

The never-ending debate: DITA or...?

Lisa MAUCHOSSÉ, Tasmiya NOOR-MOHAMMED, Alima SANOGO

Paris Cité University, July 2023

The Darwin Information Typing Architecture (DITA), introduced in 2001 at IBM and launched to the public in 2005, has set new standards in technical documentation practices, meeting the needs of both companies and technical writers. Nonetheless, as the field of technical writing has evolved over the last two decades, it is essential to assess whether DITA continues to live up to its reputation as a universal solution in the face of the emergence of other solutions.

To get to the bottom of this, we conducted a survey among 34 technical writers around the world. The aim of our research is to analyze the current relevance of DITA in modern technical documentation practices, examining its advantages and limitations, and exploring alternative solutions. The article looks at important considerations for the adoption of DITA, such as content strategy, cost analysis, and tool selection. In addition, it addresses the growing popularity of alternative solutions such as Markdown, which offer simplicity and ease of implementation.

Keywords: authoring tool, DITA, Markdown, technical writing, documentation

Acknowledgements

We extend our warmest gratitude to Marie Girard, Laura Baribaud and Vishal George Palliyathu for their guidance and support. We would also like to thank all the participants of the survey for their time and contribution, without whom this study would not have been possible.

Introduction

Since its inception in 2001, the Darwin Information Typing Architecture (DITA), an open XML-based standard for authoring and publishing structured content, has had a profound impact on the field of technical documentation. It revolutionized the way information is structured, managed, and published.

According to the article *DITA: A Universal Solution?* written by Marion L., Catherine S., Jean-Michel V.D.B., and Laureen Z., we noticed that in 2015, DITA was widely adopted by numerous organizations across various industries.

However, since then, the technical writing landscape has evolved. New authoring tools and approaches have emerged, and existing tools have been updated to meet the needs of technical writers and compete with DITA editors. It is important to note that DITA itself is not an authoring tool but rather a framework for structuring and organizing information. Nevertheless, selecting a documentation authoring tool is part of the DITA adoption process.

As of 2023, DITA has been in existence for two decades, and eight years have passed since the publication of the aforementioned article. Considering this timeframe, it becomes pertinent to examine the current state of DITA and assess whether it still retains its universal appeal as the go-to choice for technical documentation. Furthermore, it is crucial to evaluate whether DITA continues to meet the ever-changing demands and challenges faced by technical communicators.

Our research article aims to critically analyze the current relevance of DITA in the context of modern technical documentation practices. By exploring the advantages and limitations of DITA, we seek to shed light on whether DITA remains the universal choice or if it faces significant challenges that might prompt a shift towards alternative solutions.

To understand the evolution and current state of DITA, we surveyed 34 technical communicators worldwide. The questions of the survey focused on aspects such as the use or non-use of DITA in their organizations, the advantages and disadvantages of DITA, and alternative authoring environments. Through our survey, we also seek to gain insights into the migration experiences to DITA or other authoring solutions.

Before we discuss the results of our research, we will provide a detailed description of the methodology employed. Additionally, we will share the raw data collected from the survey questions and answers to ensure transparency and comprehensiveness in our analysis.

Methods

For our research, we took a multi-faceted approach that included three different sources of information.

Firstly, we designed a survey on authoring tools that was inspired by a similar survey employed in the research study titled *DITA: A Universal Solution?*. Building on this foundation, we added new questions to analyze additional elements that may influence technical writers and their use of authoring tools.

Secondly, we searched for articles on related subjects to gain deeper insights into our topic.

Lastly, we contacted several professionals from the technical writing field to ask them more questions. To achieve this, we used LinkedIn to connect with fellow technical writers. We also endeavored to contact the authors of the articles referenced in our research. As a result, we successfully conducted interviews with two individuals:

- A technical writer currently working in a software company with expertise in user assistance.
- A technical writer with 20 years of experience in technical writing, information architecture, content strategy, and copywriting.

Results

This part presents the survey results obtained on July 18th, with a total of 34 participants. We decided to divide the survey into five main sections:

- The first section aims to understand the working environments of all the participants.
- The second section focuses on the writing habits of all the participants.
- The third section is specifically presented to the participants who have experienced a migration during their career, to see how and why the technical writing field evolved.
- The fourth section gathered the opinions of participants who had never undergone a migration, seeking their views on possible future migrations.
- The final section allowed everyone to add additional comments.

