурою. Чим менше реальної різниці між доменами, тим слабше бізнесове виправдання їх існування.

Рис. 29.1. Багатодоменна стратегія має проходити перевірку на реальну відмінність і операційну спроможність.

29.1. Коли окремий домен виправданий

Сильні причини: юридично окремий бренд, окремий продукт, суттєво інший ринок із власною командою і контентом, різна регуляція, acquisition або продаж бізнесу, технічна ізоляція, яка має реальну ціну на користь незалежності. Слабка причина — “хочемо зайняти ще один результат”.

Для міжнародного SEO окремі ccTLD можуть бути логічними, але не є єдиним варіантом. Підкаталоги часто простіші для консолідації authority, deployment і measurement. Архітектура повинна виходити з бізнесу, а не з магії доменної зони.

29.2. Вартість портфеля

Якщо у вас 20 доменів, зміна privacy page, schema, tracking або footer link — це не одна задача, а distribution problem. Я рахую не тільки domain/hosting cost, а engineering hours, content maintenance, link acquisition, monitoring, compliance і incident surface.

Саме тому маленькі affiliate portfolios часто мають технічний борг, який не видно в P&L: half-broken analytics, expired certificates, stale bonuses, noindex після redesign, різні canonical rules. Це не “дрібниці”, а множник ризику.

Таблиця 29.1. Прихована ціна додаткового домену.

Контур	Що множиться	Як вимірювати
Технічний	DNS, TLS, hosting, deploy, monitoring	інциденти/місяць, час підтримки
SEO	GSC, sitemaps, crawl, index QA	час аудиту, coverage
Контент	оновлення, локалізація, legal claims	stale pages, редакторські години
Authority	links, mentions, brand demand	витрати/домен, referring domains
Аналітика	GA4, consent, events, attribution	відсутні/різні events
Compliance	регуляція, disclosures, age/geo rules	помилки, ручні перевірки

29.3. Cannibalization між доменами

Два домени можуть конкурувати за однакову query space. Це не завжди погано: різні бренди справді можуть бути конкурентами. Але якщо вони належать одній команді й мають однакову пропозицію, ви самі розподіляєте links, content і brand signals між копіями.

Я будую cross-domain query matrix: які domains отримують impressions за одні й ті самі query clusters, де ranking URLs змінюються, де один домен витісняє інший, чи є різниця conversion rate. Іноді другий домен дає incremental reach. Іноді це просто дорожчий спосіб отримати той самий трафік.

29.4. Локалізація має бути реальною

Multi-GEO не означає “перекласти шаблон”. Країни відрізняються валютою, payment methods, регуляцією, delivery, availability, terms,

support, competitors, brand recognition. Сторінка для Іспанії й сторінка для Німеччини мають різнитися не мовою, а фактичним продуктом.

Якщо одна команда не може підтримувати локальні факти, hreflang не зробить сторінку локально корисною. Hreflang допомагає співвіднести локалізовані версії, але не створює релевантність сам по собі.

29.5. Expired domains: відновлення активу vs abuse

Купівля старого домену сама по собі не заборонена. Google визначає expired domain abuse через ціль: домен купують і перепрофільовують переважно для маніпуляції rankings, розміщуючи low-value content. В офіційних прикладах є casino content на колишньому сайті початкової школи.[1]

Тому в оцінці дропа мене цікавить не DR. Я дивлюся історію тематики, backlinks, redirects, власників, індексацію, spam anchors, legal/trademark risk і чи має новий продукт правдоподібний зв’язок із минулим. Навіть сильний link graph не перетворює нерелевантне перепрофілювання на довгостроковий asset.

29.6. Міграції між доменами

Коли портфель консолідується, міграція повинна мати URL mapping, 1:1 redirects там, де є еквівалент, оновлені internal links, canonicals, hreflang, sitemaps, analytics і monitoring. Google попереджає, що під час site move rankings можуть коливатися, поки система recrawl/reindex URL; більші сайти можуть відновлюватися довше.[2]

Не варто редиректити тисячі нерелевантних сторінок на homepage лише “щоб передати вагу”. Redirect має вести до найближчого реального еквівалента. Якщо еквівалента немає, 404/410 може бути чистішим рішенням.

29.7. Єдина операційна площина

Портфель починає бути керованим, коли у всіх сайтів одна telemetry model: однакові event names, dashboard schema, uptime

checks, canonical tests, content freshness fields, ownership. Не обов'язково один hosting або CMS; важлива однакова мова контролю.

Я зводжу domains у registry: owner, purpose, geo, brand, CMS, DNS provider, analytics property, Search Console, release channel, renewal date, regulatory scope, critical contacts. Це банально, але саме відсутність такого реєстру перетворює 50 доменів на хаос.

29.8. Портфель як капітал

У affiliate бізнесі домен може бути активом із traffic, backlinks, brand demand і cash flow. Тому рішення "залишити / інвестувати / об'єднати / продати / закрити" краще приймати як portfolio management.

Я оцінюю contribution margin домену після реальних SEO-витрат, а не тільки gross revenue. Сайт із $2k revenue і $1.8k links/content/maintenance — не такий самий актив, як сайт із $1.5k revenue і $200 підтримки. Це важливо, коли бюджет обмежений.

29.9. Ризикова межа

Коли десятки майже однакових доменів створені лише для захоплення схожих SERP і ведуть до тієї самої пропозиції, ви наближаєтеся до doorway pattern, який Google прямо описує у spam policies.[1] Я не оцінюю це морально; це питання risk/reward. Але називати таку модель "безпечним white-hat multi-domain SEO" — неправильно.

Для кожної тактики в портфелі я фіксую: policy status, ймовірність detection/enforcement, blast radius, recoverability і бізнесову залежність. Risk management починається з чесного називання механіки.

> **ПОПЕРЕДЖЕННЯ.** Не плутайте технічну ізоляцію портфеля з гарантією незалежності в пошуку. Немає публічно підтвердженого правила, що окремий IP, CMS або registrar автоматично робить сайти "незв'язаними" для Google.

Джерела до Глави 29

1. Google Search Central. Spam policies. https://developers.google.com/search/docs/essentials/spam-policies — Doorway abuse, expired domain abuse, scaled content abuse.

2. Google Search Central. Site moves with URL changes. https://developers.google.com/search/docs/crawling-indexing/site-move-with-url-changes — Міграції між доменами.

3. Google Search Central. International sites. https://developers.google.com/search/docs/specialty/international/managing-multi-regional-sites — Варіанти міжнародної архітектури.

4. Google Search Central. Localized versions / hreflang. https://developers.google.com/search/docs/specialty/international/localized-versions — Зв’язування мовних і регіональних версій.

ЧАСТИНА VI / МАСШТАБ

ГЛАВА 30

Автоматизація SEO: від Excel до API, Python, SQL і BigQuery

Автоматизувати треба повторювані детерміновані операції, а не саме мислення. Сильна SEO-автоматизація збирає факти, нормалізує їх, знаходить відхилення й залишає людині рішення.

Стан даних і джерел: 14 вересня 2026 року

Більшість SEO-команд починає з таблиць. Це нормально. Проблема починається, коли таблиця стає ручним data pipeline: щопонеділка хтось завантажує GSC, вставляє GA4, копіює rankings, оновлює VLOOKUP, фарбує червоні клітинки й робить скріншот у Slack. Така “звітність” дорога, крихка і не відтворюється.

Автоматизація для мене має три критерії: операція повторюється; вхід і вихід можна формально описати; помилка автоматизації помітна й контрольована. Стратегічна пріоритизація, interpretation або content judgement можуть використовувати автоматичні підказки, але не мають зникати за `if/else`.

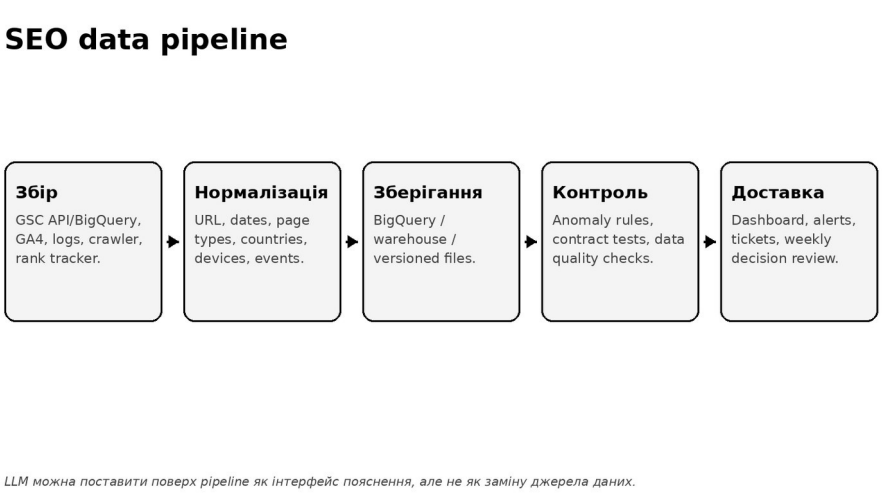


Рис. 30.1. Автоматизація SEO як pipeline: збір → нормалізація → зберігання → контроль → рішення.

30.1. Що автоматизувати першим

Я починаю з задач, де найбільше людського копіювання: exports, URL normalization, регулярні comparisons, template checks, anomaly detection. Потім — класифікація page types, mapping queries, internal-link suggestions. І лише після цього — генеративні задачі.

Хороша автоматизація повинна економити senior time, а не створювати новий dashboard, який ніхто не читає. Перед розробкою я рахую minutes per run × runs per month × loaded cost. Якщо скрипт економить 40 хвилин раз на квартал, його підтримка може коштувати дорожче.

30.2. GSC: UI, API і BigQuery — різні режими

Search Console UI зручний для діагностики руками, API — для керованих запитів і інтеграцій, bulk export — для повної історичної аналітики в BigQuery. Google документує, що Search Analytics API має default 1 000 rows, `rowLimit` до 25 000 і pagination через `startRow`; верхня доступна межа даних для API описувалася як до 50 000 рядків на день на сайт і search type.[1][2]

Bulk export щодня вивантажує performance data в BigQuery без щоденного row limit, але anonymized queries вилучаються з міркувань privacy.[3] Це робить bulk export кращою базою для warehouse і довгих trend analysis, якщо обсяг виправдовує BigQuery.

30.3. GA4 не є заміною GSC

GSC вимірює взаємодію з Google Search, GA4 — поведінку після завантаження вашого сайту. Через різні визначення, timezone, canonicalization, privacy і tracking coverage цифри не повинні збігатися 1:1. Google прямо рекомендує використовувати обидва джерела разом і, для максимальної деталізації, зводити bulk exports у BigQuery.[4]

Автоматизація повинна зберігати source semantics. Не можна створити “єдину правду”, просто склавши GSC clicks і GA sessions у одну колонку. Це різні події.

30.4. Схема даних важливіша за інструмент

Поганий pipeline легко перенести з Excel у Python і отримати автоматизований хаос. Перед кодом я визначаю canonical dimensions: `date`, `site`, `url`, `page_type`, `country`, `device`, `query_cluster`, `channel`, `event`. Потім кожне джерело мапиться до цієї схеми.

URL normalization має бути детермінованою: protocol, host, trailing slash, parameters, case, fragments. Якщо crawler і GSC називають одну сторінку по-різному, joins даватимуть фальшиві “missing URLs”.

30.5. Anomaly detection без магії

Не кожне -20% WoW — інцидент. Я порівнюю з day-of-week, seasonality, rolling baseline і update dates. На ранньому етапі прості правила часто кращі за складну ML-модель: якщо clicks падають >30% YoY у page type із >10k baseline і impressions теж падають, створити alert; якщо clicks падають, а impressions стабільні — перевірити CTR/SERP.

Alert повинен вести до action. Якщо команда отримує 200 повідомлень на день, система автоматично навчає людей їх ігнорувати.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ. У мене є два CSV з Search Console: gsc_old.csv і gsc_new.csv. Колонки: page, clicks, impressions. Порівняй періоди по URL, агрегуй дублікати, порахуй абсолютну й відносну зміну, окремо покажи URL із baseline >=100 clicks і падінням >=30%. Не трактуй причину автоматично: додай колонку “що перевірити” на основі комбінації clicks/impressions.

АЛЬТЕРНАТИВА БЕЗ ПРОГРАМУВАННЯ. У Google Sheets імпортуйте два експорти на окремі аркуші. Через QUERY або зведену таблицю агрегуйте page, далі XLOOKUP підтягніть старі clicks/impressions і порахуйте `(new-old)/old`. Фільтр: old clicks >=100 та зміна <=-30%.

QR: компактна версія коду для копіювання без передруковування.

КОД 30.1. Простий контроль аномалій між двома GSC-експортами.

```
# Порівняння двох CSV-експортів GSC за URL.
# Очікувані колонки: page, clicks, impressions.
import pandas as pd

old = pd.read_csv("gsc_old.csv")
new = pd.read_csv("gsc_new.csv")

# Агрегуємо на випадок кількох рядків на URL.
def prep(df):
    return (df.groupby("page", as_index=False)[["clicks", "impressions"]]
              .sum())

a = prep(old).rename(columns={"clicks":"clicks_old", "impressions":"imp_old"})
b = prep(new).rename(columns={"clicks":"clicks_new", "impressions":"imp_new"})

m = a.merge(b, on="page", how="outer").fillna(0)

# Відносна зміна. Для нульового baseline залишаємо NaN.
m["click_change"] = (m.clicks_new - m.clicks_old) / m.clicks_old.replace(0,
pd.NA)
m["imp_change"] = (m.imp_new - m.imp_old) / m.imp_old.replace(0, pd.NA)

# Приклад порога: старий baseline >= 100 кліків і падіння >= 30%.
alerts = m[(m.clicks_old >= 100) & (m.click_change <= -0.30)]
alerts.sort_values("click_change").to_csv("gsc_alerts.csv", index=False)
print(alerts.head(30))
```

30.6. Оркестрація

Коли скриптів більше трьох, потрібен scheduler і журнал виконання. Це може бути GitHub Actions, Cloud Run/Functions, cron, Airflow, Make/n8n — інструмент залежить від масштабу. Важливі retries, timestamp, input version, output location і alert при failure.

Я зберігаю raw data окремо від transformed data. Якщо логіка розрахунку змінилася, raw дозволяє перебудувати історію. Якщо ви зберігаєте лише фінальний Excel, зміна формули робить попередні місяці непорівнюваними.

