

USE CASE · TEAM KNOWLEDGE & ONBOARDING

A knowledge base that **answers back.**

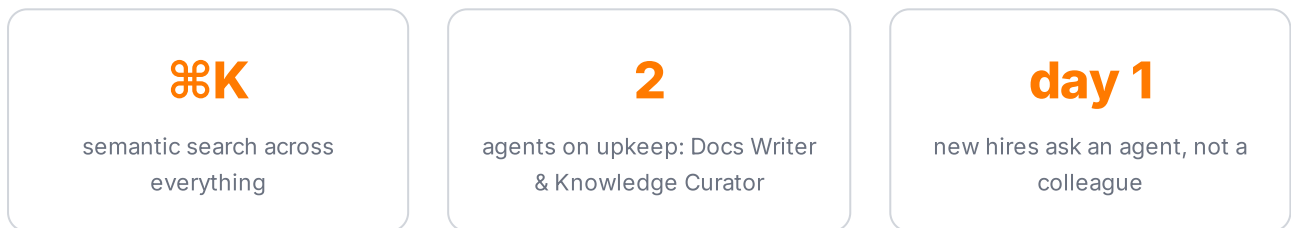
How teams use Neuphlo to build a living knowledge base — pages, courses and published sites — kept current by agents, searchable by meaning, and able to onboard the next hire by conversation.

§ Documentation debt is people debt

Every team owes itself documentation it never wrote. The cost arrives as interruptions: the same question asked for the fifth time, the new hire blocked on day two, the process that lives in one person's head.

Wikis fail for a predictable reason: writing them is extra work, and reading them is keyword roulette. The knowledge exists — in closed tasks, in threads, in the heads of the people who shipped the thing — but turning it into pages and keeping those pages true costs effort nobody has.

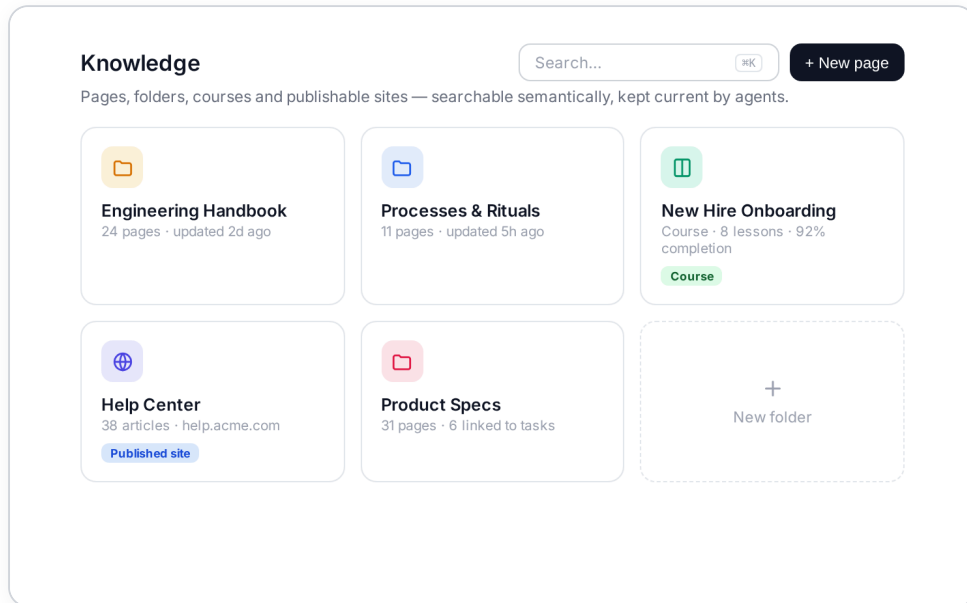
Neuphlo's answer is to make knowledge a **by-product of work** rather than a chore beside it: pages live next to the tasks they describe, agents draft and update them from what actually shipped, and a conversational agent answers from them — flagging the gaps it finds.



In this paper: the Knowledge module — folders, courses and published sites — the agents that keep it current, and onboarding as a conversation. Screenshots are taken directly from the product.

§ Pages, courses, sites

Knowledge in Neuphlo is more than a wiki tree: folders of pages, structured courses with completion tracking, and whole sections publishable as public sites.



The Knowledge home. Handbook folders beside an onboarding course (92% completion) and a help-center site published to your own domain.

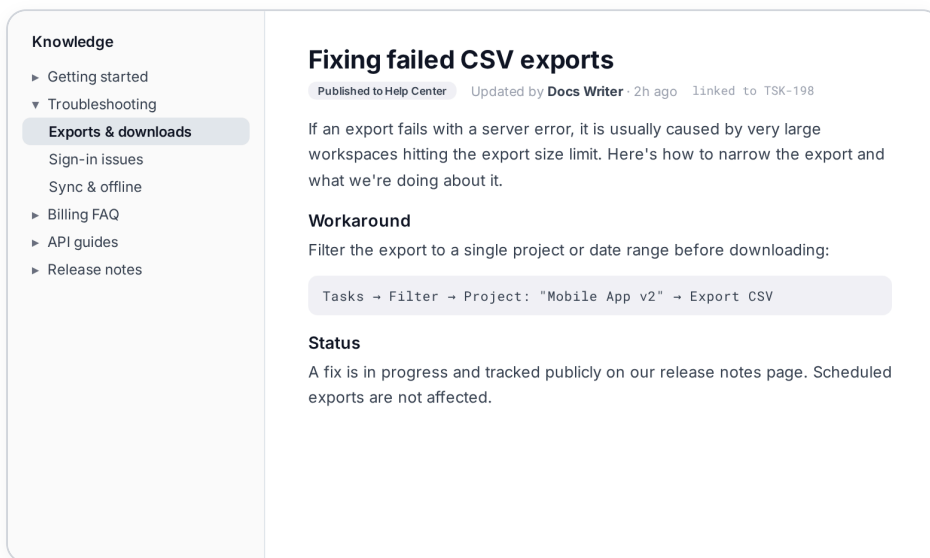
Pages link to tasks in both directions — a spec references the work implementing it, and closed tasks point back at the doc they updated. **Semantic search** means "how do I get staging access" finds the environments page even when no word matches; the same search powers every agent's grounding.