The profile of the participants

In the first section, we asked the participants about their job titles at their company, which are:

- Technical Communicator
- Technical Writer (Junior and Senior)
- Lead Technical Writer
- Knowledge Specialist
- CEO
- User Assistance Development Manager
- User Assistance Developer
- User Assistance Leader
- UA Team Manager
- Information Architecture & Strategy
- Senior Director of Content Platforms
- Technical Documentation Manager
- Team Leader
- Information Architect

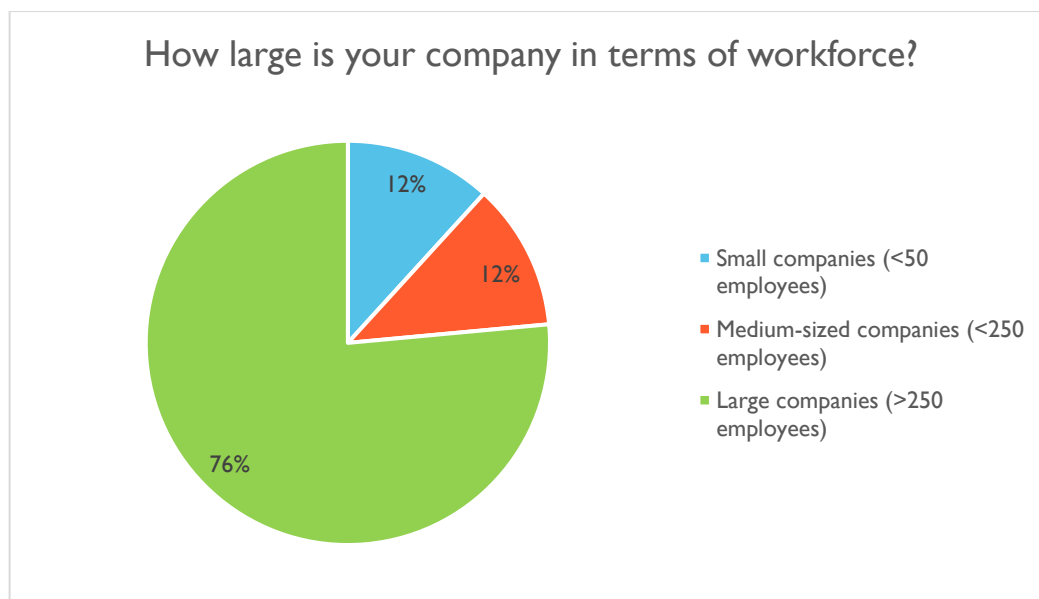
The majority of respondents have technical writing backgrounds.

We also asked about their current industries. Their answers encompass a diverse range of sectors, including:

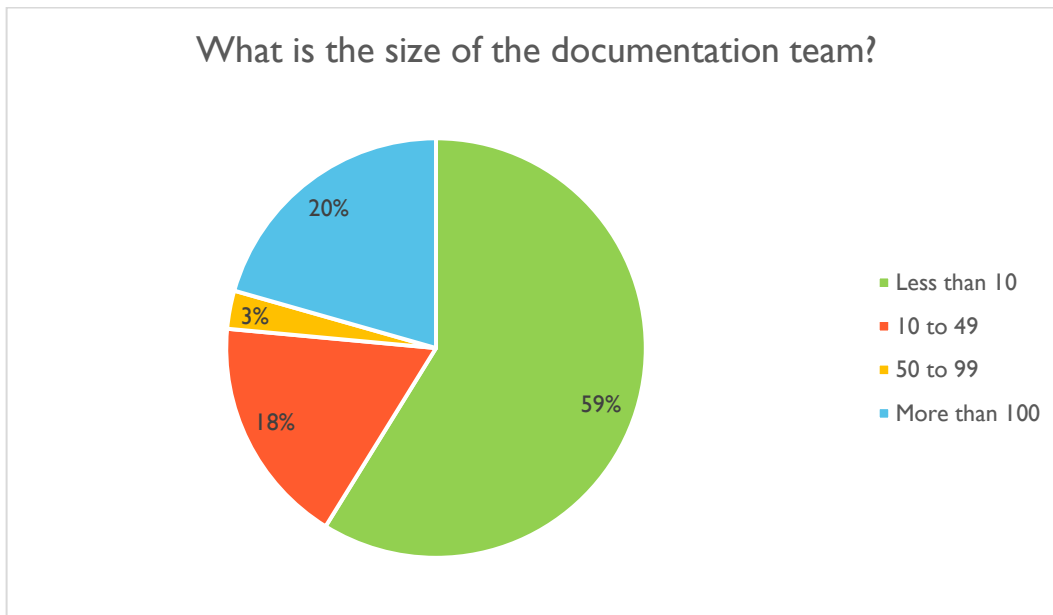
- Aeronautics
- Cloud computing
- Electronic Signature
- Enterprise Telephony
- General Engineering
- Information Technology
- Localization
- Medical
- Microcontrollers
- Pharmaceutical
- Semiconductor