30.7. Версіонування SEO-логіки

Regex, page-type rules, redirects, schema generators, prompt templates — це код або конфігурація, а значить, мають жити у version control. Git дає answer на питання “хто і коли змінив правило”, а не тільки latest file у Drive.

Для критичних змін я додаю test cases. Наприклад, page classifier має список 50 URL з очікуваними типами. Якщо нове regex правило раптом класифікує `/casino/brand/review/` як category, тест впаде до production.

30.8. LLM як інтерфейс до даних

LLM може генерувати SQL, пояснювати anomaly, перетворювати natural-language question у query. Але production pipeline не повинен залежати від того, що модель “пам’ятає” schema. Я передаю schema, дозволені таблиці, приклади й обмеження, а SQL проходить validation.

Для необоротних дій — зміна redirects, видалення URL, publishing — потрібен human approval. AI може підготувати diff, але не має автоматично “прибирати сміття”, якщо definition сміття не формальна.

30.9. Секрети і доступи

API keys, service-account JSON, database passwords не повинні лежати в коді або prompt history. Використовуйте secret storage, environment variables, мінімальні permissions і ротацію. Особливо це важливо, коли LLM отримує доступ до файлів або репозиторіїв.

SEO automation швидко стає частиною production infrastructure. Її потрібно захищати так само, як інші internal tools, а не як “невинний скриптик маркетолога”.

30.10. Що має автоматизувати SEO Lead

Мій пріоритет: data collection, monitoring, recurring audits, changelogs, anomaly detection, content inventory, link inventory, indexation segmentation, release QA. Не автоматизую фінальне рішення про intent, brand positioning, risk acceptance і бюджет лише тому, що це можливо.

Найкращий результат автоматизації — команда витрачає менше часу на добування цифр і більше на постановку гіпотез та перевірку рішень.

Джерела до Глави 30

1. Google Search Console API. Query Search Analytics data. https://developers.google.com/webmaster-tools/v1/how-tos/search_analytics — Pagination і rowLimit.

2. Google Search Central Blog. Performance data deep dive. https://developers.google.com/search/blog/2022/10/performance-data-deep-dive — Ліміти API і природа даних.

3. Google Search Central Blog. Bulk data export. https://developers.google.com/search/blog/2023/02/bulk-data-export — Щоденний BigQuery export без daily row limit; anonymized queries вилучаються.

4. Google Search Central. Search Console + Analytics. https://developers.google.com/search/docs/monitor-debug/google-analytics-search-console — Спільний аналіз джерел.

ЧАСТИНА VI / МАСШТАБ

ГЛАВА 31

AI для SEO-операцій: де він економить час, а де створює помилки

AI найсильніший не там, де “пише SEO-текст”, а там, де скорочує шлях від сирих даних до перевіреної гіпотези: класифікує, витягує, перетворює, пояснює й допомагає написати код.

Стан даних і джерел: 14 вересня 2026 року

Я не поділяю SEO-задачі на “можна AI / не можна AI”. Я поділяю їх за tolerance до помилки. Якщо модель неправильно класифікує 2% із 100 000 keywords, це може бути прийнятно для exploration, але неприйнятно для автоматичної зміни canonical на production.

Тому головний параметр AI workflow — не модель і не prompt. Це verification cost. Якщо результат легко перевірити правилом, schema або контрольним набором, AI добре масштабується. Якщо correctness суб’єктивна або помилка дорога, потрібен сильніший human review.

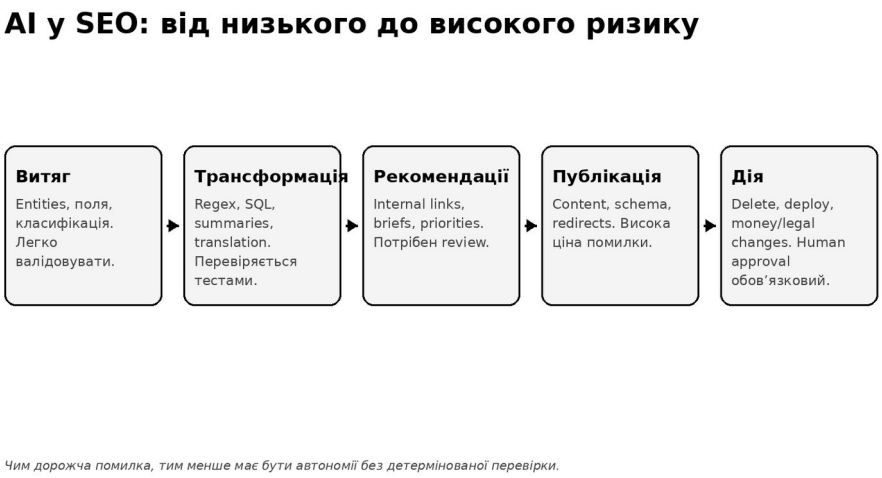


Рис. 31.1. Рівень автономії AI має залежати від вартості помилки й можливості перевірити результат.

31.1. Clustering: не просіть “зроби кластери”

Keyword clustering через LLM корисний для semantic grouping, але без чіткої unit of clustering результат плаває. Одні й ті самі keywords можна кластеризувати за SERP overlap, intent, entity, page type або funnel stage. Це різні задачі.

Я задаю hierarchy: спочатку market/task, потім entity, потім intent, потім desired page type. Для high-value cluster перевіряю SERP руками або через дані. LLM не має доступу до фактичного SERP, якщо ви його не передали.

31.2. SERP classification

Модель добре класифікує вже зібрані SERP features: commercial/listicle/product/local/forum/video. Але prompt повинен отримувати title, URL, snippet, features, а не тільки keyword. Інакше це intent guessing за фразою.

На sample із 100–300 queries я роблю manual labels, потім рахую confusion matrix. Якщо “review” і “comparison” постійно плутаються, змінюю taxonomy, а не просто модель.

31.3. Content briefs

AI може швидко підсумувати конкурентні coverage, entities, questions і gaps. Але brief стає корисним лише після додавання власної позиції: first-party data, експеримент, product knowledge, constraints, source requirements. Інакше brief оптимізує автора на середнє арифметичне SERP.

Я прямо додаю секцію “що НЕ повторювати”, щоб commodity paragraphs не займали половину статті. Для factual topics додаю список primary sources, які мають бути перевірені до drafting.

31.4. Entity extraction і нормалізація

Це один із найкращих AI use cases. З тисяч сторінок можна витягнути brand, country, payment methods, currencies, regulators, product attributes. Але output краще змусити відповідати JSON schema з enum, nullable fields і source snippets.

Після extraction я запускаю deterministic validation: unknown enum, invalid dates, impossible currency/country pair, missing source. Модель не має “виправляти” невідоме значення без позначки uncertainty.

31.5. Regex, SQL і Python

LLM різко знижує поріг входу в технічний аналіз. SEO може описати задачу, отримати SQL і розібрати його з поясненням. Але код треба тестувати на sample. Найнебезпечніший формат помилки — синтаксично правильний SQL, який рахує не те.

Я прошу модель спочатку описати assumptions і grain таблиці, потім SQL, потім три sanity checks. Якщо query агрегує GSC, треба прямо уточнити dimensions, canonical URL behavior, date timezone і denominator. “Працює без error” не означає “правильно”.

31.6. Log analysis

LLM може допомогти написати parser, regex для bots, класифікувати URL patterns, пояснити anomaly. Але raw logs краще агрегувати детерміновано, а моделі передавати summaries. Це дешевше, приватніше і контрольованіше.

Наприклад, замість мільйона рядків передаємо top page types by Googlebot hits, status distribution, response-time percentiles, top parameters, day-over-day change. Модель допомагає сформувати hypotheses, а факт залишається в aggregate data.

31.7. Internal linking

AI добре генерує candidate anchors і semantic matches, але не знає вашої graph strategy, якщо її не описати. Я передаю source URL, target URL, title/H1, page type, existing anchors, priority score і exclusions.

Автоматичне вставлення links без QA легко створює circular boilerplate, надлишкові anchors або links у нерелевантних місцях. Тому AI пропонує candidates, а deterministic layer перевіряє duplicates, max links, target status і canonical.

31.8. Localization

Модель корисна як перший шар перекладу й локалізації, але regulated verticals потребують local reviewer або primary-source checks. Currency, tax, legal age, bonus terms, licensing і payment methods не можна “перекласти” з іншого GEO.

Я відділяю language QA від market QA. Текст може бути ідеально іспанським і фактично неправильним для Іспанії.

31.9. Evaluation set

Без eval set AI workflow перетворюється на смак. Я створюю невеликий набір реальних examples із expected output і edge cases. Для classification рахую accuracy/F1; для extraction — field-level

precision/recall; для generation — factual error rate, citation coverage, style violations.

Коли змінюється модель або prompt, проганяю той самий set. Це єдиний спосіб зрозуміти, чи “новіша модель” реально краща саме для нашої задачі.

Таблиця 31.1. Приклад контролю AI-задач.

Задача	Що вимірювати	Автоматична перевірка	Людський review
Keyword classification	accuracy/F1	enum, required fields	edge cases
Entity extraction	precision/recall	schema, source span	невідомі значення
SQL generation	correct result	unit tests, row counts	business logic
Content draft	factual error rate	source/number checks	позиція, стиль, ризик
Internal links	precision candidates	status/canonical/ duplicates	semantic fit

31.10. Hallucination budget

Фраза “AI іноді галюцинує” занадто абстрактна. Я задаю допустимий error budget. Для brainstorm 10% помилкових ідей не страшні. Для фінансового розрахунку 0,1% може бути неприйнятним. Це одразу визначає архітектуру QA.

У high-stakes factual content модель не повинна генерувати claim без source field. Якщо source відсутній, output має бути `SOURCE_REQUIRED`, а не правдоподібний текст.

31.11. Prompt injection і дані

Коли AI читає web pages, connected docs або user-generated text, вхід може містити інструкції, які конфліктують із нашою задачею. Тому external content треба трактувати як data, а не authority.

Не передавайте в модель secrets, приватні клієнтські дані й production credentials без відповідних enterprise controls. AI-інтеграція — це частина data governance, а не тільки productivity feature.

31.12. Де AI не дає переваги

Якщо всі competitors можуть за 30 секунд отримати однаковий “SEO article”, саме використання моделі не є moat. Перевагу створюють inputs, workflow, data, testing, distribution і speed of iteration.

Тому я оцінюю AI не за кількістю створеного контенту, а за скороченням cycle time: від питання до перевіреної відповіді, від anomaly до діагнозу, від dataset до production page, від brief до published evidence.

> **МІФ.** “AI замінить SEO-команду” — погана операційна модель. Реалістичніше: команда, яка вміє будувати перевірені AI-workflows, замінить частину ручних операцій команди, яка цього не робить.

Джерела до Глави 31

1. Google Search Central. Creating helpful, reliable, people-first content. https://developers.google.com/search/docs/fundamentals/creating-helpful-content — Вимоги до корисності й надійності output.

2. Google Search Central. AI optimization guide. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide — Generative AI не створює окремої магічної SEO-дисципліни.

3. Google Search Central. Spam policies. https://developers.google.com/search/docs/essentials/spam-policies — Scaled content abuse оцінюється за метою і цінністю, а не самим фактом AI.

ЧАСТИНА VII / ВИМІРЮВАННЯ

ГЛАВА 32

Google Search Console: що показує, чого не показує і як не брехати собі даними

Search Console — найближче до first-party telemetry Google Search, але це не “повна база всіх запитів”. У сильному аналізі важливо знати не тільки метрики, а й правила агрегації, privacy limits, canonical attribution і різницю між UI, API та bulk export.

Стан даних і джерел: 14 вересня 2026 року

Я майже не роблю SEO-рішень без Search Console, але ще рідше вірю одному скріншоту з Performance report. GSC показує Google Search очима Google: impressions, clicks, CTR, position, query/page/country/device/search appearance. Це критично важливо. Водночас дані агрегуються, частина queries прихована через privacy, UI має row limits, а page data часто прив’язується до canonical URL.

Тому головна навичка роботи з GSC — не “вміти поставити фільтр”, а розуміти grain кожної таблиці й не порівнювати непорівнюване.

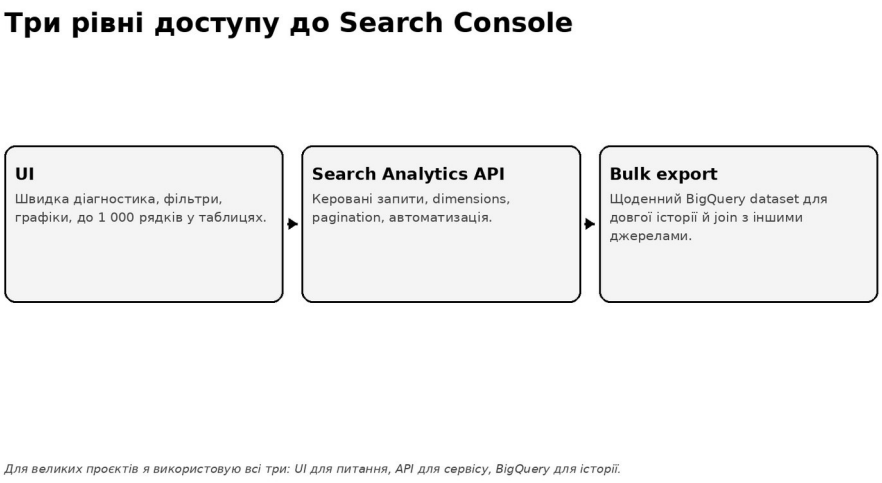


Рис. 32.1. UI, API і bulk export вирішують різні задачі, хоча походять із тієї самої системи.

32.1. Impression не дорівнює перегляду сторінки

Impression у Search Console — подія показу search result за правилами конкретної поверхні, а не pageview. Click — перехід із Google Search. CTR = clicks / impressions. Position — середня topmost position вашого результату в наборі, а не "позиція keyword" у класичному rank-tracker sense.

У SERP з різними features одна average position може приховувати зовсім різний click opportunity. Тому я не прогнозую traffic лише з "position 3". Дивлюся query type, device, country, search appearance і, де можливо, SERP composition.

