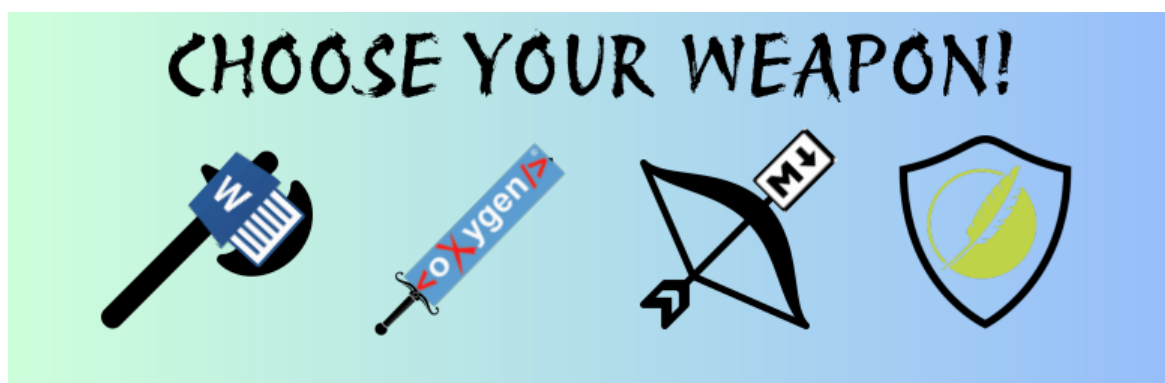


# From Word to Content Management Systems: A Study of Tools Used by Technical Writers

MARTINEZ GANDARA Eduardo, PELLERIN Valentin, YANG Christian

Université Paris Cité, June 2025



This study explores the evolving landscape of tools used by technical writers, from traditional word processors like Microsoft Word to modern Content Management Systems (CMS) such as Oxygen XML Editor and GitHub.

The research is based on a survey of 38 professionals from various industries and experience levels. The study analyses how factors such as company size, documentation team structure, professional experience and translation requirements influence tool selection. The findings indicate that the adoption of tools is context-dependent, with simpler tools being favored by smaller teams and early-career writers. Conversely, more complex systems are adopted by experienced professionals and large organizations managing multilingual content.

The report concludes that a single tool cannot be universally applicable, and that the optimal choice depends on the specific needs and maturity of the documentation environment.

**Keywords :** Technical Writing, Content Management Systems (CMS), Documentation Tools, Structured Authoring, Oxygen XML Editor, Microsoft Word, Translation Management, Tool Adoption, Survey Study, Technical Communication



## Acknowledgements

We would like to start by thanking Ms. Marie Girard for giving us the opportunity to work on this research article and her guidance throughout its course.

We would also like to thank the attendees of the colloquium on February 21st, where we had the opportunity to present our poster and engage in insightful conversations with both students and professionals.

Finally, we would like to thank every technical writer and professional who took the time to participate in our survey which was posted on LinkedIn and shared with the alumni of University Paris Cité. Their contributions were instrumental in shaping the direction and depth of our research.



Table of contents	
<b>Introduction</b> .....	<b>4</b>
<b>Methods</b> .....	<b>6</b>
1. Survey Design and Distribution.....	6
2. Colloquium Poster and Early Discussions.....	6
<b>Results</b> .....	<b>8</b>
1. The participants of our survey.....	8
2. The company of our participants.....	9
3. The tools of our participants.....	11
4. The translation management of our participants.....	12
<b>Discussions</b> .....	<b>14</b>
1. Professional Experience and Tool Maturity.....	14
2. Influence of Company Size on Tool Choice.....	15
3. Influence of Number of Technical Writing Teams.....	16
4. Documentation Team Size and Tool Scalability.....	16
5. Team Experience Levels and Technical Sophistication.....	18
6. Translation Requirements as a Predictor of Tool Complexity.....	18
<b>Conclusion</b> .....	<b>19</b>
<b>References</b> .....	<b>21</b>



# Introduction

In recent decades, the field of technical communication has undergone a profound transformation. It went from the production of static, print-based documentation to the adoption of dynamic, digital-first content strategies. This indicates a fundamental evolution in the field. Microsoft Word, once the go-to tool for all writing tasks, is now often supported or entirely replaced by powerful CMS (CMS) that support structured authoring, collaborative workflows, version control, and multi-channel publishing.

This transformation became particularly evident during the course of our internships, as all three of us used different tools. Each of us used either Microsoft Word, Oxygen XML Editor or a combination of GitHub and Markdown. These differing experiences gave rise to the question: what form would our work take if we were to exchange tools? From there, a second question naturally came to us: is there such a thing as a "the best" tool for technical writing?