The most prevalent industry in the survey is the software field.

Regarding company size, a significant portion (approximately 75%) work for large companies with over 250 employees.

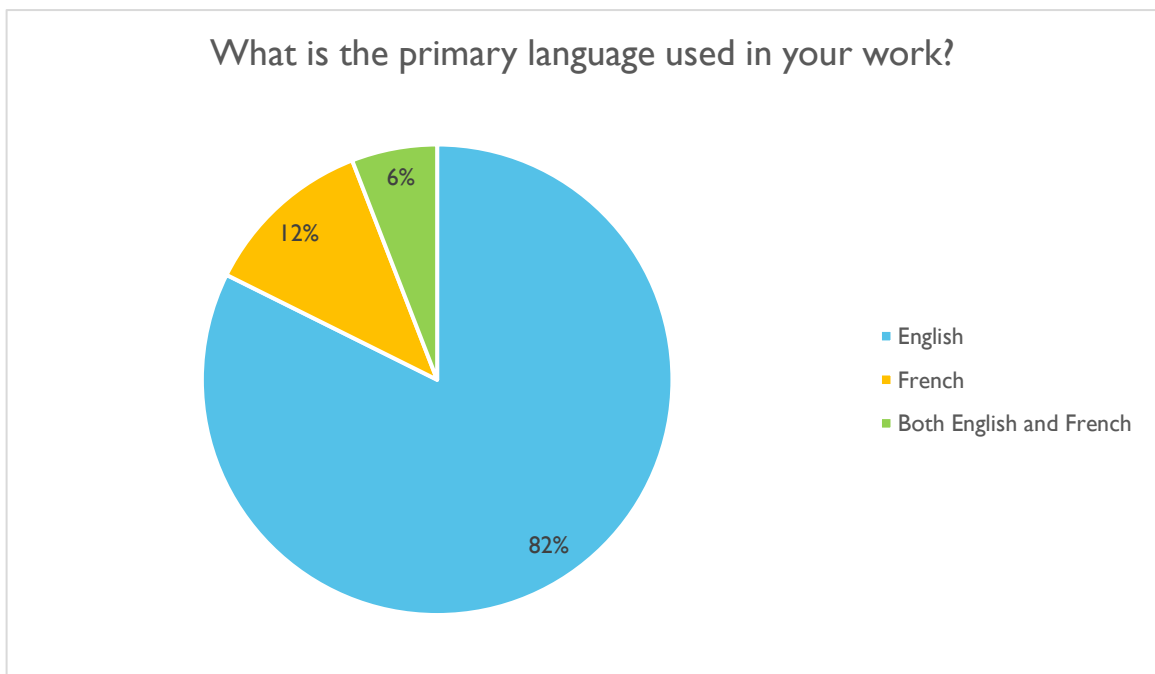


In terms of documentation team size, the results varied, but the most common case was teams with 10 or fewer writers, involving 20 participants.