32.2. Query tables не дорівнюють усім queries

Search Console приховує частину queries із privacy reasons. У bulk export anonymized queries вилучаються, а в агрегованих totals можуть бути дані, які не можна розкласти до конкретного query.[1] Це означає, що сума рядків query table може не дорівнювати total chart.

Висновок простий: “інші/невідомі queries” — не помилка Excel. Не треба штучно розподіляти їх пропорційно по відомих queries і видавати це за факт.

32.3. Canonical attribution

Page-level Search Console data зазвичай призначається canonical URL. Це особливо важливо під час міграцій, дублів, mobile/desktop alternates і canonical conflicts. Якщо ви шукаєте old URL і не бачите clicks, це не доказ, що він не мав visibility — дані могли бути attributed canonical.

Перед page-level аналізом я додаю crawler/GSC URL Inspection canonical mapping. Інакше “сторінки без clicks” можуть виявитися duplicate URLs, чия performance сидить на іншій адресі.

32.4. UI: сильний для діагностики, слабкий як warehouse

UI хороший для швидкого comparison: останні 28 днів vs попередні, country/device split, query/page filter, regex. Але table row limit означає, що export не є повним long-tail dataset. Google документує звичайне обмеження таблиць 1 000 rows, яке також поширюється на generative AI report.[2]

Тому “я вивантажив CSV із GSC” не означає “я маю всі queries”. Для великих сайтів це лише верхівка distribution.

32.5. Search Analytics API

API дозволяє автоматизувати запити й pagination. Офіційна документація показує `rowLimit` до 25 000 і `startRow` для наступної порції; у Search Central deep dive також описувалась доступність до 50 000 рядків на день на site/search type у цьому інтерфейсі.[3][4]

Я використовую API, коли потрібен сервісний запит: щодня top pages по country/device, автоматичне оновлення внутрішнього

dashboard, невеликий monitoring. Для повного багаторічного long-tail analysis краще bulk export.

32.6. Bulk export у BigQuery

Bulk export щодня створює `searchdata_site_impression`, `searchdata_url_impression` та `ExportLog`. URL table дає детальний query × URL рівень і search appearance fields; site table — property aggregation. Експорт не обмежений daily row limit Search Analytics API, але anonymized queries вилучаються.[1]

Я зберігаю цей dataset як історичний факт. Не роблю щотижневі CSV snapshots, які втрачають long tail. Потім можу приєднати CMS inventory, page type, GA4 revenue, log data і content changes.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ. У мене налаштований Search Console bulk export у BigQuery. Напиши SQL для `searchdata_url_impression`: останні 28 днів, search_type=WEB, групування по URL, clicks, impressions, CTR, average position. Врахуй, що `sum_position` zero-based, тому до середньої позиції треба додати 1. Не вигадуй назву project.dataset — залиш placeholder.

АЛЬТЕРНАТИВА БЕЗ ПРОГРАМУВАННЯ. У Search Console відкрийте Performance → Pages, виберіть 28 днів і експортуйте до Sheets. Це підходить для top-1000 сторінок. Якщо потрібен повний long tail, UI не замінює bulk export.

QR: компактна виконувана версія коду.

КОД 32.1. URL-level performance з Search Console bulk export.

```
-- Приклад: performance за URL за останні 28 днів із GSC bulk export.
-- Замініть project.searchconsole на свій project.dataset.
SELECT
  url,
  SUM(clicks) AS clicks,
  SUM(impressions) AS impressions,
  SAFE_DIVIDE(SUM(clicks), SUM(impressions)) AS ctr,
  SAFE_DIVIDE(SUM(sum_position), SUM(impressions)) + 1 AS avg_position
FROM `project.searchconsole.searchdata_url_impression`
WHERE data_date BETWEEN DATE_SUB(CURRENT_DATE(), INTERVAL 28 DAY) AND
CURRENT_DATE()
  AND search_type = 'WEB'
GROUP BY url
HAVING impressions >= 100
ORDER BY clicks DESC;
```

32.7. BigQuery cost discipline

Search Console tables partitioned by `data_date`, тому date filter треба ставити завжди. Google окремо рекомендує обмежувати partition range, фільтрувати рано, під час розробки використовувати sampling і, де допустимо, approximate functions.[5]

Я не підключаю Looker Studio прямо до сирого dataset для кожного графіка. Роблю pre-aggregated tables: daily site, daily page type, weekly query cluster. Це дешевше, швидше й стабільніше.

32.8. Generative AI report 2026

Станом на 31 серпня 2026 Google розгорнув Search Generative AI performance report worldwide.[2] Він показує generative AI impressions і дозволяє аналізувати Pages, Countries, Dates, Devices. Важливе обмеження: це не query report і не окремий click/CTR/position report.

У chart data aggregation property-level: якщо два results того самого site з’явилися в generative feature, це може рахуватися як одна property impression за описаними правилами. При URL filter aggregation змінюється. Тому метрику треба читати буквально, а не називати її “AI rank”.

32.9. Search Console data anomalies

Перед панікою через різкий провал я перевіряю Search Console Data Anomalies. У 2026 Google публікував, наприклад, logging issues для Generative AI report і Discover. Якщо anomaly в logging, “SEO recovery plan” не потрібен.

Це маленька звичка з великим ROI: спочатку перевірити, чи не змінився measurement layer, потім шукати проблему сайту.

32.10. Regex як аналітичний інструмент

Regex у GSC корисний не для “магічних keyword hacks”, а для швидкої сегментації: branded/non-brand, question patterns, page folders, product IDs. Я зберігаю основні regex у repository, щоб місячні звіти не залежали від того, як аналітик цього разу написав `brand|бренд|brand typo`.

Для важливих сегментів regex краще замінити на taxonomy у warehouse: query cluster або page type. Regex — хороший exploration layer, але слабке джерело довгострокової істини.

32.11. Що GSC не показує

Search Console не дає conversion value, complete user journey, повний SERP layout, backlinks, competitor rankings, усі queries, causal explanation ranking changes. Генеративний report не показує exact user query. Усе це треба добирати іншими системами.

Саме тому GSC — ядро SEO measurement, але не повна SEO-аналітика. Сильний стек об’єднує Search Console з GA4/business data, logs, crawler і зовнішнім SERP measurement.

Таблиця 32.1. Яке джерело Search Console коли використовувати.

Задача	UI	API	Bulk export
Швидка діагностика	найкраще	зайве	зайве
Регулярний невеликий звіт	можна	найкраще	можна

Задача	UI	API	Bulk export
Повний long-tail dataset	ні	обмежено	найкраще
Багаторічна історія	обмежено	треба зберігати	найкраще
Join із CMS/GA4	ручний	можна	найкраще
Generative AI exposure	окремий UI report	залежить від доступності інтерфейсу	не підміняти офіційний report припущеннями

НОТАТКА З ПРАКТИКИ. Якщо ваш GSC dashboard не може пояснити, чому total chart не дорівнює сумі visible query rows, команда ще не готова робити складні висновки з цих даних.

Джерела до Глави 32

1. Google Search Central Blog. Bulk data export. https://developers.google.com/search/blog/2023/02/bulk-data-export — Схема tables і privacy limits.

2. Search Console Help. Generative AI performance report. https://support.google.com/webmasters/answer/16984139?hl=en — Global rollout 31.08.2026, dimensions і limits.

3. Search Console API. Search Analytics. https://developers.google.com/webmaster-tools/v1/how-tos/search_analytics — rowLimit і pagination.

4. Google Search Central Blog. Performance data deep dive. https://developers.google.com/search/blog/2022/10/performance-data-deep-dive — UI/API limits і aggregation.

5. Google Search Central Blog. BigQuery efficiency tips. https://developers.google.com/search/blog/2023/06/bigquery-efficiency-tips — Partition filters і cost optimization.

6. Search Console Help. Data anomalies. https://support.google.com/webmasters/answer/6211453 — Known logging/processing anomalies.

ЧАСТИНА VII / ВИМІРЮВАННЯ

ГЛАВА 33

Серверні логи: реальна поведінка краулера замість припущень

Логи не кажуть, чому Google ранжує сторінку. Вони кажуть щось інше й не менш важливе: які URL бот реально запитував, коли, з яким статусом і з якою ціною для сервера.

Стан даних і джерел: 14 вересня 2026 року

У crawler export ми бачимо, що може пройти наш crawler. У Search Console — агреговану telemetry Google. У server log — конкретний HTTP request. Саме тому logs незамінні для великих сайтів, migrations, crawl diagnostics і випадків, коли “Google чомусь не бачить сторінки”.

Але log analysis легко зробити псевдоточним. User-Agent можна підробити, reverse proxy може ховати client IP, CDN може мати інший формат, кеш може не потрапляти в origin logs. Перед аналізом треба зрозуміти, який саме шар записує requests.

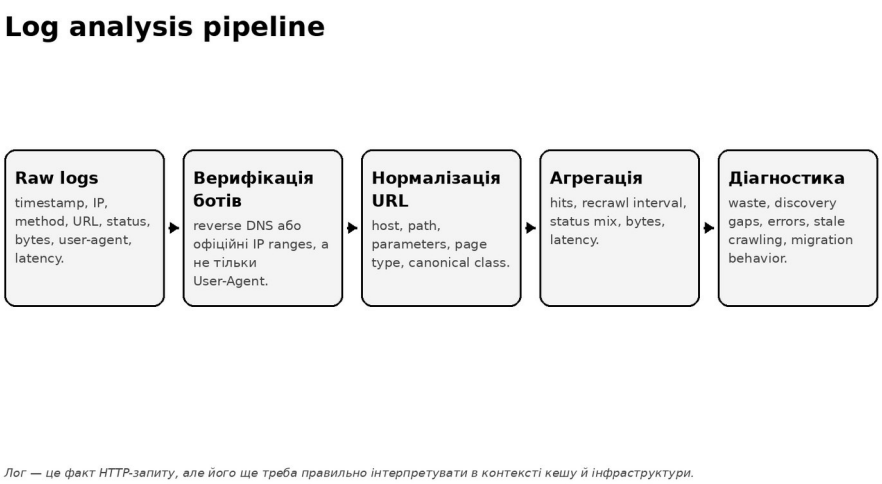


Рис. 33.1. Від сирого access log до SEO-діагностики.

33.1. Верифікуйте Googlebot

Google прямо попереджає, що User-Agent Googlebot часто імітують інші crawlers. Найнадійніші підходи — reverse DNS verification або зіставлення IP з опублікованими Googlebot IP ranges.[1]

Я не роблю висновок “Googlebot атакує параметри” на невіріфікованому user-agent. На high-traffic сайті це може повністю спотворити distribution.

33.2. Мінімальні поля

Для SEO потрібні timestamp, client IP, HTTP method, host, URL/path+query, status, response bytes, response time, user-agent. Referrer корисний не завжди, але зберігаю, якщо є. Якщо CDN додає cache status або edge/origin latency — ще краще.

З access logs я формую derived fields: verified_bot, bot_family, normalized_url, page_type, parameter_flags, date/hour, status_class. Не змінюю raw — derived layer можна перебудувати.

33.3. Crawl frequency

Кількість hits сама по собі не є метрикою “важливості”. Частота залежить від freshness, demand, signals, server capacity і багатьох внутрішніх рішень. Але в рамках одного сайту distribution корисний: які templates отримують 60% requests, які нові URL не бачать bot, які старі 404 запитуються місяцями.

Я рахую median/percentile recrawl interval за page type, а не average по домену. Homepage і мільйон архівних сторінок в одному average роблять число беззмістовним.

33.4. Status distribution

200 говорить, що fetch succeeded, але не гарантує indexation. 301/302 показують redirect overhead; 404/410 — removed URLs; 429 і 5xx можуть сигналізувати перевантаження або incident. Google рекомендує 500/503/429 як тимчасовий emergency спосіб різко знизити crawling, але це грубий інструмент і тривалі помилки можуть призвести до випадання URL.[2]

У log dashboard я дивлюся status mix саме для verified Googlebot і сегментую по page type. 2% 5xx на overall traffic і 20% 5xx на Googlebot requests до product pages — різні ситуації.

33.5. Response time і crawl capacity

Повільний сервер не є “ranking penalty” у простій формулі, але crawl capacity пов’язана зі здатністю host відповідати без перевантаження. Якщо latency росте й server errors з’являються, crawling може адаптуватися.

Я дивлюся p50/p95 latency bot requests, а не лише average. Один slow API endpoint у template може створювати довгий tail і робити рендер/response нестабільним.

33.6. Parameter waste

Логи добре показують, чи crawling витрачається на sort, tracking, session IDs, faceted combos, internal search. Я виділяю top query parameters by hits і unique URLs. Потім перевіряю, звідки вони discoverable: internal links, old backlinks, sitemaps, JS.

Не всі “зайві” requests треба блокувати robots.txt. Якщо параметричний URL уже має signals або дублює content, рішення може вимагати canonical, internal-link cleanup, redirects або removal. Robots просто не дає crawl content і може залишити URL відомим.

33.7. Discovery vs refresh

Умовно корисно розділяти first-seen URL і recrawl existing URL. Якщо launch має 50 000 new products, але за три дні Googlebot вперше побачив 5 000, проблема ближче до discovery/crawl demand. Якщо всі побачені, але не indexed — інший шар.

Для цього зберігаю `first_googlebot_hit`, `last_googlebot_hit`, `hit_count`. При міграції додаю old/new URL mapping і дивлюся, як швидко bot проходить redirect graph.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ. Я завантажую CSV verified Googlebot requests. Колонки: timestamp, url, status, response_ms, bytes, page_type. Побудуй: hits і unique URLs за page_type, status distribution, p50/p95 response time, top 404/5xx URLs, top URL parameters, first_seen/last_seen. Не роби висновків про ranking; відділяй факт HTTP request від гіпотези про crawl priority.

АЛЬТЕРНАТИВА БЕЗ ПРОГРАМУВАННЯ. Screaming Frog Log File Analyser або аналог може імпортувати access logs, відфільтрувати Googlebot і показати hits/status/URLs. Перед довірою до User-Agent перевірте bot verification та формат вашого CDN/origin log.

QR: компактна виконувана версія коду.