As our research progressed, it became evident that identifying a single "optimal" tool was neither feasible nor practical. The additional features of these tools, essential for large organizations, would just become superfluous for the writers of smaller companies who have no use for them. This would in turn raise the entry barrier for technical writers in the smaller organization. This unnecessary complexity goes against the very principles of our profession. And so we reach a more nuanced question: when is the right time for an organization or an individual technical writer to transition from one tool to another?

The transition from conventional word processing software to CMS (software application that is used to manage, edit and create digital content), often shortened to CMS, platforms signifies more than a mere preference for particular software; it denotes a profound transformation in the manner in which technical content is created, administered, and used. As documentation becomes increasingly modular, reusable, and translation-friendly, technical communicators are expected to master a wide array of tools and methodologies, including:

- Topic-based authoring: method of writing content in small, modular chunks of information rather than in a linear format.



- Single-sourcing: using one single document or piece of content to create multiple documents.
- Structured content reuse: organized, reusable and machine-readable data.

The methodology used to present this research article to you is explained in detail before moving on to the raw data collected from the survey that will be presented in order to support the analysis. Finally, we will discuss various topics concerning the tools of a technical writer.



## Methods

The present question was addressed by the creation of a survey, the implementation of which was deemed necessary for the collection of the relevant data. The objective of this study was to collate a wide range of insights from practicing technical communicators across a variety of industries, experience levels and tool environments.

### 1. Survey Design and Distribution

Our primary way to gather data was a survey, designed to collect both quantitative and qualitative data on tool usage. The survey was developed through a collaborative process, undergoing multiple iterations during the academic year. These updates were informed by feedback from both peers and instructors, enabling the refinement of the clarity, relevance, and comprehensiveness of the questions.

In order to maximize the reach and diversity of the respondent pool, the survey was distributed through multiple channels. A big part of our strategy was the use of the professional networking platform LinkedIn. One group member proactively expanded their network by establishing connections with as many technical writers as possible. This approach enabled the dissemination of the project's findings to a more extensive and globally diverse audience of practitioners.

Furthermore, institutional connections were leveraged to enhance the response rate. With the assistance of a member of the academic staff, we made contact with former students of our program who are now working in technical writing roles. The participation of the alumni proved to be of significant value, as it facilitated the acquisition of invaluable insights from early-career professionals working in a wide range of fields and who had recently entered the sector and had experienced tool transitions firsthand.

### 2. Colloquium Poster and Early Discussions

In the context of our studies, we were given the task of creating a poster that would briefly present the research topic that we had chosen. The poster was exhibited during a student colloquium, where we were given the opportunity to



present our research projects to an audience made of other students, instructors and professional technical writers. The event provided a valuable platform for informal discussion and early feedback. We used this chance to also discuss the survey we were preparing and the questions it would contain to seasoned professionals. These interactions proved instrumental in two aspects: firstly, they assisted in the refinement of the survey and the questions it contained. Secondly, they facilitated the acquisition of preliminary insights and observations that helped shape our work and the interrogations that led it.

The methodological approach employed in this study incorporated iterative survey design, professional outreach, and early field discussions. This ensured that the findings were grounded in real-world experience and influenced by feedback from both academic and industry perspectives.



# Results

This section presents the results that we collected with our survey. We closed our survey when we had reached 38 participants. We decided this was an acceptable sample size and the participation rate had fallen drastically throughout the last couple of weeks despite our renewed requests for participation. It stood to reason that this was also close to as many responses as we could expect.

The survey was divided into four main sections:

- a section that determines the profile of the participant
- a section that focuses on the company in which the technical writer works
- a section that addresses the tools that are used by the technical writer
- a section that indicates whether the documentation is translated

These four sections aim to determine criterias that could answer our problematic: “when is the right time for an organization or an individual technical writer to transition from one tool to another?”. Each section embodies a factor we determined could influence this decision.

## 1. The participants of our survey

The first question in our survey deals with the job title of the participants to determine whether they were a manager, a senior technical writer, or an apprentice for example.

