Notion

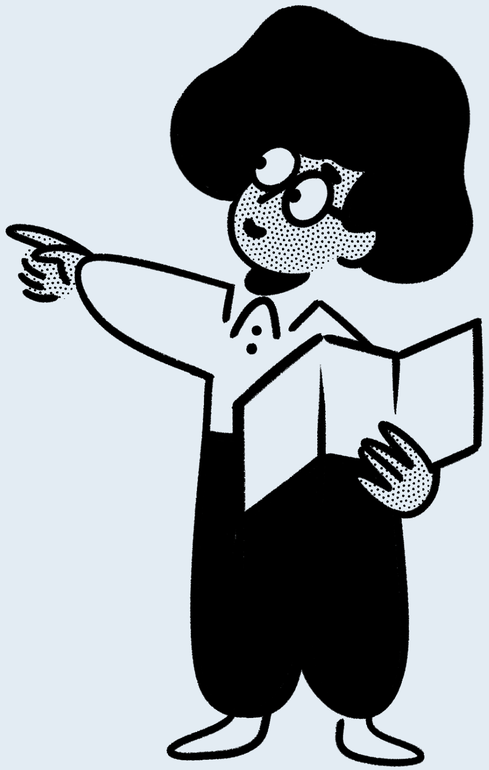
Use this framework to document knowledge at your company

Knowledge at your company shouldn’t feel like buried treasure. But often, it does — when there’s no searchable source of truth, teams are wasting time, working incorrectly, or unnecessarily duplicating work.

You’re failing if:

1. Knowledge only lives in the heads of employees
2. Documentation isn’t easily found / used by teams

To fix that, knowledge must be documented — and that documentation needs to be set up in a way where it’s accessible, so teammates actually use it.



Think about documenting knowledge as “how” vs “why”

Together, they give people the information they need to succeed — combining executional guidance and context.

- **“How” →** useful when your team needs to execute and scale. This helps teammates operate more autonomously.
- **“Why” →** conveys context. Helps create buy-in and transparency when teammates know the purpose behind their work.

4 ways to document knowledge so it’s usable and actionable

It’s likely your company already has a lot of documentation swirling around. To be useful, it needs to be organized, transparent, and up to date.

Here’s how you can think about building a system for documentation, creating order out of chaos — instead of adding to the swirl.

1. Know where documentation should go

Documented knowledge is useless if it’s floating around the digital ether. Knowledge needs a home (in a document). Thus, your documents need a home.

To organize documentation, think about who’s using it and what they’re using it for.

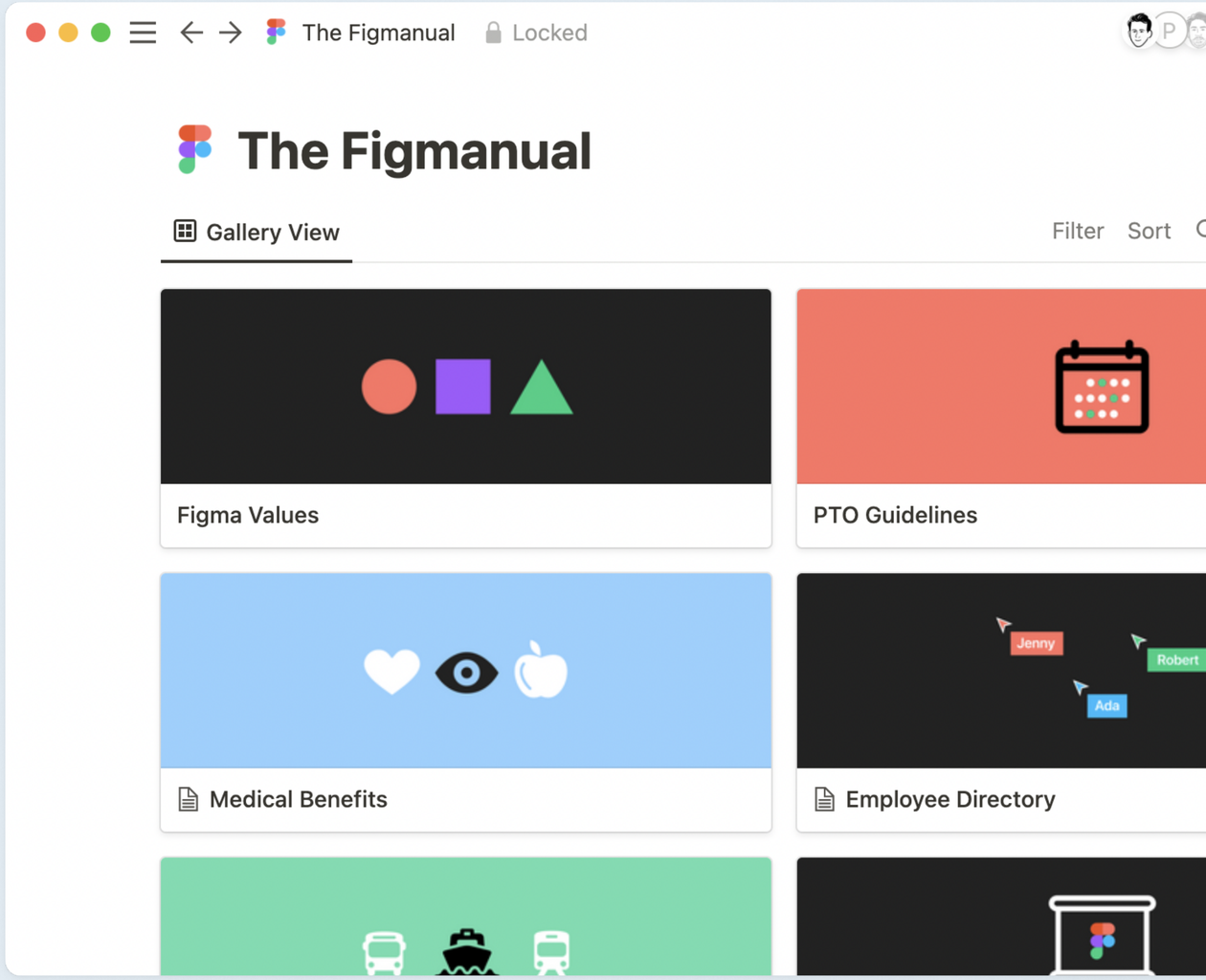
1. **Know what documentation the whole company needs and put it on one place** — things everyone uses should go into one, clear, organized space.
2. **Know what documentation is team-specific, and separate it by team** — create individual team wikis that house all their information, like processes, projects, and structure. This creates transparency too!
3. **Do you need a place for shared processes or docs?** — you might want a single database of all your company’s memos, meeting notes, or bug tickets.