КОД 33.1. Базова агрегація verified Googlebot logs.

```
# Аналіз уже верифікованого Googlebot CSV.
# Колонки: timestamp,url,status,response_ms,bytes,page_type
import pandas as pd

df = pd.read_csv("googlebot_verified.csv", parse_dates=["timestamp"])

# Базове зведення за типом сторінки.
summary = (df.groupby("page_type")
             .agg(hits=("url","size"),
                  unique_urls=("url","nunique"),
                  p50_ms=("response_ms","median"),
                  bytes=("bytes","sum"))
             .sort_values("hits", ascending=False))

# Частка статусів усередині page_type.
status = (df.assign(status_class=(df.status//100).astype(str)+"xx")
           .groupby(["page_type","status_class"])
           .size().rename("hits").reset_index())
status["share"] = status["hits"] / status.groupby("page_type")
["hits"].transform("sum")

print(summary.head(30))
print(status.sort_values(["page_type","share"],
ascending=[True,False]).head(100))
```

33.8. Migration monitoring

Після migration logs дають фактичний recrawl old URLs, переходи через redirects, requests до new URLs і errors. Я будую cohorts: top traffic old URLs, top backlinks, categories, deep pages.

Не потрібно чекати два тижні, щоб дізнатися, що Googlebot отримує redirect loop на 15% old inventory. Це видно в перші години.

33.9. Що logs не показують

Логи не показують, чи URL indexed, rank, citation у AI, user intent або “crawl budget score”. Request — лише request. Тому logs треба join із crawl inventory, Search Console і status/canonical data.

Найгірша аналітика — перетворити crawl frequency на причинний ranking factor: “Googlebot заходить частіше, отже треба змусити його заходити частіше”. Частота може бути наслідком важливості/freshness, а не причиною rankings.

33.10. Зберігання й sampling

Логи великого сайту можуть бути терабайтами. Я зберігаю raw у cheap storage, а для аналізу — partitioned table з потрібними fields. Для recurring dashboard достатні daily aggregates; raw потрібен для incident drill-down.

Privacy/security важливі: access logs можуть містити user IPs, query parameters, tokens. SEO не повинен отримувати більше персональних даних, ніж потрібно для bot analysis. Фільтруйте і мінімізуйте.

Таблиця 33.1. Питання, на які добре відповідають logs.

Питання	Метрика	Чого не стверджувати
Що бот реально сканує?	hits / unique URLs	це не indexation
Де errors?	status mix	404 не завжди SEO-проблема
Що повільне?	p50/p95 response_ms	latency не дає точний ranking weight
Чи бачить launch?	first_seen cohort	crawl ≠ index
Чи багато параметрів?	hits by parameter	robots.txt не є універсальним fix
Як іде migration?	old→new request cohorts	traffic recovery має інші фактори

Джерела до Глави 33

1. Google Search Central. Googlebot. https://developers.google.com/search/docs/crawling-indexing/googlebot — Верифікація через reverse DNS/IP ranges.

2. Google Search Central. Crawl budget management. https://developers.google.com/search/docs/crawling-indexing/large-site-managing-crawl-budget — Crawl capacity/demand і server responses.

3. Google Search Central. HTTP network errors. https://developers.google.com/search/docs/crawling-indexing/http-network-errors — HTTP status behavior.

4. Google Search Central. Debug traffic drops. https://developers.google.com/search/docs/monitor-debug/debugging-search-traffic-drops — Відокремлення technical issues від інших причин.

ЧАСТИНА VII / ВИМІРЮВАННЯ

ГЛАВА 34

GA4, конверсії та атрибуція органічного трафіку

SEO без бізнес-метрики легко оптимізувати до красивих показів. GA4 допомагає зв’язати сесію й подію з каналом, але attribution — модель, а не фізична правда про те, “хто створив продаж”.

Стан даних і джерел: 14 вересня 2026 року

Search Console закінчується приблизно там, де користувач переходить на сайт. Бізнес тільки починається. Для lead gen мене цікавить qualified lead, для ecommerce — revenue/margin, для affiliate — outbound click, registration, FTD або інша доступна postback-подія.

GA4 корисний як behavioral layer, але він залежить від consent, client-side execution, ad blockers, cross-domain setup, event implementation і attribution settings. Тому “GA4 показує 812 organic sessions” — це measurement result за конкретною конфігурацією, а не абсолютна кількість людей із пошуку.

Рис. 34.1. Search Console і GA4 описують різні частини одного шляху користувача.

34.1. Session source/medium vs first user

GA4 має різні scopes traffic-source dimensions. `First user` описує джерело першого залучення користувача, `Session` — джерело конкретної сесії. Google пояснює, що user- і session-scoped dimensions використовують paid and organic last click, тоді як event-scoped attribution може використовувати вибрану attribution model.[1]

Це означає, що “organic users” і “organic sessions” не відповідають на одне питання. Для SEO performance я найчастіше починаю з Session source/medium, а для acquisition cohort — First user.

34.2. Default channel group — зручність, не закон природи

Organic Search у GA4 — rule-based channel grouping. Він корисний для dashboard, але source/medium треба зберігати. Новий AI referral може спочатку потрапити в Referral, доки classification rules не відповідають вашій аналітичній потребі.

OpenAI прямо зазначає, що ChatGPT search referrals додають `utm_source=chatgpt.com`, тож їх можна окремо відстежувати в

analytics.[2] Я не змішую такий трафік з Organic Search автоматично; створюю власний reporting dimension “AI referral” поверх raw source/medium.

34.3. Attribution — це розподіл credit

У GA4 доступні data-driven attribution, paid and organic last click та Google paid channels last click.[3] Data-driven використовує property data й оцінює внесок touchpoints; last click віддає credit останній відповідній взаємодії.

Якщо користувач побачив бренд в AI answer, через день зробив branded Google search і купив, стандартна web analytics може не мати observable touchpoint “AI mention”. Тому ніяка attribution model не може розподілити credit події, яку ви не виміряли.

34.4. Не порівнюйте GSC clicks і GA sessions 1:1

Google сам документує відмінності Search Console і Analytics: різна точка вимірювання, canonical URL attribution, tracking, timezone та інші правила.[4] Click може статися, але GA не запуститься; session може мати кілька pageviews; користувач може повернутися direct.

Я використовую ratio GSC clicks → GA organic sessions як monitoring metric, а не “правильний коефіцієнт”. Якщо ratio різко змінюється після consent banner release, це сигнал tracking change, а не обов’язково SEO.

34.5. Події мають бути бізнесовими

`scroll_90` і `time_on_page` можуть допомагати діагностиці, але KPI SEO повинні наближатися до грошей. Для ecommerce — purchase/revenue/margin. Для SaaS — signup, activated account, paid conversion. Для affiliate — outbound click плюс, якщо є, server-to-server postback із реєстрацією/депозитом.

Я документую event contract: назва, trigger, parameters, deduplication, source of truth. Якщо “lead” спрацьовує і при submit, і на thank-you page, конверсії подвояться незалежно від SEO.

34.6. Affiliate: outbound click — ще не revenue

У affiliate часто browser analytics бачить click to operator, але не бачить downstream registration/FTD через інший домен. Тоді потрібен postback або партнерський report, який можна join за click ID, sub ID, date/geo/landing у межах дозволеного tracking.

Без цього SEO-команда оптимізує CTR до оператора, а не якість трафіку. Два landing pages можуть мати однакові outbound clicks і втричі різний FTD rate.

34.7. AI referrals

ChatGPT, Perplexity, Copilot та інші можуть віддавати referral traffic. Це невеликий, але вимірюваний шматок AI visibility. Я створюю список known AI referrers і щотижня переглядаю new referral domains.

Не називаю AI referrals “повним AI traffic”. No-click visibility, branded follow-up search і agent action можуть не мати прямого referral. Це тільки observable clicks.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ. Я завантажу CSV з GA4 Traffic acquisition. Колонки: session_source, session_medium, sessions, key_events, revenue. Створи окремий custom channel “AI referral” для chatgpt.com, perplexity.ai, copilot.microsoft.com; Organic Search — для medium=organic; решта Other. Порахуй sessions, key events, key event rate, revenue. Не роби висновок, що це вся AI visibility: це лише прямі referrals.

АЛЬТЕРНАТИВА БЕЗ ПРОГРАМУВАННЯ. У GA4 Explorations створіть segment/filter за Session source для відомих AI-referrers і порівняйте Sessions, Key events, Total revenue з Organic Search. Збережіть raw source/medium, не покладайтеся лише на Default channel group.

QR: компактна виконувана версія коду.

КОД 34.1. Власна група AI referrals у GA4-export.

```
# Аналіз CSV з GA4 Traffic acquisition export.
# Колонки: session_source, session_medium, sessions, key_events, revenue
import pandas as pd

df = pd.read_csv("ga4_acquisition.csv")

# Власне правило для AI-referral джерел.
ai_sources = {"chatgpt.com", "perplexity.ai", "copilot.microsoft.com"}
df["channel_custom"] = "Other"
df.loc[df.session_medium.eq("organic"), "channel_custom"] = "Organic Search"
df.loc[df.session_source.isin(ai_sources), "channel_custom"] = "AI referral"

out = (df.groupby("channel_custom", as_index=False)
         .agg(sessions=("sessions","sum"),
              key_events=("key_events","sum"),
              revenue=("revenue","sum")))
out["key_event_rate"] = out.key_events / out.sessions.replace(0, pd.NA)
print(out.sort_values("revenue", ascending=False))
```

34.8. Landing page economics

Для SEO я дивлюся не тільки channel aggregate, a landing page × page type × query cluster. Revenue per organic session і key-event rate показують різницю між informational traffic і money pages.

Низький conversion rate informational article не означає, що article непотрібний: він може створювати assisted conversion або brand demand. Але такі claims треба доводити path/cohort analysis, а не storytelling.

34.9. Server-side і consent

Чим суворіші privacy restrictions, тим більше browser analytics має gaps. Server-side measurement може підвищити reliability окремих business events, але не є способом “обійти згоду”. Legal/compliance layer має бути частиною design.

Я завжди зберігаю business source of truth окремо: CRM, orders, partner postback. Analytics — модель поведінки, а бухгалтерія/CRM — інший контур.

34.10. Атрибуція не вирішує причинність

Навіть data-driven attribution не доводить, що SEO touchpoint спричинив conversion у causal sense. Це allocation model. Щоб оцінити causal lift, потрібні експерименти або квазіекспериментальні дизайни.

Тому в reporting я розділяю “attributed revenue” і “incremental effect”. Перше можна рахувати регулярно, друге — значно складніше і не для кожної задачі.

Таблиця 34.1. Що міряти після organic click.

Рівень	Приклад	Джерело	Ризик інтерпретації
Сесія	sessions, engaged sessions	GA4	consent/tracking gaps
Мікроконверсія	signup, outbound click	GA4/events	implementation errors
Макроконверсія	purchase, qualified lead	GA4/CRM	attribution
Affiliate outcome	registration, FTD	postback/network	cross-domain matching
Цінність	revenue, margin, LTV	backend/BI	затримка й refund

Джерела до Глави 34

1. Google Analytics Help. Traffic-source scopes. https://support.google.com/analytics/answer/11080067?hl=en — User/session/event scopes.

2. OpenAI Help Center. Publishers and Developers FAQ. https://help.openai.com/uk-ua/articles/12627856-publishers-and-developers-faq — ChatGPT referrals і utm_source=chatgpt.com.

3. Google Analytics Help. Attribution. https://support.google.com/analytics/answer/10596866 — Три доступні attribution models.

4. Google Search Central. Search Console and Analytics. https://developers.google.com/search/docs/monitor-debug/google-analytics-search-console — Чому дані не збігаються 1:1.

5. Google Analytics Help. Campaigns and traffic sources. https://support.google.com/analytics/answer/11242841?hl=uk — Traffic acquisition basics.

ЧАСТИНА VII / ВИМІРЮВАННЯ

ГЛАВА 35

SEO-експерименти: причинність, контроль і статистичні пастки

SEO живе в середовищі, де одночасно змінюються сайт, конкуренти, попит і Google. Саме тому “ми змінили X, через два тижні трафік виріс” — слабкий доказ причинності.

Стан даних і джерел: 14 вересня 2026 року

У SEO мало ідеальних рандомізованих експериментів, але це не виправдання для до/після storytelling. Можна покращити доказовість через control groups, matched pages, staggered rollout, difference-in-differences, synthetic controls і Bayesian time-series models.

Мета експерименту — не отримати green arrow. Мета — зменшити uncertainty настільки, щоб рішення про rollout було кращим. Негативний або нульовий результат теж економить бюджет.

Рис. 35.1. Чим сильніший контроль контрфактуального сценарію, тим сильніший причинний висновок.

35.1. Гіпотеза до даних

Я записую hypothesis до запуску: change, target metric, expected direction, unit, period, exclusion rules, stop condition. Це зменшує HARKing — вигадування пояснення після того, як графік уже побачили.

Наприклад: “Додавання contextual links із hub pages на orphan-ish product pages збільшить impressions/eligible URL за 28 днів порівняно з matched control, без росту 404/redirect crawl”. Це значно краще за “тестуємо internal links”.

35.2. Одиниця експерименту

Page, template, category, domain, market — різні units. Якщо ви змінюєте global navigation, page-level control практично неможливий, бо treatment впливає на весь graph. Якщо змінюєте title 500 products, можна випадково розподілити comparable URLs.

Потрібно враховувати interference: treatment однієї сторінки може впливати на іншу через internal links або cannibalization. Класичний A/B assumption незалежності часто порушується.

35.3. Randomization

Randomized controlled experiments вважаються gold standard причинності, бо random assignment балансує відомі й невідомі confounders у середньому. Microsoft описує controlled experiments як спосіб науково встановлювати причинність у web products.[1]

У SEO randomization можлива не завжди, але для великих наборів однотипних сторінок — іноді так. Головне не рандомізувати URL, які фундаментально різні за demand, age або category, якщо sample малий.

35.4. Matched controls

Коли randomization неможлива, я підбираю control за pre-period metrics: impressions, clicks, average position, page type, age, country, query class. Потім перевіряю parallel trends до treatment.

Control не повинен “виглядати схожим” лише за URL. Якщо treatment group сезонна, а control — ні, difference після запуску нічого не доведе.

35.5. Difference-in-differences

Проста логіка: дивимося, наскільки treatment змінився до/після, і віднімаємо зміну control. Якщо treatment +20%, control +15%, estimated incremental change близький до +5%, а не +20%.

Це все одно залежить від assumptions, зокрема parallel trends. Але вже сильніше, ніж назвати весь +20% ефектом SEO-зміни.