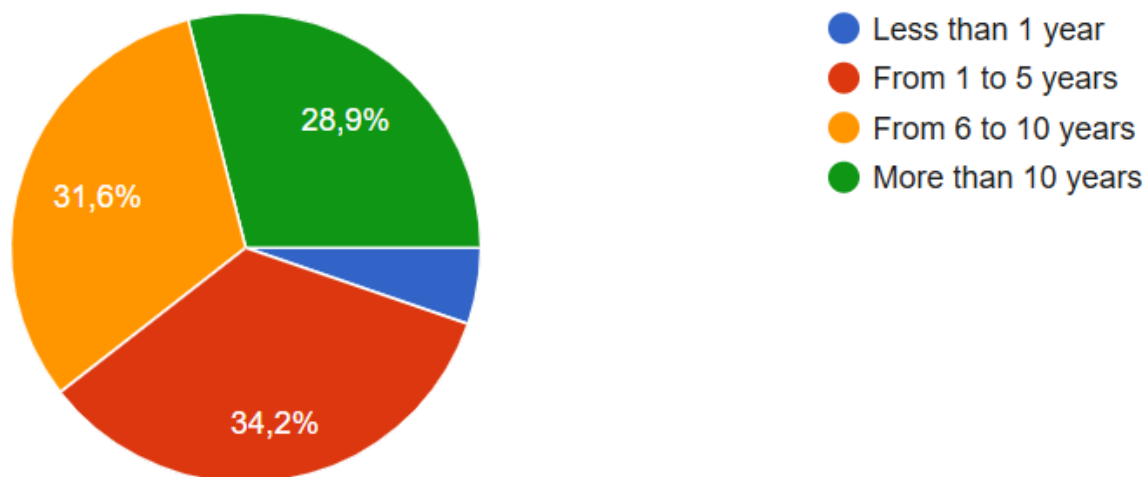
We gathered the following results for this section:

- technical writers
- technical documentation manager
- content developer
- head of learning
- staff content engineer
- content architect
- documentation lead
- senior product marketing manager
- technical information developer
- confirmed product knowledge expert
- lead content designer
- content facilitator
- knowledge base manager

In spite of the large variety of job titles that were amongst our participants, we should notice that 62,16% of the answers came from technical writers (junior and senior).



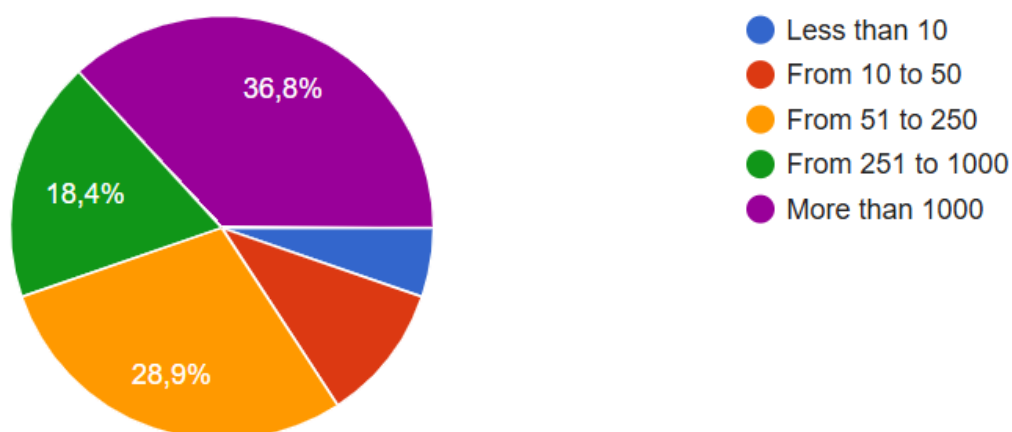
In addition to their job title, we also asked our respondents their level of experience as a technical writer. This question aims to determine the first criteria in our initial hypothesis regarding whether the experience of a technical writer impacts the tool that they are using. The results were extremely interesting for our research as they were very balanced:



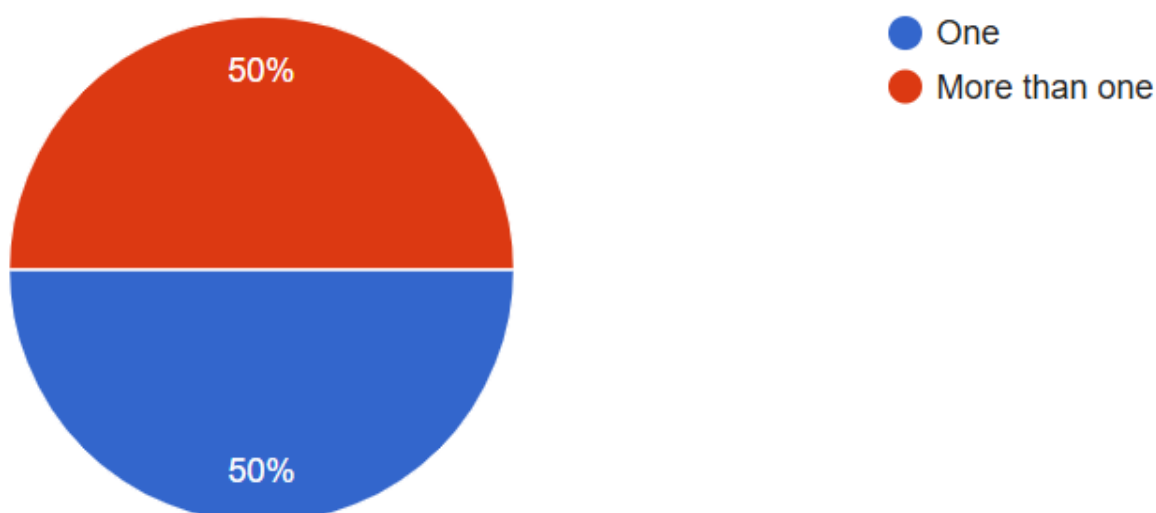
## 2. The company of our participants

This section focuses on the company of the respondents. One of our hypotheses was that the environment that the technical writer works in necessarily has an influence on the tools that they are using.

Therefore, we asked about various parameters of their company, the first being its size. As with the years of experience of the participants, their answers regarding the size of their company were quite balanced. We still observe a majority from companies that have more than 1000 employees:

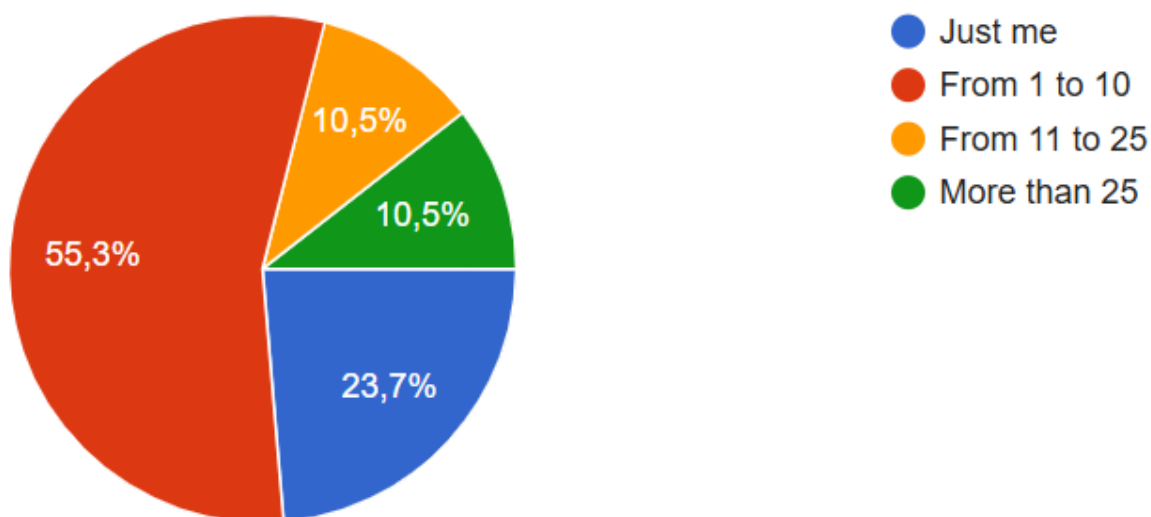


We then asked our participants to answer whether there is only one team of technical writers in the company. We obtained perfectly balanced answers as shown here:

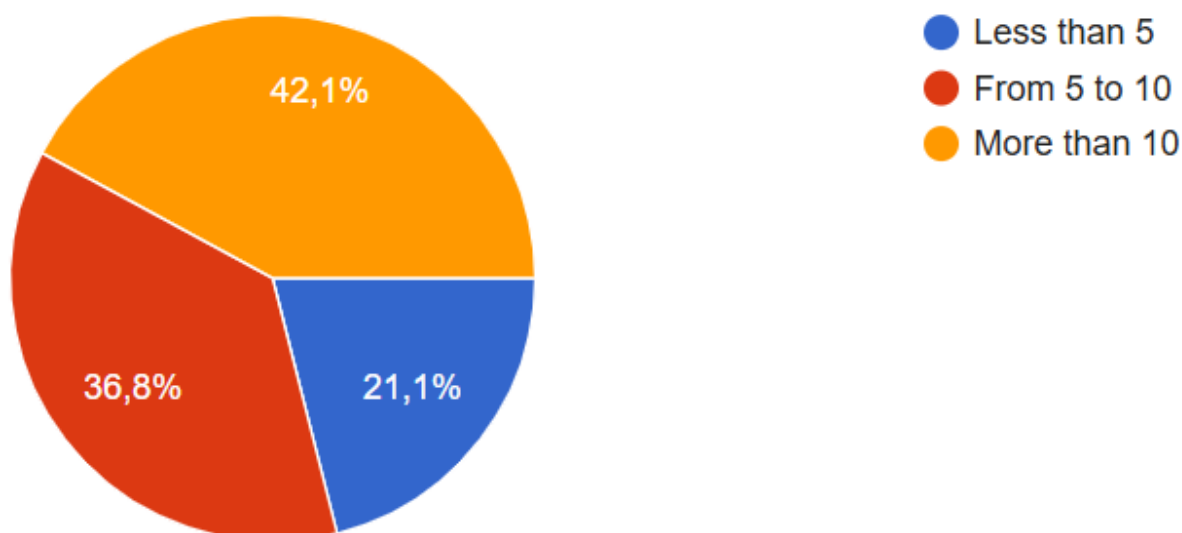




When it comes to the size of the documentation, there is a large majority that stands out. The category “1 to 10” has 55% of the results while the other three categories share the other half of the answers.



We decided that it was also relevant to gather information regarding the average experience of the team. The results were quite even but with a majority of very experienced teams.





### 3. The tools of our participants

The section that deals with the tools that are used by the technical writers was the most important section of our survey since it is at the heart of our problematic. This was one of the only questions that we left open for answer to avoid excluding tools we may not have thought of or unusual ones. All other questions were multiple choice.

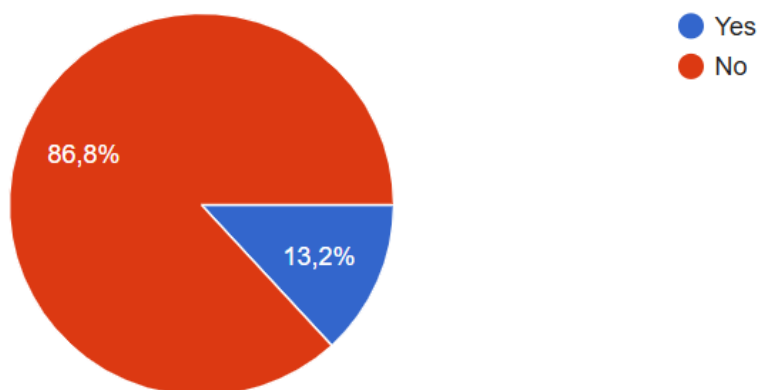
The main authoring tools for which there was in the answers were the following:

- Microsoft Word (13 occurrences)
- Oxygen XML Editor (7 occurrences)
- Visual Studio Code (5 occurrences)
- MadCap Flare (5 occurrences)
- Arbortext (4 occurrences)

The respondents also mentioned other authoring tools, some of which being complementary tools to the five listed above:

- |                    |                           |
|--------------------|---------------------------|
| ● Calenco          | ● Adobe RoboHelp          |
| ● Confluence       | ● Paligo                  |
| ● Author It        | ● IntelliJ                |
| ● Document 360     | ● Adobe Indesign          |
| ● Google Docs      | ● Notion                  |
| ● Notepad ++       | ● Heretto                 |
| ● Textpad          | ● ZenDesk Guide           |
| ● Nearandfar       | ● Some internal softwares |
| ● Adobe Framemaker | ● Salesforce Knowledge    |
| ● Quadralay        |                           |

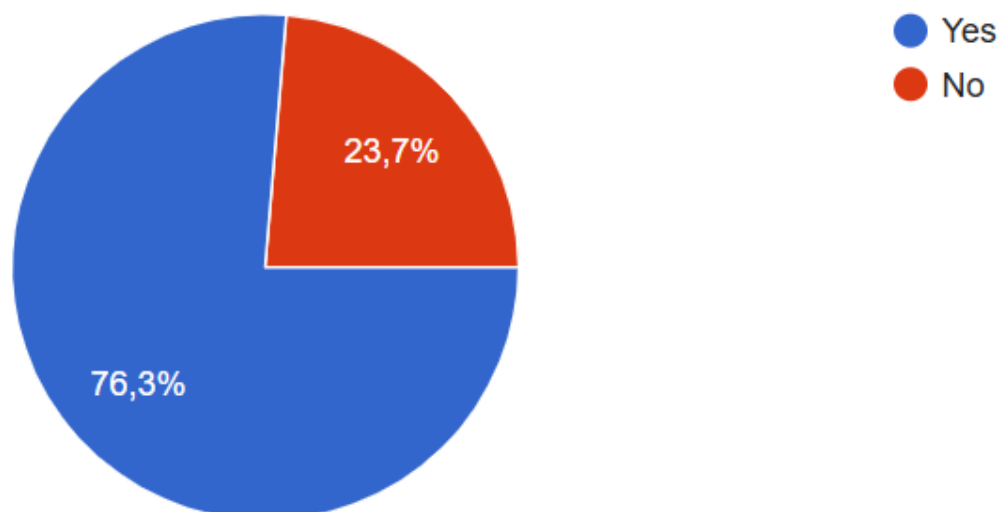
We also asked our participants whether they were currently considering migrating to another tool.