See it in action: Figma's knowledge base keeps everyone informed and aligned

Figma puts all company info in one section of its wiki, and each has its own page: one for benefits, one for policies, one for OKRs, etc. This makes everything accessible to people who join the company, and it consistently gets updated as new information is added — meaning everything is fresh and up to date.

Figma also customizes its Notion sidebar. Each team has its own wiki. These wikis house information for individual teams, but they're also helpful when cross-functional teams need info. With specific headers, databases, and more, everything is organized and easily found, whether it's members of that team or partners.

Duplicate their template → here



2. Start by documenting explicit knowledge — it’s the easiest one

Explicit knowledge *comes* from documentation, meaning it’s gained by reading structured, recorded information. Usually it details how to do something (and will probably make up a majority of your company’s documentation). Examples include data sheets, how-to’s, reports, and any other instructional documents.

It's most obvious to start here when it comes to documentation (and some of it probably exists already). Consider all the information teams and individuals need to work and organize it by the principles mentioned above.

When documenting explicit knowledge, remember...

1. **It should be organized and usable —** think about information hierarchies and how to nest relevant knowledge.
2. **It should be easily accessible** — and teams should know where to find it. A standardized naming convention can go a long way.
3. **Think about it as if you were onboarding to the company** — what do I need to know to do my job?

3. Incorporate the tacit — get information out of people’s heads

Tacit knowledge is gained from experiences. It’s rarely written down, and much of it might be considered intangible. You might get tacit knowledge from a conversation or a personal experience. Much of it provides important context, like ways your team / company work, leadership guides, or how you make decisions.

Figure out which types of tacit knowledge you might be able to document and turn into explicit knowledge. This enables scale.

A good place to start is with all your documented explicit knowledge. What tacit knowledge can be folded in?

Then, go broader. Ask your team…

1. What do you wish you knew about teams, processes, or how the company operates when you started working here?
2. What have you learned in the last six months that has improved your workflows?
3. What are the top three unwritten rules of our company?

See it in action: Branch's living documentation system propels enterprise growth

The documents for Branch’s engineering team used to be scattered around Google Docs, Confluence, and Jira. People didn’t know where to find information — so no one wanted to write anything down.

Now, they’re able to keep engineering documentation up to date, giving Branch a detailed history of major decisions and a foundation on which to continue building the team. Because documentation lives in Notion, it’s all in one place, making it easier to find, use, and update with new information.

Duplicate the template → here

"People actually enjoy writing documentation in Notion. 50% of our documentation was newly created because people are motivated to do it. The whole team is more productive because the information we're looking for is easier to access, and that information is more thorough.”



Dmitri Gaskin,
Co-founder of
Branch

4. Maintain and update with implicit knowledge on a quarterly basis

You gain implicit knowledge by putting explicit knowledge to work, like running a process, and learning from what's been documented. It's not something that's usually written down, but it should be — because it helps everyone perform better and build knowledge. Examples include sales techniques used by specific account executives or better ways of working with cross-functional teams.

When implicit knowledge is added to your documentation, you are providing teammates with the most up-to-date information, helping avoid confusion or the running of outdated processes.

It's easiest to create a schedule to review your documentation, such as at the start of every new quarter.

- **Start with company-wide, explicit documentation** — update the docs the most people use at the company.
- **Then, move to team-specific, explicit documentation** — have team leads update their process docs and team wikis.
- **Encourage the bottoms-up motion** — those who are actually using explicit documentation know how it can be updated. Ask for support from individual employees, creating buy-in across the entire team.