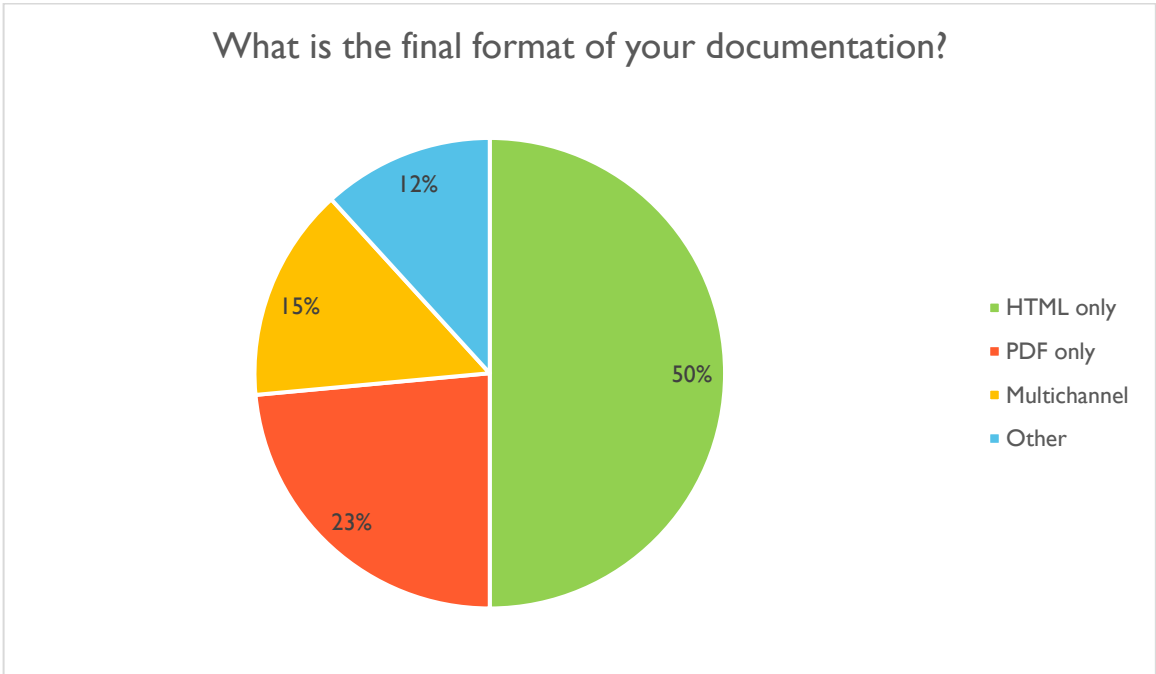


Geographically, about 80% of the participants are from Europe, while the remaining 20% consist of 4 participants from North America and one participant each from Africa and Oceania.

English is the primary language for most of the participants' documentation, followed by French.



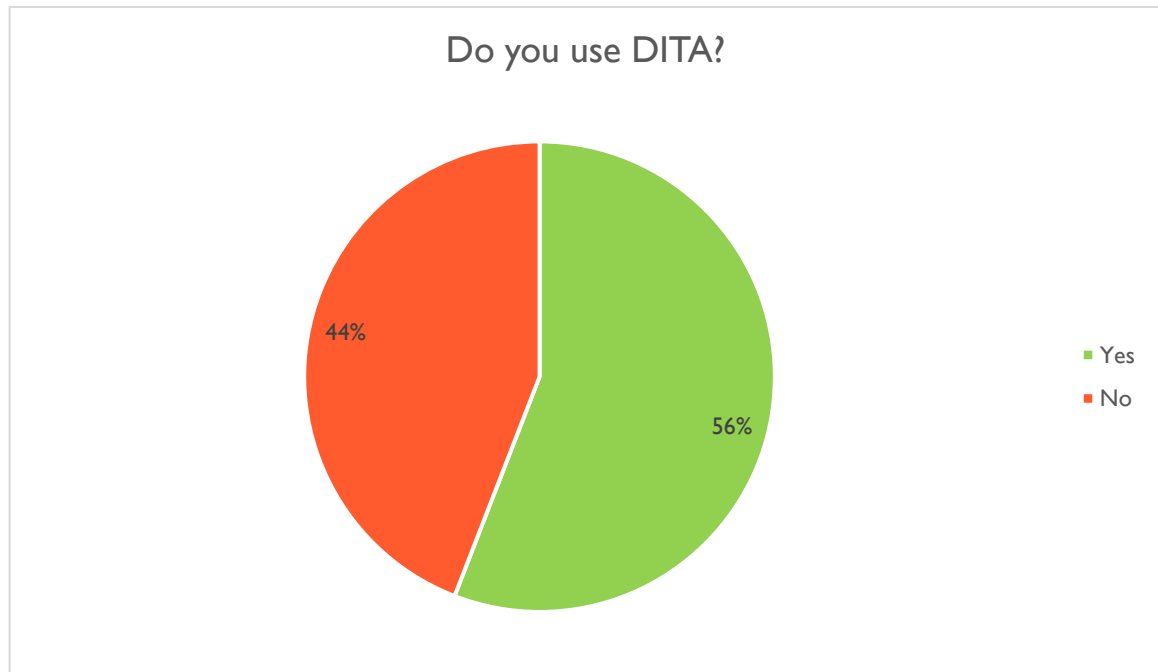
The final question in this section focused on the output format of the participants' documentation. The most frequent format was HTML alone, with 17 answers.