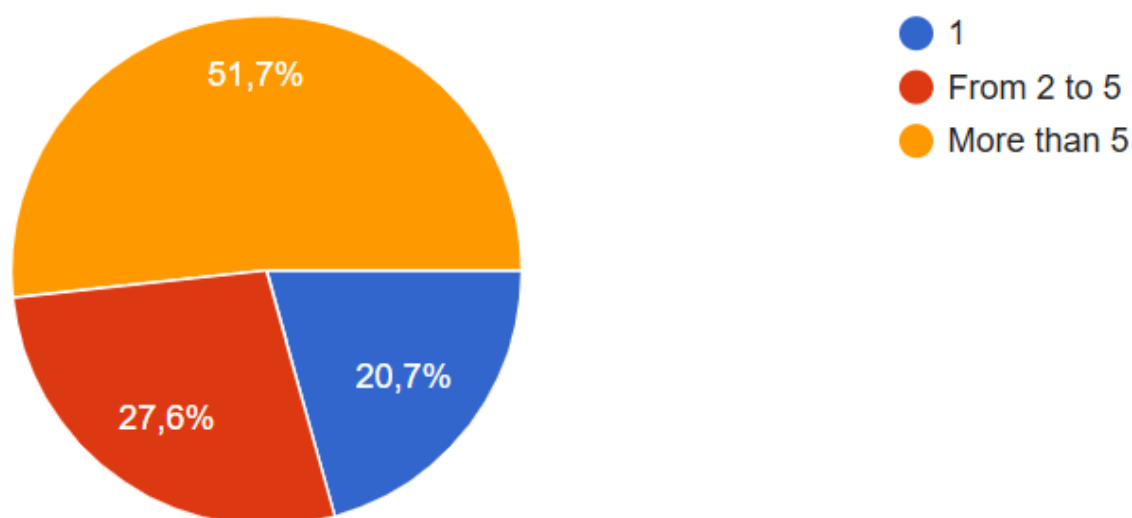
Overall, the respondents were largely satisfied with their tools.

## 4. The translation management of our participants

The question of whether the documentation of our participants is translated is important as well since the management of the translation could also be a criterion to select an authoring tool as some tools may be more adapted than others. Most of our answers indicated that the documentation was translated:



While the fact that the translation matters, the number of languages is also important because the more languages there are to translate to, the more important an efficient tool becomes. In our results, most of the documentation of our respondents is translated in more than five languages:





## Discussions

This section analyses the trends in tool usage among technical writers based on our survey. The data reveals that tool selection is not arbitrary; rather, it is influenced by a complex interplay of factors, including individual experience, team dynamics, company size, and the strategic requirements of documentation projects, such as localization or the need for translation. By analyzing these factors in relation to tool adoption patterns, we gain valuable insight into how technical writers' technology choices adapt to their content needs and organizational requirements.

### 1. Professional Experience and Tool Maturity

We noticed a strong correlation between a technical writer's years of experience and the tools they use. Early-career professionals—those with fewer than five years in the field—tend to favor user-friendly, more general-purpose tools that are either widely available or do not require a lot of practice. Tools like Microsoft Word, Markdown, Notepad++, and Jira are prevalent in this group. These tools are often sufficient for simple documentation tasks or environments with limited technical infrastructure. The writers working with these tools most likely have more liberties in how they work on their documents which makes general tools an easier fit.

Technical writers with six to ten years of experience typically have toolkits that reflect a deeper dive into structured content and more advanced authoring workflows. At this stage, tools like Oxygen XML Editor and Visual Studio Code start showing up more often, pointing to a shift toward environments that require structured documentation, integration with development teams, and more complex formatting features.

For those with over ten years in the field, the tools become even more specialized. These experienced writers usually work in well-established documentation setups and manage large, often complex content sets that need to serve different formats and audiences. Tools like Oxygen XML Editor, Adobe FrameMaker, Adobe InDesign, and Calenco are common here. They support advanced features such as topic-based authoring or single-sourcing—essential for keeping everything consistent and efficient across the board.



## 2. Influence of Company Size on Tool Choice

The size of a company often plays a big role in the tools technical writing teams use.

In smaller companies—typically those with fewer than 50 employees—budget and simplicity tend to drive decisions. Tools like Markdown, Google Docs, and Microsoft Word are popular here because they're affordable, easy to use, and great for collaboration without much setup. These companies might not have a formal documentation system in place, and technical writing is often seen as a supportive function rather than a major investment area. Because of this, less budget may be allocated towards documentation teams which limits their ability to use tools that require special licenses (as opposed to a Microsoft Office license which can be useful to a lot more people in the company).

