

What is a knowledge base, really? A practical guide for teams building one

By Nabila Husseni | Knowledge Management | September 2026

When I was working at Citi as a Knowledge Management Author, I inherited a knowledge base that had been built over several years by different teams with different ideas about what it was for.

Some sections looked like an internal Wikipedia. Others looked like a filing cabinet. Some articles were step-by-step guides. Others were policy documents. Some were clearly written for new employees. Others seemed to assume you already knew everything and just needed a reminder.

It was all technically a knowledge base. But it was not actually functioning as one. People were not using it. They were asking each other the same questions it was supposed to answer. When I dug into why, the answer was pretty consistent: they could not find what they needed, and when they did find something, they were not sure if they could trust it.

That experience taught me a lot about what a knowledge base actually needs to do, and how most of the advice about building one skips the part that matters.

What most definitions get wrong

If you search for a definition of a knowledge base, you will find something like: a centralized repository for storing and managing information. Which is technically true and practically useless.

A filing cabinet is a centralized repository for storing and managing information. So is a shared drive full of PDFs named 'final_v3_revised_ACTUAL_FINAL.pdf'. The problem is not storage. It is whether people can find what they need, understand it quickly, and trust that it is current.

A knowledge base is not a place to put information. It is a system for helping people answer their own questions.

That shift in framing changes everything about how you build one.

If it is a storage system, you organize it the way that makes sense to the people creating the content. By department, by document type, by project. The information goes in, it is technically findable, done.

If it is a self-service system, you organize it the way that makes sense to the people who need answers. By the question they are asking, by the task they are trying to complete, by the moment they are in when they need help.

Those two organizing principles produce very different knowledge bases. And the first one is why most knowledge bases fail.

The three things people actually need

At Citi I did something that felt obvious but apparently is not standard practice: I went through the support tickets and internal questions that were coming in repeatedly, and I used those to figure out what the knowledge base needed to cover.

Not what we thought people needed. What they were actually asking for.

Most of the questions fell into three categories:

- How do I do this specific thing? Step by step, in plain language, without assuming I already know the context.
- What does this term or policy actually mean in practice? Not the official definition. What does it mean for what I am trying to do right now.
- Something went wrong. What do I do? This was the category most underrepresented in the existing knowledge base.

That third category is where a knowledge base earns its value. Anyone can document the happy path. The thing that actually saves people time is documentation that tells them what to do when something does not go as expected.

If your knowledge base does not have a strong troubleshooting and exception section, it is not really a knowledge base. It is an optimistic description of how things work when they work.

The freshness problem

The other thing I kept running into at Citi was articles that were accurate when they were written and wrong now. Processes had changed. Policies had been updated. Systems had been replaced. But the knowledge base articles still described the old way.

This is the thing nobody talks about when they talk about building a knowledge base. The build is not the hard part. The maintenance is the hard part.

An outdated knowledge base is worse than no knowledge base. With no knowledge base, people ask each other and get the current answer. With an outdated knowledge base, people read the wrong answer confidently and make mistakes.

Content with no owner goes stale. Every article needs someone responsible for keeping it current.

The fix is not complicated but it requires discipline. Every article needs an owner. Every article needs a review date. When processes change, the people responsible for those processes need a way to flag the articles that cover them.

This sounds like overhead. It is actually the thing that makes the difference between a knowledge base people trust and one they eventually stop using.

How to know if yours is working

The simplest signal is whether people are using it without being told to. If people only consult the knowledge base because they were directed there, it is not working well enough to be a self-service tool.

Other signals worth watching:

- Are the same questions still coming in on topics the knowledge base covers? That means either the content is not findable or it is not clear enough.
- Are people finding the troubleshooting content? If the troubleshooting section has very low traffic but you are getting a lot of support questions, that is a findability problem.
- When you ask someone to find something specific in the knowledge base, how long does it take? If it takes more than a minute or two, the structure is not working.

None of these are complicated to measure. They just require paying attention to how people are actually using the thing rather than assuming the content is doing its job because it is there.

The short version

A knowledge base is a self-service system for helping people answer their own questions. It is organized around the questions people ask, not around the structure of the organization that built it. It covers the troubleshooting cases, not just the ideal path. It has owners for every article and a process for keeping content current. And it gets measured by whether people can actually find and use it, not by how much content it contains.

Most of the knowledge bases I have seen fail on one or more of those criteria. Usually the organization structure one and the freshness one. Those two are fixable with process changes that do not require rebuilding everything.

Start there. Get those right first. Then worry about the rest.

Nabila Husseni is a technical writer and knowledge management specialist with direct enterprise KM experience from Citi and Ericsson. nabilahusseni.com