The writing environment of the participants

Moving on to the second section, participants' writing habits.

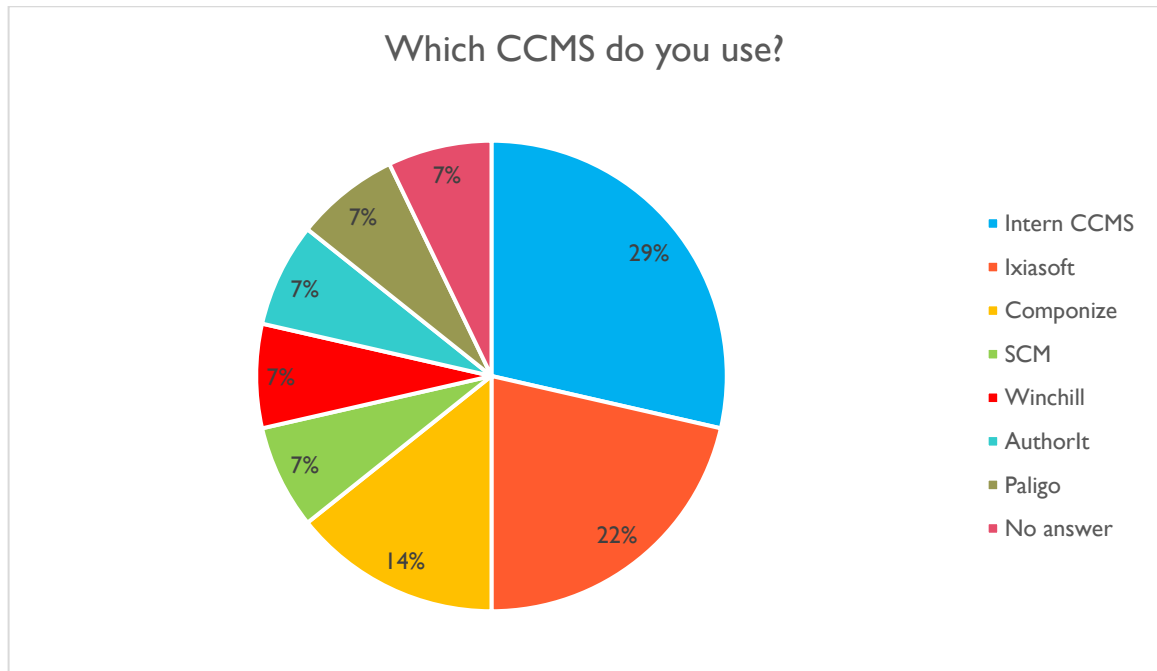
We observed that 19 participants (or 56% of the participants) use DITA, while 15 participants (or 44% of the participants) don't follow this norm in their documentation.



When asked about the respondents' current authoring tool, the most frequently mentioned software was Oxygen, specifically its XML Editor and XML Author, with 18 mentions. The participants also mentioned the following authoring tools:

- Visual Studio Code
- Microsoft Office
- Zendesk
- Confluence
- Angora
- Arbortext
- Madcap Flare
- AuthorIt
- Adobe FrameMaker
- Paligo
- SDL
- Jekyll
- Doxygen
- Slate
- Adam Author
- IXIASOFT with Oxygen embedded

When asked "Do you use a CCMS?", the results show a relatively equal split, with 16 participants using a CCMS versus 15 participants who don't use one. We can see mentions of services like Componize, Confluence, Paligo, Ixiasoft, or an in-house CCMS.