§ Agents on the maintenance shift

The reason wikis rot is that maintenance is nobody's job. In Neuphlo it's two agents' job.

Docs Writer turns shipped work into articles: mention it on a resolved task and it drafts the documentation — overview, steps, pitfalls — links it to the task, and files it in the right folder. Documentation happens at the moment the knowledge is freshest, by an agent that read the whole thread.

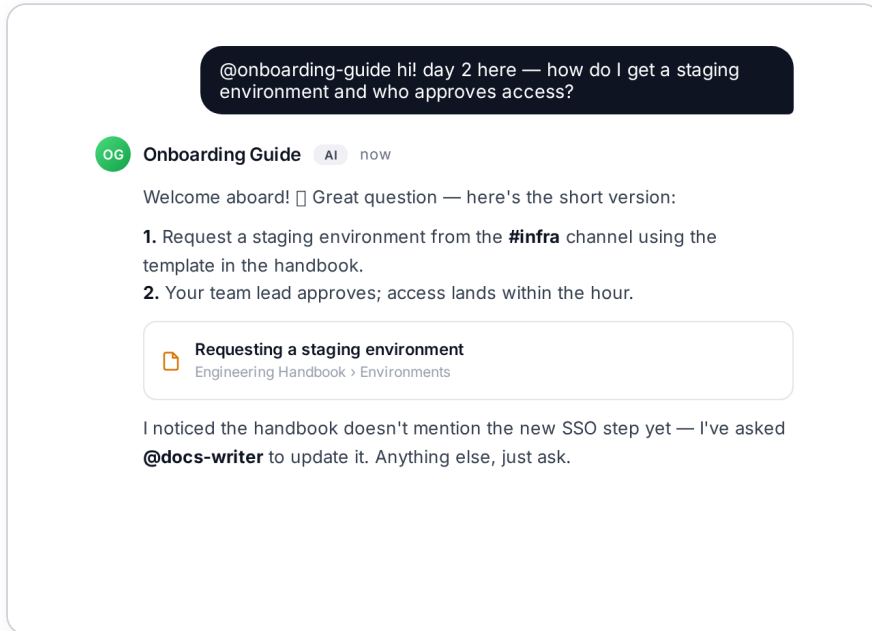
Knowledge Curator patrols the structure: duplicate articles, stale content, weak titles, misfiled pages. It merges, retitles and re-files — and explains what it changed and why discoverability improves.



From task to page to public site. An article updated by Docs Writer, linked to the engineering task it came from, published to the help center.

§ The new hire's first teammate

Onboarding is where knowledge debt gets paid with interest. The Onboarding Guide agent answers from the handbook — patiently, on day one, without anyone being interrupted.



Question to answer to fix. The guide answers with steps and the source article — and, having found a gap in the handbook, delegates the update to @docs-writer.

That last move is the system working as designed: every onboarding question is also an audit of the documentation. Gaps don't get worked around — they get filed and fixed. Structured **courses** carry the curriculum side: an eight-lesson onboarding course with completion tracking replaces the "read these 40 links" email.

§ How knowledge compounds

Each part feeds the next — work produces pages, pages answer questions, questions expose gaps, gaps become pages.

| MOMENT | WHAT HAPPENS |
|------------------------------|---|
| Feature ships | Docs Writer drafts the article from the closed task |
| Question asked in chat | Onboarding Guide answers with the source linked |
| No article exists | The gap is flagged; Docs Writer is delegated the draft |
| Quarter ends | Knowledge Curator merges duplicates and retires stale pages |
| Customer asks the same thing | The article is already on the public help-center site |

Beyond the team

Published sites extend the same knowledge outward — a public help center, a partner handbook, course material on your own domain — all maintained in the same place the team already works, with the same agents on upkeep. Write once; answer everyone.

§ Institutional memory, under your control

- **Your knowledge, your model.** BYOM means the handbook is processed by the provider you choose — or stays entirely on-prem with a local model on self-host.
- **Agents that cite.** Answers link their source articles, so "the agent said so" is always one click from the page that says it.
- **Bounded edits.** Curator and writer agents work within tool scopes; what they changed is a visible, reviewable run.

§ Getting started

Enable the **Knowledge** app, create two folders — a handbook and a troubleshooting section — and add Docs Writer, Knowledge Curator and Onboarding Guide from the template library. Seed ten pages, then route the next new hire's questions to the guide and watch the gaps surface themselves.

In short: Neuphlo turns documentation from a chore into a flywheel — written by agents from real work, searched by meaning, taught by conversation, and published wherever it's needed.