35.6. CausalImpact / synthetic control

Google Research опублікував підхід Bayesian structural time series для оцінки causal impact intervention, коли randomized experiment

неможливий.[2] Модель будує counterfactual time series із control predictors і порівнює observed post-period із очікуваним.

Це не магічна кнопка. Якщо control variables самі зазнали treatment, якщо structural break збігається з Google update, якщо pre-period короткий — модель може бути переконливо неправильною.

35.7. Seasonality і Google updates

Перед запуском фіксую Search Status Dashboard, known updates, holidays, promotions, product launches. Якщо core update почався посеред тесту, я не “коригую” data в Excel. Або продовжую test до стабілізації, або визнаю confounding.

Google рекомендує чекати щонайменше повний тиждень після завершення core update перед аналізом у Search Console.[3] Це корисна дисципліна і для експериментів: не робити причинний висновок під час rollout.

35.8. Sample size і minimum detectable effect

Малий sample може не відрізнити +5% від шуму. Я визначаю minimum effect, який взагалі має бізнес-сенс, і будую sample/period навколо нього. Якщо для +3% CTR потрібні мільйони impressions, а сайт має 20k, не треба обіцяти точність, якої дані не дадуть.

В SEO часто краще агрегувати homogeneous page cohort, ніж тестувати одну сторінку. Але агрегація не повинна змішувати intent і seasonality.

35.9. Multiple testing

Якщо протестувати 100 змін із $p<0.05$ і вибрати лише “переможців”, частина буде випадковою. SEO-команди рідко коригують multiple comparisons, бо багато тестів неформальні. Мінімум — вести registry всіх тестів, включно з нульовими.

Я не використовую statistical significance як binary magic. Дивлюся effect size, confidence interval/credible interval, business value, consistency і mechanism plausibility.

35.10. Primary і guardrail metrics

Primary metric має відповідати hypothesis: indexed rate, clicks, conversions, revenue. Guardrails — те, що не повинно погіршитися: 404 rate, crawl errors, conversion rate, page speed, returns.

Наприклад, SEO title test може підняти CTR, але знизити conversion через overpromise. CTR winner не обов'язково business winner.

БЕЗ КОДУ: ПРОМПТ ДЛЯ ШІ. У мене CSV SEO-тесту з колонками group (treatment/control), period (pre/post), metric. Порахуй mean для 4 комірок і difference-in-differences: (treatment post-pre) - (control post-pre). Потім поясни assumptions: parallel trends, independence, відсутність одночасних змін. Не називай результат причинним, якщо assumptions не перевірені.

АЛЬТЕРНАТИВА БЕЗ ПРОГРАМУВАННЯ. У Google Sheets зробіть зведену таблицю group × period із середньою metric. Формула DID: `(T_post-T_pre)-(C_post-C_pre)`. Перед висновком побудуйте pre-period time series для treatment/control і перевірте, чи тренди були схожі.

QR: компактна виконувана версія коду.

КОД 35.1. Базовий difference-in-differences.

```
# Простий difference-in-differences на агрегованих даних.
# CSV: group (treatment/control), period (pre/post), metric
import pandas as pd

df = pd.read_csv("seo_test.csv")
p = df.groupby(["group","period"])["metric"].mean().unstack()

treat_change = p.loc["treatment","post"] - p.loc["treatment","pre"]
control_change = p.loc["control","post"] - p.loc["control","pre"]
did = treat_change - control_change

print(p)
print("Treatment change:", treat_change)
print("Control change:", control_change)
print("Difference-in-differences:", did)
```

35.11. Rollout після тесту

Переможець на 10% pages не гарантує такий самий ефект на 100%. Інші categories, demand distributions і link graph можуть змінити результат. Я rollout роблю поетапно й залишаю holdout, якщо business дозволяє.

Після rollout моніторю durability: ефект через 7, 28, 90 днів. SEO system може переоцінити сторінки не одразу.

35.12. Культура негативних результатів

Якщо команда публікує тільки success cases, вона навчається неправильно. Я зберігаю failed/neutral tests із hypothesis і context. Це створює internal evidence base і знижує повторення тих самих ідей.

Найцінніший результат іноді: “Ми думали, що X дає +15%, але контроль показав 0–2%; більше не витрачаємо на це sprint”.

Таблиця 35.1. Мінімальний паспорт SEO-експерименту.

Поле	Що записати
Гіпотеза	change → mechanism → metric → direction
Unit	URL/template/category/market

Поле	Що записати
Treatment	точний diff
Control	як підібраний/рандомізований
Pre-period	тривалість і parallel trend
Post-period	тривалість, lag
Primary metric	одна основна
Guardrails	що не має погіршитися
Confounders	updates, seasonality, releases
Result	effect size + uncertainty
Decision	rollout / repeat / reject

Джерела до Глави 35

1. Microsoft Research. Online Experimentation at Microsoft. https://www.microsoft.com/en-us/research/publication/online-experimentation-at-microsoft/ — Controlled experiments і причинність.

2. Google Research. Inferring causal impact using Bayesian structural time-series models. https://research.google/pubs/inferring-causal-impact-using-bayesian-structural-time-series-models/ — Квазіекспериментальна оцінка intervention.

3. Google Search Central. Core updates. https://developers.google.com/search/docs/appearance/core-updates — Час аналізу після rollout.

4. Microsoft Research. Benefits of Controlled Experimentation at Scale. https://www.microsoft.com/en-us/research/publication/the-benefits-of-controlled-experimentation-at-scale/ — Практична культура експериментів.

ЧАСТИНА VII / ВИМІРЮВАННЯ

ГЛАВА 36

Прогнозування: як оцінювати трафік, результат і ресурс без магії

SEO forecast — не обіцянка майбутнього. Це прозора модель припущень: попит × досяжна видимість × CTR × конверсія × цінність, плюс час, обмеження і діапазон невизначеності.

Стан даних і джерел: 14 вересня 2026 року

Найгірший SEO forecast — число з двома знаками після коми на 12 місяців: “очікуємо 1 284 732 clicks”. Воно виглядає точно, але приховує невідомі: rankings, SERP changes, competitor actions, AI answers, seasonality, indexation, release delays.

Я прогнозую не одне число, а систему сценаріїв. Forecast потрібен для порівняння рішень і capacity planning, а не для імітації контролю над Google.

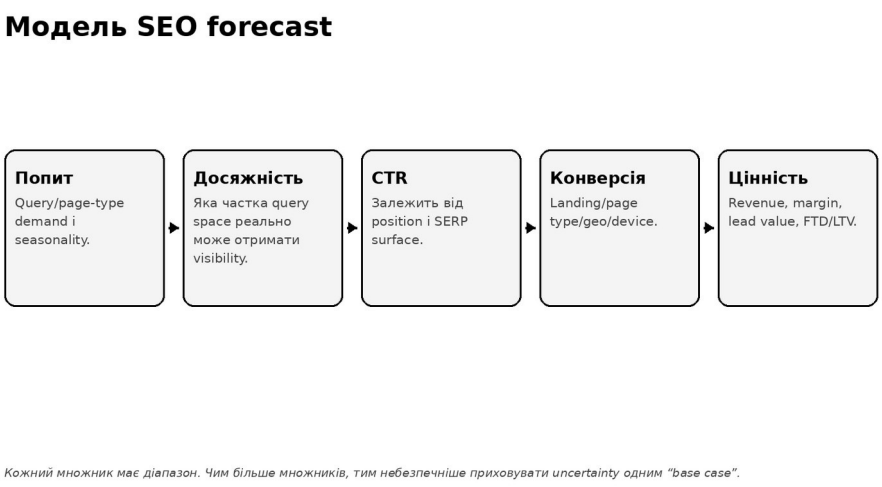


Рис. 36.1. Forecast як ланцюг припущень, а не як одна магічна метрика.

36.1. Прогнозуйте unit economics

Для content cluster я можу прогнозувати monthly clicks і conversions. Для technical fix — recovered indexable inventory або avoided loss. Для migration — risk range. Для link campaign — не "DR +10", а expected contribution to target pages, але з великою uncertainty.

Модель повинна відповідати типу рішення. Не треба змушувати кожну SEO-задачу давати "traffic forecast", якщо цінність — reliability або risk reduction.

36.2. Demand baseline

Search volume із third-party tool — оцінка, не census. Я калібрую його власним GSC там, де сайт уже має impressions. Для нового market використовую ranges і сезонність. Google Trends допомагає зрозуміти direction, але не дає absolute volume.

Для programmatic set я не сумую volumes тисяч keywords без deduplication intent. Одна сторінка може ранжуватися за сотнями синонімів; сума keyword volumes завищує addressable demand.

36.3. Ranking/visibility scenarios

Замість “вийдемо в top-3” я задаю distributions: low — page 2/top10 small share; base — частина clusters top5-10; high — сильний top3 share. Для кожної групи використовую власний observed CTR, якщо є.

Generic CTR curve з дослідження — старт, не істина. Brand/non-brand, device, rich results, AI Overviews і query intent різко змінюють CTR. Найкраща CTR model — власний historical GSC за схожими query classes.

36.4. AI/zero-click adjustment

У 2026 informational visibility може мати нижчий external CTR через AI Overviews та інші SERP answers. Я не вставляю один “AI penalty = 40%” у всю модель. Сегментую queries за surface/intent і використовую observed data або scenario range.

Може зрости search usage, але впасти click yield. Тому demand growth і traffic growth — різні параметри forecast.

36.5. Indexation ramp

Для 50 000 new URLs не припускаю, що всі indexed у день launch. Модель має ramp: generated → discovered → crawled → indexed → impressions. Коефіцієнти беру з pilot або historical template.

Якщо pilot показав 65% indexed after 28d і 80% after 90d, forecast повинен відображати це. Інакше revenue “з першого місяця” просто математично неможливий.

36.6. Conversion і value

SEO traffic неоднорідний. Money pages і informational articles мають різний conversion rate. Я прогнозую на рівні page type / intent. Для ecommerce краще margin, а не gross revenue, якщо рішення стосується бюджету.

Для lead gen використовую qualified rate і close rate; для affiliate — EPC або registration→FTD, якщо дані є. Якщо downstream data немає, прямо показую, де модель обривається.

Таблиця 36.1. Приклад сценарної моделі.

Параметр	Low	Base	High
Eligible URLs	8 000	10 000	10 000
Index rate 90d	55%	75%	85%
Impressions / indexed URL	120	180	240
CTR	2.0%	3.0%	4.0%
Conversion rate	1.5%	2.0%	2.5%
Value / conversion	€35	€35	€35

36.7. Forecast resource, not only outcome

Проєкт може мати високий traffic upside і бути поганим рішенням через engineering/content cost. Я додаю required effort: developer days, editorial hours, data engineering, links, design, maintenance.

Це дозволяє порівняти “10k pages programmatic” з “fix canonical bug” не за excitement, а за expected value / cost / confidence / time-to-impact.

36.8. Time to impact

SEO changes мають лаг. Technical fix може бути помітний після recrawl; content — після index/re-evaluation; authority — ще менш детерміновано. Forecast із cumulative curve реалістичніший за рівну лінію.

Я показую month 1–3 ramp, 4–6 scale, 7–12 steady/decay, якщо це відповідає механіці. Якщо даних немає — range ширший, а не “середній лаг 47 днів”.

36.9. Confidence score

Confidence — не відчуття 8/10. Я розкладаю: demand evidence, implementation certainty, indexability, historical analog, conversion data, competitive gap. Потім пояснюю, чому scenario range такий широкий.

Forecast із low confidence може бути корисним, якщо upside великий і тест дешевий. Він просто не має продаватися CFO як committed revenue.

36.10. Backtesting

Кожний квартал я беру старі forecasts і порівнюю з фактом: де помилилися demand, index rate, CTR, conversion, timeline. Це найкращий спосіб покращувати модель.

Якщо forecasts ніколи не backtest-яться, команда не прогнозує — вона пише sales documents.

36.11. Не перетворюйте forecast на KPI

Коли forecast стає quota, команда починає підганяти assumptions або оптимізувати короткострокові vanity metrics. KPI — performance system, forecast — decision model. Вони пов’язані, але не тотожні.

Я фіксую assumptions version і змінюю forecast тільки коли з’явилися нові факти, а не щоб “повернути план у green”.

36.12. Проста формула

Для cohort сторінок базова модель може бути: `Eligible URLs × Index rate × Impressions per indexed URL × CTR × Conversion rate × Value`. Вона груба, зате прозора. Складніша модель має сенс тільки коли покращує рішення.

Кожний множник треба мати можливість замінити actual data після launch. Тоді forecast поступово перетворюється на operating model.

МІФ. Точніший forecast — це не більше десяткових знаків. Точніший forecast — це краще визначені припущення, сегментація, діапазони й регулярний backtesting.

Джерела до Глави 36

1. Google Search Central. Debugging traffic drops. https://developers.google.com/search/docs/monitor-debug/debugging-search-traffic-drops — Seasonality, updates, technical issues і demand.

2. Google Search Central. Search Console Performance data. https://developers.google.com/search/blog/2022/10/performance-data-deep-dive — Власні impressions/clicks як calibration data.

3. Search Console Help. Generative AI performance report. https://support.google.com/webmasters/answer/16984139?hl=en — AI exposure як окремий measurement layer.

4. Google Analytics Help. Traffic source dimensions. https://support.google.com/analytics/answer/15612152?hl=uk — Conversion/value segmentation за джерелом і сесією.

ЧАСТИНА VIII / РИЗИК

ГЛАВА 37

Системи боротьби зі спамом, апдейти й ручні санкції

Падіння після апдейту, автоматична спам-демоція і ручна дія можуть виглядати однаково в графіку. Якщо їх змішати, команда починає лікувати не ту систему.

Стан даних і джерел: 14 вересня 2026 року

Рис. 37.1. Системи боротьби зі спамом, апдейти й ручні санкції

37.1. Три різні класи подій

Я розділяю три речі: широку переоцінку ранжування, автоматичне застосування систем боротьби зі спамом і ручну дію. Core update, за поясненням Google, не є санкцією проти конкретного сайту: це широка зміна того, як системи оцінюють контент і конкурентний

набір.[1] Manual action, навпаки, виникає після рішення людини-перевіряльника і відображається в Search Console.[2]

Це не семантика. Від класифікації залежить план. При ручній дії є конкретний report і процедура reconsideration. При технічному інциденті потрібен rollback або fix. При broad re-ranking може не бути одного дефекту, який можна “видалити”.