The mentioned tools in response to the question "Which other authoring environments have you experienced?" were the following:

- Microsoft Office (Word and PowerPoint)
- Google Docs
- XML/Coding software (XMetal, Arbortext, SDL Trisoft, Notepad++)
- Adobe FrameMaker and RoboHelp
- Frontpage

In response to the question about the pros and cons of their authoring tools, participants provided insights for different types of editors. According to them:

- Editors compatible with the DITA standard make documentation creation more efficient and consistent, for example with the reuse option. This has led some people to move away from tools such as FrameMaker that lack this feature. However, adopting the DITA standard requires substantial investment, as it can be complex to learn, especially for documentation layout, and can be expensive in the long run.
- Confluence and other "WYSIWYG" software are more accessible to technical writers of all levels but can be more difficult to use with large amounts of documentation.
- Docs-as-code tools were acknowledged to require some learning, both for the software and for publishing on platforms like Git. However, with a well-established environment, they can efficiently manage large volumes of documentation.

- Markdown can be a good and easy-to-use alternative, particularly suitable for small enterprises.

As a prelude to the next section, we asked participants if they had ever been through an authoring tool migration. Out of the total participants, 10 individuals had experienced a migration, while 24 had not. Those who had experienced a migration were redirected to the dedicated section, while the remaining participants were redirected to the section seeking their opinions and their company's stance on a potential authoring tool migration.

For the participants who were involved in a migration

Among the 10 participants involved in a migration, 6 initiated it. They initiated the migration after discussing the idea with the executive team. Some participants took the same migration path, such as making a presentation to convince their company to migrate its documentation. Others explained their migration progress, whether through a conversion service or a manual conversion.

The migration duration varied from 3 months to a year for different participants.

During this period, participants from diverse roles, ranging from technical writers to managers, were involved. Notably, most of those who initiated the migration held managerial positions within their companies.

Before the migration, participants mentioned using the following software such as:

- FrameMaker,
- Word and Google Docs,
- Frontpage and Visual Studio Code

When we asked if the participants had technical difficulties, 7 participants answered "Yes". For example, a participant stated that: "The integration between Paligo and Freshdesk was very weak. We have to make a custom script to change the output from Paligo to better match the rendering capabilities of Freshdesk." In other answers, participants also mentioned the lack of knowledge and support during the migration, working in parallel between two editors, and the formatting of the documentation.

Regarding changes in habits post-migration, 6 of them answered "Yes". They described the learning process required to effectively use the new tool, and the adjustments made by their documentation team, with DITA and reuse content, for example.

In response to the question about how colleagues and contributors reacted to the migration, most participants reported positive responses. However, one participant explained that their colleagues were "at first hesitant, but then they relished the challenge and helped solve the integration problems."

5 participants stated that their company had produced an evaluation report after the migration, all of which showed positive results, like cost savings. For instance, one participant explained: "We were expected to increase our overall productivity by 25%." This was the condition set by their company to

implement the authoring tool. According to the participant, this goal has been successfully achieved by the end of the first half of the year.

Participants highlighted various benefits from the migration including improved output, enhanced features, increased consistency in documentation, and an overall more professional tool.

When we asked if participants had any reservations about their migration, one participant answered: "I wish I had better researched and tested the integration between our help center platform Freshdesk and the tool, Paligo, before signing the contract. It was very stressful at first to find the problems and then solve them after committing to the expensive tool." Another one argued that "companies should allocate resources for training and provide support to ensure a smooth transition", and that "The migration process involves cleaning up and restructuring legacy content to fit the XML structure, which is time-consuming and resource-intensive".

When asked if they experienced a change in productivity, the majority of participants in this category reported a noticeable increase in productivity as their writing process became more efficient.

Regarding the migration cost, two tendencies in the answers were observed. While some respondents were unsure of the cost, others provided answers that ranged from relatively inexpensive, especially for those migrating to more budget-friendly software solutions, to significantly expensive in terms of money and time.

