



We deserve better than Confluence and Notion

If you ever were in the situation of looking for a tool to manage the knowledge of your organization, chances are you ended up choosing between Notion and Confluence. A handful of other tools also exist in this field, large players like Microsoft Sharepoint or Google Drive, niche tools like Slite, Guru, Coda etc and very old weird guys (KMS, I'm looking at you).

Sun 26 September 2021

by Paul
in Opinion

In the end, this critical and fragmented market is mostly served by Notion and Confluence, especially for startups and fast-growing companies.

Confluence is an old tool (V1 in 2004) and is admittedly hated by its users, as for any product made by Atlassian really. I'm not convinced Confluence deserves this rather harsh criticism given that it is overall better than its competitors.

Notion is the new, fancy note taking tool. It's the new Evernote and its growth is huge, but it was never built as a tool for organizations. Rather, it properly serves the needs of individuals and very small teams though now its creators are working hard to try to pivot into the Enterprise game.

What is important for an organization

The two objectives organizations try to achieve using these tools are making it easier to find information and encouraging the creation of new content. The issue with these two objectives is that they can easily become opposite goals. If the information architecture is not properly defined, additional content adds complexity and makes finding relevant information more frustrating, harder and time consuming.

Let's take an example, Confluence and Notion both advertise that they can have infinitely nested folders. But is that really a good thing? If you are the one creating the structure, you will give titles for each folder that match the way your brain works and you will be able to find deeply nested content. But for any other person, the folder system becomes a maze after the second or third nested folder. We have all heard of "I can't find document X" - "Oh it is easy you need to go in Q then P then T and yeah for this one it is a bit unusual you also have to search for C instead of X, and here you have it, easy no?" The structure is basically helpful to almost no one and if you want to find something you have to scan everything. The result is that deep nested folders should be banned because they are unhelpful.

Making information easy to find needs to include knowledge that belongs in other tools. It is not because your company has Notion that employees will stop using Google Sheets, Google Presentations, Microsoft Words, Figma, Github and so on. Some tools are amazing tools and heavily specialized in their respective fields, but each tool is also its own silo. You are never going to replace Google Sheets or Excel with Notion tables, and neither are you ever going to use Confluence as a filesystem storage to replace Google Drive. This means that your knowledge's single-source of truth needs to accept external tools content as possible documentation. This interfacing, though, can't simply be about linking files and urls in one place, it has to be deeply integrated so that it feels natural, native even. We should be able to put a Google Sheet file in a folder, attribute tags for example.

If we want to have a common single source of truth of knowledge through the same software we also have to take into account the variability of skills of members. Some teams are using these products way more than others, some users will look up information every day, when others will need it once a year. Some people like dealing with a flat file system while others prefer nesting folders very precisely. If you don't have a tool in which all members feel welcome they will not use it. This makes user experience a determining factor of success. An area in which Notion was particularly good especially for non technical teams compared to Confluence.

But we should not make the mistake of being too flexible either. There is a balance to be found to keep a common framework. That's why not everyone should be responsible for information architecture. A place where Notion and Confluence have chosen opposite approaches. Notion doesn't provide its users with an admin interface, changing the information structure is easy and done quickly, newly created content doesn't seem to belong to some precise location, all users have rights on anything and information seems to be floating around. Confluence makes it more rigid, it provides admin interfaces and a very detailed rights-management system, but admins feel powerless to handle all new contents and for making sure users respect the framework. And as users are not made to feel responsible for properly organizing the content they add, it is nobody's responsibility and things become a mess .

This result is not a consequence of unawareness because a lot of people understand the value of creating and using documentation, but tools are built in such a way that it ends up in chaos and nobody is able to use them as they wish.

Is better than that even possible?

Better structure

Considering the limitations of Notion and Confluence approaches, is there a better way to handle knowledge? We could for example use Google style of search to simplify finding information. As tempting this solution is, it presents unbearable downsides. Most of the criterias that make a search engine powerful are not working in the context of a business organization: using backlinks to rank results, matching synonyms to understand meaning, skipping words, words stemming, using large amounts of data and learning from it etc. None of this really works for an organization's internal knowledge where data volumes are low and acronyms are the norm (IT can be Information Technology or the verb, same for THE, how can we know that 'dockerd' is not a typo ? ...).

