Typewise AI Discoverability Audit

Preliminary observations & hypotheses

Prepared by: Anindo Neel Dutta

Time spent: Approximately 3–4 hours.

This isn't an exhaustive audit.

I became interested in Typewise after noticing an interesting pattern while working on one of my own products, DocPilot.

Despite very little active marketing over the past few months, DocPilot continued receiving citations from AI systems. That observation made me curious about how modern LLMs discover, retrieve, and recommend software products.

When I started researching Typewise, I expected to find a company that was consistently recommended across common enterprise AI customer support queries.

Instead, I found a noticeable gap between the quality of the product and how frequently it was recommended by AI systems.

This document summarizes my initial observations, hypotheses, and what I'd prioritize if I joined the team.

Observation 1

Product quality doesn't match AI visibility

I tested a series of common enterprise customer support discovery queries across ChatGPT, Claude, Gemini, and Perplexity.

Examples included:

- Best enterprise AI customer support platform
- Best AI support agent software
- Enterprise AI helpdesk
- AI customer support software
- AI customer service platform
- Zendesk alternatives

Across these queries, competitors such as Zendesk, Intercom, Ada, Salesforce, Kore.ai, NICE, and Botpress appeared consistently across multiple AI systems.

In contrast, I rarely saw Typewise recommended—even for queries where I expected it to be highly relevant. This pattern was consistent enough that it immediately stood out to me.

Observation 2

This doesn't appear to be an engineering problem

From what I observed, Typewise already has many of the qualities I'd expect from a leading enterprise AI company.

- Strong visual identity
- Fast, polished website
- Enterprise positioning
- Strong customer credibility
- Industry recognition

These don't look like the bottleneck.

Instead, I think there's an opportunity to improve how AI systems understand, categorize, and retrieve information about Typewise.

Hypothesis

My current hypothesis is that discoverability is limited more by information architecture and technical messaging than by product quality.

A few examples stood out to me.

The homepage communicates customer outcomes very well, but developer capabilities such as APIs, SDKs, integrations, and technical workflows require more exploration than I expected.

Because LLMs rely heavily on explicit, structured technical information, I think these capabilities deserve much greater prominence throughout the website and documentation.

Similarly, I'd expect more implementation-focused content such as:

- API guides
- SDK documentation
- Integration tutorials
- Architecture documentation
- Technical comparison pages
- Developer use cases

This type of content is useful for developers, but it's also exactly the sort of information AI systems can reliably retrieve and reference.

Why I think this matters

LLMs don't recommend products simply because they're the best.

They recommend products because they repeatedly encounter clear, structured, and consistent information describing:

- What a product is
- Who it's for
- What problems it solves
- When it should be recommended

The easier this information is to retrieve and reinforce, the more frequently a product tends to appear during AI-assisted product discovery.

A small experiment from my own work

The observation that sparked my interest in this space came from DocPilot.

Although I hadn't meaningfully worked on the product for nearly a year, Bing Webmaster Tools began reporting sustained AI citations around March.

AI Performance (Last 90 Days)

- 185 AI citations
- 118 cited pages
- 76 / 90 days with citations
- Average: ~2.1 citations/day
- Peak: 11 citations in a single day

I don't present this as proof that I've solved AI discoverability.

Rather, it made me curious about why some products continue to surface consistently while others remain relatively invisible.

That curiosity is what eventually led me to analyze companies like Typewise.

What I'd do in my first 30 days

Week 1 — Establish a baseline

- Audit AI mentions across major LLMs
- Benchmark competitors
- Review technical documentation
- Evaluate structured data and information architecture

Week 2 — Strengthen technical positioning

Prioritize clearer messaging around:

- APIs
- SDKs
- Integrations
- MCP (where applicable)
- Developer workflows

Week 3 — Build AI-readable content

Focus on content that benefits both developers and AI systems.

Examples include:

- Integration guides
- Technical implementation tutorials
- Comparison pages
- Architecture documentation
- Enterprise deployment examples

Week 4 — Build a measurement system

Create a repeatable process for monitoring:

- AI citations
- LLM visibility
- Query coverage
- Competitor movement

Then iterate continuously based on real-world observations.

Final thoughts

This isn't intended to be a comprehensive audit.

It's simply the result of spending a few hours researching a company whose product I genuinely found interesting.

I'm sure many of these hypotheses would evolve after speaking with customers, reviewing analytics, and understanding internal priorities.

The purpose of this document isn't to present definitive answers.

It's to demonstrate how I approach unfamiliar problems:

- Observe patterns
- Form hypotheses
- Validate them
- Iterate quickly

I hope some of these observations are useful, regardless of whether we end up working together.

— **Anindo Neel Dutta**

Appendix A — Methodology

This audit is based on exploratory testing across multiple leading AI systems:

- ChatGPT
- Claude
- Gemini
- Perplexity

I tested a range of common enterprise customer support discovery queries and compared:

- Whether Typewise was recommended
- Which competitors appeared consistently
- How each platform described the products
- Common positioning patterns across AI systems

The objective wasn't to produce statistically significant benchmarks, but to identify qualitative patterns in AI-assisted product discovery.

The complete prompt logs and response data are available upon request.

Appendix B — Why This Problem Interests Me

My interest in AI discoverability started after noticing an unexpected pattern in DocPilot's AI Performance data.

Rather than treating those citations as a vanity metric, I became interested in understanding **why** AI systems repeatedly surfaced certain

products.

That curiosity has led me to spend time researching how product messaging, documentation, structured content, and information architecture influence AI recommendations.

This audit is one example of that ongoing exploration.