37.2. Спам-політики — це не список “ranking factors”

Spam policies описують поведінку, яка може призвести до нижчого ранжування або виключення з результатів. Google прямо включає cloaking, doorway abuse, scaled content abuse, scraping, expired domain abuse, link spam і site reputation abuse.[3] У 2026 правила також прямо поширюються на спроби маніпулювати generative AI responses у Google Search.

Я не перетворюю цей документ на зворотну інструкцію “що саме ловить Google”. Він не розкриває thresholds, weights чи detection logic. Політика показує boundary condition: яку мету й поведінку Google вважає маніпулятивною.

37.3. Core update: не шукайте чарівну кнопку

Актуальна документація Google рекомендує спочатку дочекатися завершення rollout, а потім порівнювати коректні періоди. Для core update Google радить зачекати щонайменше повний тиждень після завершення, перш ніж робити висновки в Search Console.[1] Малий рух, наприклад 2→4, не потребує радикальних змін. Великий і стійкий рух типу 4→29 — привід для глибшої оцінки.

Найгірша реакція — масово переписати titles, видалити блоки, скоротити тексти або змінити internal links лише тому, що хтось у X/Telegram назвав один елемент “тригером апдейту”. Це руйнує baseline й унеможливлює causal reading наступних даних.

37.4. Manual action: у вас є observable evidence

Search Console Manual Actions report показує, чи застосована ручна дія, її тип і охоплення. Google повідомляє про неї також через message center. Після виправлення можна подати Request review; якщо Google вважає порушення усуненим, manual action відкликається.[2]

Тут я працюю як з incident response: зберігаю snapshot сайту й доказів до змін, описую scope, видаляю першопричину, перевіряю весь клас схожих URL і тільки після цього подаю reconsideration. “Ми видалили одну сторінку” не допоможе, якщо проблема системна.

37.5. Site reputation abuse у 2026

Site reputation abuse стосується third-party content, опублікованого переважно для використання ranking signals host-сайту.[3] У серпні 2026 Google окремо оновив enforcement у Європейській економічній зоні: з 30 серпня manual actions за цією політикою мають інший ефект для пошуку в EEA, ніж поза нею.[4]

Для міжнародних publishers це означає, що policy analysis не можна робити без GEO. Один і той самий host/content relationship може мати різні наслідки у різних search experiences. Це ще один аргумент зберігати country dimension у діагностиці.

37.6. Spam update ≠ core update

Google розрізняє ranking systems і updates до цих систем. Історично Search Central окремо документує core updates і spam systems. Тому “апдейт був того дня” — тільки temporal correlation, а не доказ конкретного механізму.[5]

Я прив’язую timeline до Google Search Status / ranking update history, але далі дивлюся pattern: які templates, queries, countries і page types втратили. Якщо падає тільки один комерційний каталог, а editorial стабільний, sitewide історія вже менш імовірна.

37.7. Scaled content abuse і AI

Google прямо формулює scaled content abuse через мету і цінність, а не через інструмент створення. Генеративний AI наведений як один із можливих способів створити багато сторінок без доданої користі, але сама автоматизація не дорівнює порушенню.[3]

Це важливо для programmatic SEO: питання не “AI чи людина”, а “чому ця URL повинна існувати, яку унікальну функцію/дані/рішення вона дає, і чи створюємо ми тисячі сторінок головно заради покриття запитів”.

37.8. Коли видалення контенту виправдане

У core update guidance Google називає видалення контенту last resort: якщо контент неможливо врятувати і він створювався насамперед для search engines, тоді deletion може бути логічним.[1] Це не рекомендація “після апдейту видаліть 30% сайту”.

Я спочатку класифікую inventory: valuable, salvageable, redundant, obsolete, abusive. Для valuable — improvement; для redundant — consolidation; для obsolete — update або retirement; для abusive — removal/noindex depending on purpose. Масовий prune без класифікації — ще одна форма панічного SEO.

37.9. Моя таблиця incident classification

Перед будь-яким recovery sprint я заповнюю мінімальну матрицю: дата початку, тип сигналу, Search Console messages, security issues, change log, update timeline, affected countries, devices, page types, query classes, index coverage, crawl/log anomalies.

Це займає години, а не тижні, але різко знижує шанс почати дорогий контентний rewrite, коли причина — noindex release, неправильний canonical, зміна SERP або tracking bug.

37.10. Що вважати відновленням

Recovery — не “повернули стару позицію”. Конкурентний набір і сам SERP змінюються. Я задаю observable target: restored eligible inventory, recovered non-brand clicks, stabilized conversion value, removal of manual action, reduction of error class.

Якщо після policy fix manual action знята, але traffic не повернувся, це не означає, що reconsideration “не спрацював”. Зняття санкції повертає право конкурувати, а не гарантує колишній rank.

Таблиця. Практична матриця для цієї глави.

Сигнал	Найімовірніший клас	Перша перевірка	Типова помилка
Manual action у GSC	Ручна санкція	Scope + policy + приклади	Чекати “наступного апдейту”
Різке падіння після deploy	Технічний інцидент	robots/noindex/ canonical/redirect/ logs	Переписувати контент
Sitewide re-ranking у rollout window	Core/ranking update	порівняння cohorts і конкурентів	Шукати один “штрафний фактор”
Падіння лише spam-like templates	Автоматична policy/spam система або quality re-ranking	template + intent + policy mapping	вважати весь домен “під фільтром”

Джерела до Глави 37

1. Google Search Central. Core updates.
https://developers.google.com/search/docs/appearance/core-updates

2. Search Console Help. Manual actions report.
https://support.google.com/webmasters/answer/9044175

3. Google Search Central. Spam policies.
https://developers.google.com/search/docs/essentials/spam-policies

4. Google Search Central Blog. Update to the Site Reputation Policy, 2026-08-28.
https://developers.google.com/search/blog/2026/08/update-site-reputation-policy

5. Google Search Central. Latest documentation updates.
https://developers.google.com/search/updates

ЧАСТИНА VIII / РИЗИК

ГЛАВА 38

Дроп-домени, PBN, платні посилання, parasite SEO та редиректи

Ринок SEO не складається тільки з “безпечних рекомендацій”. Але професійна розмова про ризикову тактику повинна мати чотири колонки: механіка, очікувана вигода, політика і вартість відмови.

Стан даних і джерел: 14 вересня 2026 року

Рис. 38.1. Дроп-домени, PBN, платні посилання, parasite SEO та редиректи

38.1. Ризик — це не моральна категорія

Для бізнесу grey/black tactics — це портфельний ризик. Я дивлюся на expected upside, probability of loss, blast radius, recoverability і

replacement cost. Один PBN link на disposable affiliate site і site reputation deal на головному бренді — різні ризикові профілі, навіть якщо обидва суперечать policy.

Тому “працює/не працює” недостатньо. Тактика може працювати шість місяців і бути поганою інвестицією, якщо очікувана вартість втрати домену перевищує прибуток.

38.2. Expired domain: що саме є порушенням

Google визначає expired domain abuse як купівлю й перепрофілювання expired domain головно для маніпуляції rankings контентом із малою або нульовою цінністю. У документації прямо наведено приклад casino content на домені колишньої школи.[1]

Це не означає, що кожна купівля expired domain заборонена. Критичні ознаки політики — primary purpose і value. Але з operational perspective старий link graph не є страховкою: історія тематики, spam anchors, redirects, index state і ownership transitions треба перевіряти окремо.

38.3. 301 не є “переливом DR”

Google рекомендує permanent server-side redirects для реальних site moves і consolidation. При коректному переїзді redirects допомагають передавати signals на нові URL.[2] Це не тотожне тезі, що будь-який старий домен можна направити на money site і механічно “передати authority”.

Я оцінюю mapping relevance. Старий документ про topic A → новий еквівалент topic A має зрозумілу користувацьку й міграційну логіку. Сотні різнорідних expired URLs → homepage казино — вже інший клас поведінки і значно слабший semantic mapping.

38.4. Redirect chains і технічна ціна

Google радить вести redirect одразу до final destination і уникати chains; у site move guidance зазначено, що Googlebot може пройти до

10 hops, але практична рекомендація — тримати chain низьким, ideally не більше 3 і менше 5.[2]

Для SEO-портфеля це означає: якщо домени мігрували кілька разів, не залишайте legacy chain як “історію”. Flatten redirects, update internal links і перевіряйте final status. Кожний hop додає latency й точку відмови.

38.5. PBN: контроль в обмін на концентрований ризик

PBN приваблює контролем над placement, anchor, timing і destination. Але той самий контроль створює correlated footprint: ownership, hosting, templates, analytics IDs, link patterns, content quality, registration history. Я не описую способи приховування мережі — це гонка з системами detection, а не стійка SEO-модель.

Бізнесово важливіше інше: якщо 30% ranking dependency сидить на ресурсах, які можуть зникнути одночасно, це concentration risk. Його треба бачити в link inventory так само, як фіндиректор бачить залежність від одного постачальника.

38.6. Платні посилання

Google link spam policy вважає купівлю або продаж links для ranking manipulation порушенням; для рекламних/paid placements Google рекомендує `rel=sponsored` або інші атрибути, що не передають ranking credit.[1] Факт, що ринок купує links, не змінює policy status.

У практичному бюджеті я розділяю distribution/PR value і очікуваний SEO effect. Link, який дає qualified referral traffic, brand mention і discovery, має value навіть без “передачі ваги”. Це корисніше, ніж зводити кожний placement до DR/price.

38.7. Parasite SEO і site reputation abuse

Site reputation abuse — third-party content, розміщений переважно для використання established ranking signals host-сайту.[1] Google у 2026 уточнив enforcement у EEA.[3] Отже, “орендувати піддиректорію

сильного медіа” більше не можна оцінювати тільки через швидкість rank.

Я рахую dependency: чи маю контроль над URL, контентом, redirects, analytics, renewal terms, deletion risk. Parasite може бути швидким channel, але майже завжди має низьку asset ownership.

38.8. Doorways і multi-domain portfolios

Spam policies прямо включають multiple domains/pages із невеликими варіаціями, створені для покриття схожих queries і funnel до однієї destination, у doorway abuse.[1] Для affiliate portfolios це важлива boundary condition.

Multi-domain strategy сама по собі не doorway. Сайти можуть мати окремі brands, audiences, data, offers і editorial value. Але якщо 20 доменів відрізняються логотипом і H1, а ведуть в один funnel, політичний ризик значно вищий.

38.9. Thin affiliation

Affiliate page не стає корисною від кількості слів. Її moat — власне порівняння, data, screenshots, testing, terms normalization, availability, UX, filters, editorial judgment. Якщо сторінка лише перепаковує merchant copy, вона commodity і легко замінюється SERP/AI answer.

Для iGaming я особливо дивлюся на jurisdiction, T&C timestamp, wagering, min deposit, payment/restriction data і відповідність фактичній пропозиції. Помилка в commercial fact знищує довіру швидше, ніж слабкий keyword coverage.

38.10. Портфельний контроль ризику

Я веду risk register по доменах і тактиках: policy exposure, dependency, owner, replacement plan, maximum acceptable loss. Disposable experiment не повинен мати доступ до core analytics, brand accounts чи shared infrastructure без потреби.

Суть не в тому, щоб зробити risk tactics “безпечними”. Безпечними вони не стають. Суть — знати, який актив ви ставите під ризик і чи можете пережити його втрату.

Таблиця. Практична матриця для цієї глави.

Тактика	Upside	Головний ризик	Recoverability
Expired domain / redirect	швидший запуск, historical links	policy + irrelevance + toxic history	середня, якщо core site не залежить
PBN	контроль placement/anchor	correlated network loss	низька для links, вища для main site
Paid links	керований acquisition	link spam enforcement / wasted spend	середня
Parasite / host rental	швидка host visibility	asset не ваш + site reputation policy	низька
Multi-domain	SERP coverage + diversification	doorway footprint + ops complexity	залежить від автономності сайтів

Джерела до Глави 38

1. Google Search Central. Spam policies.
https://developers.google.com/search/docs/essentials/spam-policies

2. Google Search Central. Site moves with URL changes.
https://developers.google.com/search/docs/crawling-indexing/site-move-with-url-changes

3. Google Search Central Blog. Update to the Site Reputation Policy.
https://developers.google.com/search/blog/2026/08/update-site-reputation-policy

4. Google Search Central. Redirects and Google Search.
https://developers.google.com/search/docs/crawling-indexing/301-redirects

ЧАСТИНА VIII / РИЗИК

ГЛАВА 39

Відновлення: як діагностувати падіння й повертати систему в робочий стан

Recovery — це дисципліна діагностики. Якщо причина невідома, будь-яке “виправлення” є експериментом без контролю.

Стан даних і джерел: 14 вересня 2026 року

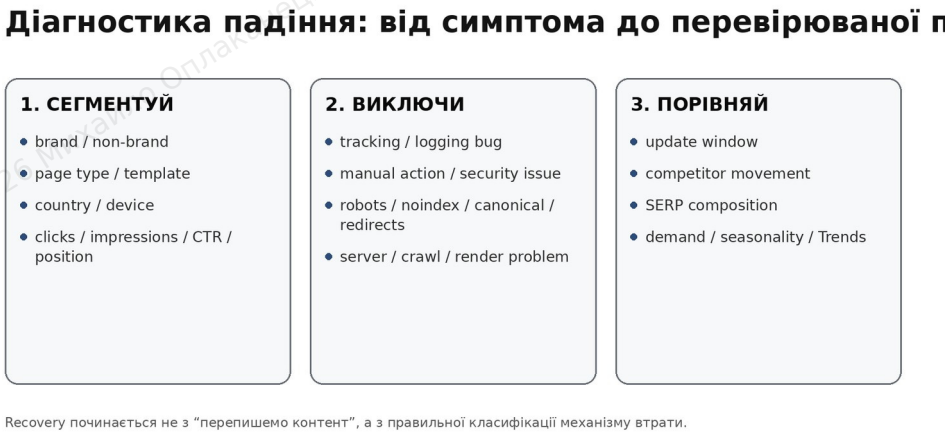


Рис. 39.1. Відновлення: як діагностувати падіння й повертати систему в робочий стан

39.1. Спочатку визначте форму падіння

Google у своїй інструкції радить починати з Performance report і дивитися, чи змінилися clicks, impressions і average position, а також порівнювати search types.[1] Я додаю ще page type, query class, country, device і brand/non-brand.