Mid-sized companies (around 51 to 250 employees) usually go halfway between ease of use and more advanced capabilities. While Microsoft Word and Confluence are still common, we also start to see more structured authoring tools like Oxygen XML Editor and Paligo enter the mix—especially as documentation grows more complex or needs to be available in multiple languages. These organizations are often starting to take content strategy more seriously, which leads to trying out new tools, platforms and workflows.

In large companies—those with more than 1,000 employees—technical documentation is often treated as a strategic asset. It's closely aligned with broader content, development, and customer support goals. To manage this complexity, these organizations tend to rely on standardized tools with more depth that ensure consistency and scalability across teams. Tools like Oxygen XML Editor, Adobe FrameMaker, and Confluence are common choices, and it's not unusual for companies to develop their own custom CMS. These tools help tackle big challenges like regulations compliance, managing multilingual content, and keeping documentation up to date across departments.



### 3. Influence of Number of Technical Writing Teams

The way a company is structured—particularly how many technical writing teams it has—can have a big impact on the tools it uses.

In organizations with a single technical writing team, it's common to see standalone tools like Microsoft Word, Markdown, or Oxygen XML Editor. The choice usually comes down to what the team is familiar with and how complex the documentation needs are. These tools are perfectly capable of producing high-quality content, but they can start to show their limits when it comes to collaboration or sharing content across other departments.

In contrast, organizations with multiple technical writing teams often face a different set of challenges—namely, how to keep documentation consistent, collaborative, and accessible across various groups. These teams might be working in different departments, regions, or even time zones, so the need for unified workflows becomes much more important. That's why tools like Confluence, Zendesk Guide, and Heretto are so widely used in these environments. These platforms were designed with scale and coordination in mind.

They support features like content reuse, so teams don't have to start from scratch every time they create a new article or guide. Real-time editing makes it easier for writers to collaborate without stepping on each other's toes, while centralized style guides help ensure a consistent voice and tone across all documentation. Localization workflows are another major plus—especially for global teams that need to deliver content in multiple languages without duplicating effort. For companies operating at this level, these tools aren't just nice to have but an essential part of keeping everything running smoothly.



## 4. Documentation Team Size and Tool Scalability

The size of the documentation team also seems to play a big role in which authoring tools get chosen.

Solo technical writers—often working in startups or on small product teams—tend to stick with tools that they can manage independently. Simplicity and speed are key, so tools like Markdown, Microsoft Word, and Google Docs are especially popular. They're easy to use, require little to no setup, and allow writers to focus on the content itself without getting bogged down in complex publishing workflows or tool administration. In these environments, documentation is often seen as a supporting resource rather than a core product, so it's harder to justify spending on expensive licenses or enterprise-grade tools.

For small to mid-sized teams—typically made up of 2 to 10 writers—the tool landscape starts to get more varied. These teams are still looking for flexibility, but they're also starting to feel the pressure of scaling: more content, more collaboration, and often more structure. As a result, it's common to see a mix of tools in use. Microsoft Word might still be in the mix, but you're also likely to find more specialized solutions like Oxygen XML Editor, Adobe FrameMaker, and Confluence. The choice usually depends on the nature of the content—whether it needs to be highly structured for things like technical specs and compliance, or more informal and adaptable for internal guides and knowledge sharing.

Larger documentation teams—typically those with 11 or more writers—often face challenges that smaller teams simply don't have to worry about. At this scale, managing content becomes a much more complex operation, and the need for robust, end-to-end systems becomes clear. These teams are usually dealing with multiple products, multiple audiences, and often multiple output formats. As a result, they tend to adopt more comprehensive tools that support modular content creation, automated workflows, and multi-channel publishing.

Platforms like Paligo, Adobe FrameMaker, and DITA-based (DITA is an XML standard for authoring, publishing, and producing technical documents) solutions such as Oxygen XML Editor are especially common in these environments. These



tools are built for scale—they offer features like version control (version control is a system that helps you track and manage changes in your documents over time), topic-based authoring, and content reuse, which help large teams stay consistent and efficient even as their documentation needs grow. By structuring content in reusable modules, writers can update information in one place and have those changes reflected across dozens of documents. And with built-in support for publishing to formats like HTML, PDF, and help portals, these tools help streamline delivery to diverse audiences. For large teams, these kinds of systems aren't just helpful—they're foundational to keeping documentation organized, up-to-date, and aligned with product development cycles.

## 5. Team Experience Levels and Technical Sophistication

A team's overall experience level often plays a big role in shaping its tools—especially when it comes to how technical or developer-oriented those tools are.