But it is also too difficult to search in the raw data. That means it needs to be indexed, which can be achieved in many different ways:

1. Put data in tree structures.
2. Assign tags to data.
3. Assign metadata to data.
4. Group data by similarity.

The first approach, putting data in tree structures is what's being used by most softwares out there. It can make searching for information way faster by reducing the list of documents to scan, but this is helpful only if folders are pretty clear of what they contain. So we should have a few fat folders and use other indexing methods inside them. For example there is a clear difference between Sales documents and engineering specifications. But there is no clear border between pricing strategy and sales presentations in an organizations of 100 ppl, so these two contents should be in the same folder

Assigning tags to data, multiple tags for a single document for example, helps avoiding the main downside of using a folder tree structure, i.e not knowing to which folder belongs a file. Though, as for the tree structure that needs to be meaningful and understood, the list of tags must be consistent. Only specific owners should be able to define allowed tags so that they can make sure the tag system is “balanced”.

Metadatas are usually preferred to tags as they allow for precise filtering and impose a common schema for indexing data. Though, their main downside is that they can only be used with similar content. It is a very robust yet limiting framework: if you want to index meeting notes, names of participants and date of the meeting are relevant metadata, but what if you need to create a totally different type of document? Metadatas are a powerful tool (as Notion showed building their Tables) but limited to cases where the data itself answers to a common pattern. One way to go beyond this limitation is not to give metadatas to documents but to give metadata depending on where they are searched. The same document “Meeting note about the board meeting” could have the metadata “date of the meeting” and “participants” when viewed from the list of meetings notes. But will have the metadata “type of document=meeting notes” when viewed from the list of document related to board decisions Grouping data by similarity has been made popular by Pinterest or Slack, under the terme “curation”. Any document can be shared in as numerous Slack channels deemed relevant. In Pinterest, each pin can exist in several collections. The collections and channels facilitate the finding of common documents (according to the definition of the collection). Even better, those collection-types can be created by other people than the creators of the document and then offer some kind of transversal structure in which you can impose a different metadata schema inside each collection. Each owner of a collection becomes the curator of the content of other peoples.

All of these solutions are relevant and useful for enhancing discoverability and relevance of organizations’ internal knowledge. In each case, it's only when everyone shares the same framework that information becomes accessible. Those tools need more than anything to build alignment between people.

Ease of use

If you implement a new tool in your company to create documentation but the collaborating experience is worse than Google Docs most people will use Google Docs and nobody will use your new tool. As a company executive or knowledge management worker you understand the benefit of having the same tool to store all your information. But on a day-to-day basis, if you have to decide for a folder, choose tags, choose a title, publish your document, invite people to comment and use a text editor with 10% of the capacities of Google Docs’ or Excel’s, that's never going to work in the long term. That's exactly what happens with Confluence, first people hate Confluence, then they don't know where to put their document, then that's probably the first time they try to create content, so they don't have rights to create content, and finally people don't receive updates to their comments because they are used to ignoring notification emails from Confluence.

A typical user of any knowledge management solution must be able to execute a few actions:

- Create new content.
- Collaborate with peers on a document.
- Quickly retrieved a previous document deemed useful.
- Explore and find specific content.
- Be notified when content is updated.

Notice how this list doesn’t include “Create rich text documents better than Google Docs”. Most existing tools do not focus on the organization part: Notion Tables are arguably an attempt to reimplement Excel inside Notion, and Confluence is trying to make people use smileys in their pages title.

We need tools to organize content, we don't need to recreate Google Sheet and JIRA inside knowledge base tools. Current vendors should make more effort to make external documents first class citizens of content and leverage existing tools, as discussed before. We could push things further and allow creation of external content from the knowledge base or one-click add an external url to the documentation

If you are interested in any of these topics, have ideas or if you are looking to solve those issues, don't hesitate to reach out to me. I'm building on those topics at [Dokkument](#). A collaborative knowledge management solution for organizations.

Categories

- [dating](#)
- [DIY](#)
- [life](#)
- [Opinion](#)
- [Product Management](#)
- [startup](#)
- [Tech](#)

Social

[Contact me](#)

Blogroll

- [My Twitter](#)
 - [HN blogs](#)
 - [Dokkument](#)
-