Раптовий cliff, повільний decay, сезонний спад і step-change після migration — різні форми. Графік не ставить діагноз, але відсікає частину причин.

39.2. Перша година incident response

Я перевіряю Search Console messages, Manual Actions, Security Issues, indexing anomalies, robots.txt, noindex, canonical, redirect rules, DNS/CDN status, server errors, deployment log і analytics release. Це cheap checks із великим expected value.

Якщо сайт учора повертав 200, а сьогодні 503 через WAF rule, немає сенсу аналізувати E-E-A-T. Спочатку повертаємо систему в робочий стан.

39.3. Друга година: локалізуйте blast radius

Втрата 40% total clicks може бути одним template, який давав половину трафіку. Або 5% падіння на кожному типі сторінок. Я будую матрицю page type × country × device × query class і дивлюся contribution to loss.

Contribution важливіший за percentage change. URL з -90%, який давав 20 clicks, не повинен відволікати від категорії з -12%, що втратила 80 000 clicks.

39.4. Алгоритмічна зміна чи попит

Google прямо радить перевіряти Trends і demand changes, а також timeline ranking updates.[1] Якщо impressions падають разом у всіх доменів ніші, це інша задача, ніж втрата share при стабільному попиті.

Я порівнюю own impressions із market proxies, branded demand і competitor visibility. Seasonality не можна “виправити SEO”. Її можна врахувати у forecast і capacity.

39.5. Малий vs великий ranking drop

Актуальний core update guide розрізняє малі й великі зміни позиції. Малий рух у top results не є приводом для drastic action; великий sitewide drop потребує глибшої оцінки.[2]

Це важливо психологічно. Команда, яка реагує на кожен -1.2 position, створює більше шуму, ніж покращень.

39.6. Міграції: recovery через mapping

При URL migration Google рекомендує server-side permanent redirects, оновлення internal links, canonicals, sitemaps і достатню server capacity. Small/medium migrations можуть оброблятися кілька тижнів; великим сайтам потрібно більше часу.[3]

Recovery тут вимірюється не “старий домен ще ранжується чи ні”, а coverage mapping: old URL → final URL, redirect correctness, canonical alignment, discovered/indexed new URLs, traffic transfer by directory.

39.7. Не робіть одночасно десять “покращень”

Після падіння бізнес тисне: перепишімо тексти, змінимо title, купимо links, видалимо 20k pages, оновимо design. Якщо зробити все, ви втратите ability to learn.

Я класифікую fixes: hard defect, policy remediation, evidence-backed improvement, speculative test. Hard defect можна виправляти одразу. Спекулятивні зміни — окремими cohorts.

39.8. Recovery log

У мене є журнал: hypothesis, evidence, change, owner, deploy date, affected scope, expected leading metric, expected lag, result. Це не бюрократія, а пам'ять системи.

Через три місяці без журналу команда пам'ятає тільки “ми щось робили з internal links”. З журналом можна backtest рішення й не повторювати невдалі fix-и.

39.9. Коли чекати

Google прямо попереджає, що після meaningful improvements частина ефектів може проявитися за кілька днів, а частина — через місяці; гарантії recovery немає.[2] Це не виправдання бездіяльності, а reminder про lag.

Чекати має сенс тільки після того, як leading signals підтверджують, що fix live: bot отримує 200, canonical правильний, URLs discovered, content changed. “Чекаємо апдейту” без перевірки implementation — не стратегія.

39.10. Вихід із recovery mode

Recovery закінчується, коли root cause контрольована або принаймні operational risk повернувся в норму. Далі сайт знову переходить у growth portfolio: experiments, content, authority, product work.

Якщо команда шість місяців живе в режимі “повернути старий трафік”, вона може пропустити новий query space. Іноді правильна мета — не повернути 2025, а побудувати кращу систему для 2026.

Таблиця. Практична матриця для цієї глави.

Pattern	Що перевіряти першим	Не робити одразу
Cliff за 1 день	deploy, robots/noindex, server, manual/security	масовий контентний rewrite
Плавний decay	demand, competition, content decay, SERP	панічний prune
Падіння після migration	mapping, redirects, canonicals, sitemap, logs	міняти URL ще раз
Тільки один template	render/data/template/ indexing	робити sitewide висновок
Impressions стабільні, clicks вниз	CTR/SERP/AI/features	звинувачувати indexation

Джерела до Глави 39

1. Google Search Central. Debugging Search traffic drops. https://developers.google.com/search/docs/monitor-debug/debugging-search-traffic-drops

2. Google Search Central. Core updates. https://developers.google.com/search/docs/appearance/core-updates

3. Google Search Central. Site moves. https://developers.google.com/search/docs/crawling-indexing/site-move-with-url-changes

4. Google Search Central. Changing hosting. https://developers.google.com/search/docs/crawling-indexing/site-move-no-url-changes

ЧАСТИНА IX / SEO-КОМАНДА У 2026

ГЛАВА 40

Дорожня карта, пріоритизація та бюджет: як керувати SEO як продуктом

SEO Lead не керує “списком рекомендацій”. Він керує портфелем ставок із різним upside, доказовістю, залежностями й часом до результату.

Стан даних і джерел: 14 вересня 2026 року

Рис. 40.1. Дорожня карта, пріоритизація та бюджет: як керувати SEO як продуктом

40.1. Audit не є roadmap

Audit описує стан системи. Roadmap описує, що команда робитиме, в якій послідовності й чому. 180 знайдених issues не створюють стратегію автоматично.

Я перетворюю findings на opportunities: recovered coverage, new query space, CTR gain, conversion gain, risk reduction, operational leverage. Тільки після цього tasks потрапляють у roadmap.

40.2. Моя формула пріоритету

Я не використовую одну магічну оцінку ICE/RICE як істину. Базово дивлюся Impact × Confidence, а потім ділю на Effort і коригую Time-to-impact, dependency та reversibility.

Critical incident може мати невеликий upside, але максимальний priority через risk. Research spike може мати нуль прямого traffic impact, але дешево знімає невизначеність для великого проєкту.

40.3. Impact має бути в одиницях бізнесу

“Покращити internal linking” — не impact. “Збільшити частку money pages із ≤3 clicks from hub до 90% і підняти discovery/index coverage” — уже operating target. Для revenue-facing задач — qualified conversions, margin, FTD, revenue.

Для risk задач impact може бути avoided loss: наприклад, прибрати accidental noindex risk перед migration.

40.4. Confidence — це evidence stack

Я даю вищу confidence first-party observational data й контрольованим тестам, нижчу — correlation studies, ще нижчу — professional hypothesis. Це не означає, що гіпотези не можна робити; просто вони мають дешевше перевірятися.

Roadmap без confidence перетворюється на політичний список того, хто голосніше говорить на meeting.

40.5. Effort рахуйте по bottleneck

SEO може оцінити task у “2 години”, але якщо потрібен backend engineer, QA, security review і реліз через три спринти — real effort інший. Я рахую constrained resource, не тільки свої години.

Так само 100 articles — не “100 × writer cost”: потрібні research, brief, editorial QA, design/data, internal links, publishing, monitoring і maintenance.

40.6. Портфель горизонту

У нормальному roadmap є quick fixes, medium bets і long-term assets. Якщо все quick wins — через квартал немає moat. Якщо все platform work — бізнес не бачить feedback.

Я зазвичай тримаю окремі lanes: reliability/risk, growth, experiments, scale/automation, research. Частки залежать від стадії продукту.

40.7. Бюджет не дорівнює links budget

SEO-бюджет — people + engineering + content/data + distribution/PR + tooling + infrastructure + experiments. У affiliate link acquisition може бути великою статтею, але без site production і measurement вона не створює системи.

Я показую CFO не “нам треба $20k на SEO”, а allocation: що купуємо, який expected range, які leading metrics і коли review gate.

40.8. Stage gates

Великі ініціативи розбиваю на discovery → pilot → validation → scale. Наприклад, programmatic SEO: 50 pages pilot, потім 500, потім 10k — тільки якщо indexation, impressions і economics підтверджуються.

Це захищає від класичної помилки: побудувати 100k URLs, а потім з'ясувати, що dataset не створює унікальної цінності.

40.9. Dependency map

Кожна ставка має dependency: tracking, data source, CMS capability, legal approval, template, canonical logic. Я показую їх у roadmap явно.

SEO часто “запізнюється” не через SEO, а через hidden dependency. Винос залежностей зменшує конфлікти з product/engineering.

40.10. Roadmap review

Щомісяця я не просто питаю “task done?”. Я перевіряю assumptions: чи demand той самий, чи search surface змінився, чи pilot дав expected signal, чи capacity змінилася.

Roadmap — living decision model. Але зміни повинні бути версійовані, інакше команда щомісяця переписує історію заднім числом.

Таблиця. Практична матриця для цієї глави.

Ініціатива	Impact	Confidence	Effort	Time	Рішення
Fix accidental noindex	дуже високий risk reduction	висока	низький	дні	негайно
Programmatic cluster	високий upside	середня	високий	місяці	pilot
Title test top categories	середній	висока	низький	тижні	test
Новий GEO без demand data	потенційно високий	низька	середній	місяці	research first

Джерела до Глави 40

Ця глава переважно описує авторську операційну модель і не містить зовнішніх статистичних тверджень, які потребують окремої бібліографії.

ЧАСТИНА IX / SEO-КОМАНДА У 2026

ГЛАВА 41

Команда, KPI та звітність: що має контролювати Lead і Head

Сильна SEO-команда не вимірює людей кількістю текстів, links або закритих задач. Вона вимірює стан системи, швидкість навчання і бізнес-результат.

Стан даних і джерел: 14 вересня 2026 року

Рис. 41.1. Команда, KPI та звітність: що має контролювати Lead і Head

41.1. Ролі навколо системи, а не назв посад

На малому сайті Senior SEO може робити research, technical, content і links. На великому — це окремі functions. Я проектую ownership від

систем: crawling/indexing, architecture, content/product, authority, measurement, experimentation, international.

Назва “SEO Manager” не пояснює, хто відповідає за canonical incident. RACI/owner map пояснює.

41.2. KPI не повинен бути легко gaming

Кількість published pages стимулює thin content. Кількість acquired links — дешеві links. Середня позиція — вибір “зручних” keywords. Я використовую outcome + guardrails.

Наприклад, content team може мати eligible indexed pages із qualified impressions і conversion contribution, але guardrail — factual QA, decay rate, policy incidents.

41.3. Leading і lagging metrics

Revenue — lagging. Index coverage, crawl health, delivery SLA, experiment velocity — leading. Якщо дивитися тільки revenue, команда дізнається про проблему занадто пізно.

Але leading metric не можна перетворювати на самоціль. 99% index rate непотрібний, якщо ви індексуєте 200k безцінних URLs.

41.4. Reporting для Head

Я хочу бачити одну сторінку: business outcome, major changes, risks, bets, confidence, next decision. Не 40 charts GSC без питання.

До appendix йдуть деталізації: query/page type/GEO, technical health, AI visibility, experiments, links, delivery.

41.5. Weekly operating review

Щотижня: incidents, shipped changes, leading metrics, experiments, blockers, next releases. 30–45 хвилин достатньо, якщо data prepared асинхронно.

Meeting не має перетворюватися на live reading dashboard. Люди приходять із висновком і рішенням.

41.6. Monthly strategy review

Раз на місяць дивлюся cohort performance, roadmap assumptions, budget burn, market/competitor changes і learnings. Тут можна reprioritize.

Місячний review — не “пояснити CEO, чому traffic -7%”. Це рішення, де наступний долар/спринт дасть найбільший expected value.

41.7. KPI для Lead vs specialist

Lead відповідає за system output: roadmap quality, incident handling, delivery, measurement, team learning. Specialist — за owned domain і якість рішень, а не за vanity total traffic, який він не контролює.

Коли одна людина “відповідає за organic revenue”, але не контролює engineering, product, stock і CRO, KPI несправедливий і провокує політику.

41.8. Experiment throughput

У 2026 AI здешевлює analysis і implementation prototypes. Тому конкурентна перевага — не кількість generated ideas, а кількість якісно закритих learning loops.

Я рахую: hypotheses created → tests launched → interpretable results → scaled wins / killed ideas. Neutral result теж output, якщо зняв невизначеність.

41.9. Документація як leverage

Decision log, migration playbook, incident runbook, content schema, measurement dictionary — це assets. Вони зменшують dependence від пам'яті одного Senior.

Команда без документації кожен рік заново “винаходить”, чому не можна закривати staging через robots.txt тільки перед launch.

41.10. Що має контролювати Head

Head не повинен вручну перевіряти кожний title. Його контрольна система: business performance, technical risk, roadmap, capacity, budget, evidence quality, people, external dependencies.

Якщо Head весь день робить keyword research, організація втрачає роль, яка має з'єднати SEO з продуктом і P&L.

Таблиця. Практична матриця для цієї глави.

Рівень	Основне питання	Метрики
Executive	Чи створює канал цінність і який risk?	revenue/margin, qualified conversions, forecast range, incidents
Head/Lead	Чи працює система й portfolio?	non-brand clicks, coverage, delivery, experiment throughput, AI visibility
Specialist	Чи працює owned mechanism?	template/query cohort, indexation, CTR, links/content QA
Engineering/Product	Що треба доставити й як перевіримо?	acceptance criteria, logs, statuses, test result

Джерела до Глави 41

Ця глава переважно описує авторську операційну модель і не містить зовнішніх статистичних тверджень, які потребують окремої бібліографії.

ЧАСТИНА X / АГЕНТНИЙ ВЕБ

ГЛАВА 42

AI-агенти, машиночитний бізнес і транзакції без класичного SERP

Наступний search interface може не просити користувача відкривати десять вкладок. Агент сам знайде, порівняє, перевірить і виконає частину дії. Для SEO це новий клас "користувача" сайту.

Стан даних і джерел: 14 вересня 2026 року

Рис. 42.1. AI-агенти, машиночитний бізнес і транзакції без класичного SERP

42.1. Агент відрізняється від відповіді

Generative answer синтезує інформацію. Agent має ще й plan/action loop: отримати state, вибрати наступну дію, взаємодіяти з сайтом/API,

перевірити результат. Google у Search I/O 2026 описав information agents і розширення agentic booking; OpenAI ChatGPT agent використовує visual browser, text browser, terminal, APIs і connectors.[1] [2]

Це важливо: visibility може завершитися не click, а action. Для бізнесу це добре, якщо economics зберігається. Для analytics — складніше.