Teams with less than five years of average experience tend to gravitate toward more user-friendly, visually guided tools. These platforms are typically easier to learn and don't require deep technical knowledge, making them a great fit for fast-paced or high-turnover environments. Tools like Google Docs, Markdown editors, or even basic CMS platforms allow writers to get up and running quickly, which is crucial when onboarding new team members or when documentation needs to move fast alongside product development.

On the other hand, teams with a decade or more of collective experience often lean into more advanced, structured toolchains. These teams are usually more comfortable working in technical environments and may be deeply integrated into the software development lifecycle. It's not uncommon to see setups that include Oxygen XML Editor, Git, Jenkins, and other developer-centric tools. These platforms support more sophisticated workflows, including automated builds, version control, modular content reuse, and continuous integration. In these environments, documentation isn't just a byproduct—it's an integral part of the product development pipeline, and the toolset reflects that level of integration and complexity.



## 6. Translation Requirements as a Predictor of Tool Complexity

Whether or not a company translates its documentation is often one of the clearest indicators of the kind of tools it will use—especially when it comes to structure and scalability.

Organizations that publish content in multiple languages usually need tools that support structured authoring, translation workflows, and localization management. In these cases, consistency across languages is key, and so is the ability to reuse content efficiently. That's why platforms like Oxygen XML Editor, Confluence, and Zendesk Guide are frequently chosen. They offer features that simplify translation handoffs, help teams manage multilingual versions of the same content, and maintain a unified voice across different regions and audiences.

On the flip side, companies that don't translate their documentation tend to have more flexibility in their tool choices. Without the added complexity of localization, they can prioritize other needs—like visual design or ease of use. As a result, you'll sometimes see older or more design-oriented tools in use, such as Adobe InDesign. These are especially common when the documentation is tightly integrated with marketing, or when it's primarily shared as static files like PDFs. In these cases, the focus is often on presentation rather than structure or scalability.



## Conclusion

This study highlights the complex landscape of tools used by technical writers and the diverse factors that influence their adoption. The survey and analysis conducted revealed that there is no universally applicable tool that is considered to be the most effective for technical writing. Instead, the selection of the appropriate tool is highly contextual, influenced by individual experience, the size and structure of the team, the scale of the company, and the strategic requirements of documentation, such as localization and translation.

The findings of this study indicate a discernible trend: as the complexity of documentation increases, the toolset required to manage it also grows in complexity. It has been observed that less experienced writers and smaller teams tend to favor more accessible tools such as Microsoft Word or Markdown. By contrast, larger organizations and more experienced professionals often employ robust platforms such as Oxygen XML Editor, Adobe FrameMaker, or specialized CMS solutions. Moreover, the existence of translation workflows and multilingual content has been demonstrated to result in a considerable increase in the necessity for structured authoring and scalable systems.

The transition from conventional word processors to sophisticated CMS is indicative of more extensive changes occurring within the domain of technical communication. In the contemporary landscape, technical writers are required to navigate not only the intricacies of the tools themselves, but also the broader ecosystem of documentation strategy. This requires a delicate balancing act, encompassing factors such as usability, scalability, and collaboration. Rather than pursuing a solution that is applicable in all situations, technical communicators ought to undertake a process of evaluation of tools with a view to ascertaining the suitability of each for the specific goals to be achieved, the contexts in which they are to be employed, and the evolving demands of their documentation processes.



## References

Adele, Toyibat. « The Importance of Version Control in Technical Writing ».

*WriteTech Hub*, 26 March 2025,

<https://writetechhub.org/version-control-in-technical-writing-2/>. Accessed June 29, 2025.

« ClickHelp Software Documentation Glossary Term: CMS ». *ClickHelp*,

<https://clickhelp.com/software-documentation-glossary/cms/>. Accessed June 29, 2025.

« ClickHelp Software Documentation Glossary Term: DITA (Darwin Information Typing Architecture) ». *ClickHelp*,

<https://clickhelp.com/software-documentation-glossary/dita/>. Accessed June 29, 2025.

« ClickHelp Software Documentation Glossary Term: Single-Sourcing ». *ClickHelp*,

<https://clickhelp.com/software-documentation-glossary/single-sourcing/>. Accessed June 29, 2025.

Stojović, Arso. « Structured Content Guide: Reusable Blocks for Scalable Sites ».

*BCMS*, <https://thebcms.com/blog/structured-content-guide>. Accessed June 29, 2025.

*Topic-Based Authoring Explained* | *Author-It*. 25 May 2022,

<https://www.author-it.com/blog/topic-based-authoring/> Accessed June 29, 2025.