42.2. Сайт як machine interface

Агент може працювати через GUI, DOM, accessibility tree, structured data, API або feed. Чим стабільніша структура, тим менше ambiguity.

Це не означає “створіть schema для AI”. Google прямо каже, що спеціальної AI schema для AI Search не потрібно. Але стандартизовані Product/Offer/Organization/LocalBusiness data допомагають машинам і пошуковим системам розуміти факти у supported contexts.[3][4]

42.3. Дані, які повинні бути правдою зараз

Для transactional agent критичні price, availability, shipping, location, opening hours, refund/cancellation terms. Застарілий informational paragraph неприємний; застаріла availability може зламати дію.

Тому content ops поступово стає data ops: source of truth, freshness timestamp, validation, API/feed monitoring.

42.4. Merchant data як приклад

Google документує Product structured data і Merchant Center feeds як способи точніше передавати price, availability, shipping і returns; Merchant Center обов'язковий для деяких shopping surfaces.[4] Це вже модель “публікувати дані, а не тільки текст”.

Для інших вертикалей еквівалентом можуть бути booking feeds, inventory API, local business data, event feeds. Не все це є ranking factor; це infrastructure для accurate eligibility і actions.

42.5. Browser agent бачить UX як функцію

Operator/ChatGPT agent показав, що модель може клікати, вводити текст і скролити через GUI.[2] Отже, “машиночитність” не закінчується API. Form labels, accessibility, predictable states і error messages стають machine usability.

Captcha, modal chaos, hidden controls, infinite spinners, нестабільний DOM — це вже не лише CRO/accessibility debt, а agent compatibility debt.

42.6. Не блокуйте потрібний шлях випадково

SEO давно думає crawler access. Agent layer додає інші user agents, browsers і authenticated flows. Політика доступу повинна бути свідомою: що дозволяємо crawl, що дозволяємо action, де потрібна login/consent/payment confirmation.

Security важливіша за “бути доступним AI”. Не треба відкривати приватні endpoints заради agent friendliness.

42.7. Attribution без класичного referral

Agent може знайти merchant, перевірити умови і привести користувача на фінальний confirmation step. А може виконати більшу частину flow сам. Referral analytics не обов'язково побачить весь contribution.

Потрібні server-side order source, partner IDs, API attribution, assisted brand lift і експериментальні методи. Старий UTM-only worldview стає ще менш повним.

42.8. SEO для агента = reliability + semantics + economics

Я не створюю окрему “Agent SEO” checklist із 100 пунктів. Беру фундамент: reachable data, correct semantics, stable actions, evidence, brand/entity consistency, fast predictable UX.

Якщо агент постійно бачить різну ціну в HTML і checkout, жоден llms.txt не вирішить проблему.

42.9. Моніторинг agent readiness

Я тестую critical flows автоматизованим browser: знайти product/service, відкрити detail, перевірити availability, start booking/cart, handle error. Це synthetic monitoring, а не ranking test.

Для public data перевіряю parity HTML/structured data/feed/API. Divergence — operational incident.

42.10. Машиночитний бізнес

Сильний digital business у агентному вебі має не тільки pages. Він має чіткі entities, structured inventory, policies, APIs/feeds де доречно, stable transaction flow і source-of-truth data.

SEO тут стає interface discipline між business data і зовнішніми discovery/action systems. Це ближче до product/data engineering, ніж до keyword density.

Таблиця. Практична матриця для цієї глави.

Шар	Людина	Crawler/Search	Agent
Контент	читає текст/візуал	отримує HTML, render, index	витягує факти для плану
Структура	навігація	links/canonical/ schema	DOM/accessibility/ API/feed
Транзакція	клікає й вводить	зазвичай не виконує	може виконувати кроки
Метрика	session/conversion	impression/click/ citation	action success / assisted outcome

Джерела до Глави 42

1. Google. Search I/O 2026: AI agents and more. https://blog.google/products-and-platforms/products/search/search-io-2026/

2. OpenAI. Introducing ChatGPT agent. https://openai.com/index/introducing-chatgpt-agent/

3. Google Search Central. AI features and your website. https://developers.google.com/search/docs/appearance/ai-features

4. Google Search Central. Merchant listing structured data. https://developers.google.com/search/docs/appearance/structured-data/merchant-listing

5. Google Search Central. Share product data with Google. https://developers.google.com/search/docs/specialty/ecommerce/share-your-product-data-with-google

ЧАСТИНА X / АГЕНТНИЙ ВЕБ

ГЛАВА 43

Роль SEO після класичного пошуку: що залишиться, коли інтерфейс знову зміниться

Інтерфейс зміниться ще раз. Метод залишиться: зрозуміти систему, побачити дані, сформувати гіпотезу, перевірити й масштабувати.

Стан даних і джерел: 14 вересня 2026 року

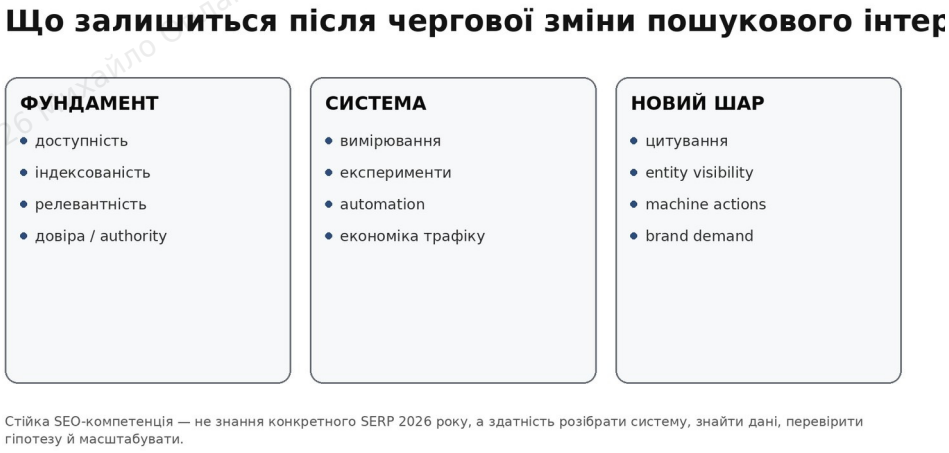


Рис. 43.1. Роль SEO після класичного пошуку: що залишиться, коли інтерфейс знову зміниться

43.1. SEO не дорівнює SERP

Десять синіх посилань ніколи не були самою дисципліною. SEO працює з discovery, crawl, index, retrieval, ranking/selection, presentation і conversion. AI Overview або agent змінює presentation/action layer, але фундамент не зникає.

Google у 2026 прямо пише, що best practices SEO залишаються релевантними для AI features і спеціальних вимог для appearance в AI Overviews/AI Mode немає.[1]

43.2. Позиція втрачає монополію на visibility

Сайт може отримати organic rank, citation, brand mention, product inclusion або agent action. Жодна метрика не є повною сама.

Тому сучасний measurement stack має працювати з multiple observable layers і чесно показувати blind spots.

43.3. Commodity knowledge дешевшає

LLM може швидко синтезувати generic explanation. Тому moat переходить у proprietary data, first-party experience, tools, fresh inventory, expert judgment, workflows і brand trust.

Це не означає, що пояснювальний текст непотрібний. Він потрібен як interface до унікального evidence, а не як копія top-10 SERP.

43.4. Technical SEO стає ще більш product-like

Коли Search і agents використовують render, structured data, feeds, browsers і APIs, межа між “SEO issue” і “product/data issue” стирається.

Senior SEO має розуміти HTTP, templates, logs, data contracts, experimentation. Не обов'язково писати production backend, але треба розуміти механіку достатньо, щоб поставити правильний acceptance test.

43.5. Brand — не магічний ranking factor

Бренд створює navigational demand, mentions, trusted first-party sources, reviews, link earning, repeat behavior. Це багато різних observable phenomena, а не один “brand score”.

У AI environments entity consistency і source quality можуть підсилювати ймовірність правильної інтерпретації. Але я не називаю кожен brand metric прямим ranking factor без evidence.

43.6. SEO Lead = allocator of uncertainty

Головна робота Lead/Head — не знати відповідь на все. Це швидко зменшувати невизначеність: які дані потрібні, який тест дешевий, що reversible, де risk asymmetric.

AI збільшує кількість можливих дій. Значить, selection quality стає важливішою за generation speed.

43.7. Автоматизація прибере операції, не відповідальність

Regex, SQL, clustering, briefs, anomaly detection, schema drafts і code prototypes автоматизуються дедалі краще. Але system owner все одно відповідає за definitions, data quality, risk і business decision.

Якщо AI швидко створив 50k pages, це не означає, що 50k pages треба публікувати.

43.8. Доказовість стане конкурентною перевагою

У світі нескінченного generated advice дорожчає здатність показати sample, method, source, limitation і causal boundary. Це стосується і контенту, і внутрішньої SEO-аналітики.

Команда, яка знає різницю між correlation, experiment і hypothesis, повільніше впадає в хайп і швидше знаходить working mechanism.

43.9. Що я б учив новому Senior у 2026

Не “200 ranking factors”. Я б учив web mechanics, information retrieval concepts, data analysis, experiment design, business economics, content differentiation, links/graph thinking і communication.

Після цього конкретний інструмент — Ahrefs, GSC, LLM, crawler — просто interface до задачі.

43.10. Модель, яка переживе 2026

Книга починалася з ланцюга від URL до AI-відповіді. Закінчується вона ще простіше: Understand the system → observe the data → form a hypothesis → test → measure → scale.

Коли Google змінить UI, назву AI Mode або спосіб показувати citations, ця модель не зламається. Саме тому я вважаю її кориснішою за будь-який список hacks.

Таблиця. Практична матриця для цієї глави.

Стара рамка	Стійкіша рамка
keyword → rank	information need → candidate set → selection
position → CTR	visibility surface → click/mention/citation/action
content volume	information advantage
SEO checklist	system model + evidence
traffic forecast	scenario + uncertainty + economics
manual operations	automation + QA + ownership

Джерела до Глави 43

1. Google Search Central. AI features and your website. https://developers.google.com/search/docs/appearance/ai-features

2. Google Search Central Blog. A new resource for optimizing for generative AI in Google Search. https://developers.google.com/search/blog/2026/05/a-new-resource-for-optimizing

3. Google. Search I/O 2026. https://blog.google/products-and-platforms/products/search/search-io-2026/

4. OpenAI. Introducing ChatGPT agent. https://openai.com/index/introducing-chatgpt-agent/

5. Google Search Essentials. https://developers.google.com/search/docs/essentials

ПІСЛЯМОВА

SEO як система мислення

Якщо після цієї книги залишити одну думку, я б залишив не список інструментів і не набір “факторів”. Я б залишив звичку розкладати проблему на механізм, дані й перевірювану гіпотезу.

Пошук уже змінився настільки, що проста модель “ключ → позиція → клік” не описує весь канал. Але фундамент не став менш важливим. URL усе ще треба виявити. Документ — отримати й відрендерити. Дублікати — канонізувати. Інформацію — відібрати під потребу. Сайт — зробити достатньо зрозумілим, корисним і надійним, щоб він був кандидатом для показу, цитування або дії.

AI зробив виробництво дешевшим. Саме тому дорожчають відмінність, evidence і judgment. Згенерувати ще сто статей легко. Значно складніше зібрати власний dataset, перевірити умови десяти операторів, провести контрольований тест, побудувати tool або знайти системну помилку в мільйоні URL.

У роботі я не намагаюся вгадати кожну зміну Google. Я хочу, щоб команда знала, які сигнали вона реально бачить, де дані неповні, що є фактом, що кореляцією, а що нашою робочою гіпотезою. Це дає спокій у момент апдейту і швидкість у момент можливості.

SEO у 2026 році все ще про пошук. Просто тепер пошук може закінчитися SERP, AI-відповіддю, цитуванням, recommendation, agent action або прямою транзакцією. Наша задача не захищати старий інтерфейс. Наша задача — зробити інформацію й бізнес доступними, зрозумілими, конкурентними та вимірюваними в тому інтерфейсі, який існує сьогодні.

Інструменти зміняться. Назви дисциплін теж. Модель залишиться: зрозумій систему → подивись на дані → сформулюй гіпотезу → перевір → виміряй → масштабуй.

ПОДЯКИ

Людям, завдяки яким цей шлях має сенс

Найперше дякую моїм батькам - за все. За життя, любов, підтримку, терпіння і внутрішню опору, яка залишалася зі мною навіть тоді, коли я сам її не помічав.

Дякую братові - за близькість, чесність і просте, але дуже важливе відчуття, що поруч є своя людина.

Дякую друзям - усім, хто слухав, сперечався, підтримував, підказував і допомагав не зупинятися. Частина цієї книги виросла з наших розмов, робочих історій, помилок і спільного досвіду.

Окремо дякую Н. - за все хороше, що було між нами протягом багатьох років, і за безліч теплих, приємних та по-справжньому крутих моментів. Не все важливе потребує повного імені: деякі спогади достатньо просто берегти.

На обкладинці стоїть одне ім'я, але жодна велика робота не народжується у повній самоті. У цій книзі є тепло, час і віра людей, яким я щиро вдячний.

ПРО ШІ ТА ЦЮ КНИГУ

Інструмент допомагав. Автор відповідає.

Було б дивно писати книгу про SEO у 2026 році й удавати, що штучний інтелект залишився поза процесом. Звичайно, я його використовував - для структурування матеріалу, пошуку слабких місць, перевірки формулювань, роботи з великими масивами тексту та технічної підготовки.

Для мене це такий самий робочий інструмент, як краулер, редактор коду чи таблиця: він може прискорити роботу, показати інший кут і допомогти перевірити деталі. Але він не проживає професійний шлях за автора, не приймає рішень на реальних проєктах і не несе відповідальності за висновки.

Досвід, позиція, сумніви, добір фактів, остаточні формулювання й кожне рішення про те, що залишити на цих сторінках, - мої. Я не ховаю інструмент, бо ця книга якраз про роботу в новій реальності.

ШІ допоміг зробити цю книгу ширшою і точнішою. Її характер, відповідальність і душу вклав автор.

Михайло Оплаканець aka Werter