For the participants who weren't involved in a migration

We will now focus on the participants who did not experience a migration.

To the question "Does your company intend to migrate towards an authoring tool?", only 2 participants answered "Yes". One participant's company is in the process of gradual migration to Astoria, and his team will soon follow suit. The other respondent mentioned a migration to Madcap, driven by positive feedback and popularity.

When asked, "Do you wish to migrate towards an authoring tool?", 4 participants expressed their wish to migrate. In comparison to the answers to the previous question, the difference isn't substantial.

In the comments, we find the previously mentioned tools and Paligo. One of the participants pointed out that he wanted "anything but DITA".

When we asked about the possible benefits of migration, three tendencies emerged in the answers:

- The need to migrate towards an easy-to-use platform,
- The need to use a more professional tool,
- The need for better content management and better structure in the company's documentation

Additional comments

At the end of the survey, the participants provided valuable comments on the subject:

- One participant highlighted the use of the Docs-as-Code approach, suggesting it as a viable alternative for technical documentation.
- Another participant explained that, to this date, there is probably no authoring tool supporting both DITA and Markdown.
- Several participants insisted on DITA and its problem with flexibility.
- A participant suggested open-source tools as a cost-effective and cohesive alternative for technical documentation. However, the main challenge is that open-source tools require you to learn how to use them, and they can be difficult and time-consuming to deploy.

Discussion

DITA's Manifold Benefits

As shown in the results, the benefits of DITA that often come up are related to content reuse and single sourcing.

When defining "content reuse" and "single sourcing", it's important to note that content reuse involves the deployment of identical content across various sections of a document set with minimal adjustments. On the other hand, single sourcing refers to the creation and management of content to generate multiple outputs or document formats from the same source content.

Content reuse and single sourcing are some of the key selling points of DITA. Teams that have migrated to DITA testify to the benefits of reducing the creation of new content and exploiting existing content in a variety of ways. DITA's reuse capabilities are present at many levels, from maps to topics to elements.

With DITA, companies can write content once and publish it across different channels and formats without having to rewrite it. This approach enables efficient content delivery and scalability without a linear increase in resources, resulting in improved productivity and reduced duplication.

In addition, the process of maintaining and updating information is streamlined. When a reused subject is modified in one place, the changes are automatically propagated to all instances where it is reused. This single sourcing method saves time and effort in managing content updates.

Reusing content also ensures consistency in documentation. This practice reduces the risk of ambiguous interpretations and helps maintain a consistent tone and message throughout the content.

Last but not least, DITA's content reuse capabilities considerably reduce translation costs. With less content to translate, especially when multiple languages are involved, organizations can achieve substantial savings. This cost-saving aspect further strengthens the business case for implementing structured authoring projects.

This argument about translation leads us to the next advantage of DITA, which is efficient localization. Localization involves adapting content to targeted markets in an increasingly global context, and DITA offers features that effectively support this process.

- The localization process is based on several key principles:
- Filters and segmentation involve selecting and then extracting the content to be translated from the verbose XML or DITA content. This extracted content is then broken down into a list of contextualized segments, facilitating targeted translation.
- Translation memories (TMs) play a vital role. Since content can be reused effortlessly with DITA, it only needs to be translated once, allowing translations to be better leveraged through a translation memory. Translation memories (TMs) store previously translated segments and their context. The use of existing translations saves time and effort, as only updates are translated, and previous translations are reused.
- Terminology databases can serve as centralized repositories for key terms, their translations, and context. They ensure the consistent use of terminology throughout the localization process. It's important to point out here that specialized glossary headings (glossentry, glossgroup) exist in DITA, but they are insufficient when it comes to multilingual support. In his post [How to leverage multilingual terminology in DITA](#), Jourik Ciesielski proposes a solution, which is to create a custom terminology topic. The following image is an example of a topic used for terminology purposes:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dita PUBLIC "-//OASIS//DTD DITA Composite//EN" "dibase.dtd">
<dita>
  <topic id="topic_yvp_yxb_flb">
    <title>Terminology</title>
    <body>
      <p id="1">
        <term xml:lang="en-US">File Manager</term>
        <term xml:lang="es-ES">Administrador de archivos</term>
        <term xml:lang="fr-FR">Gestionnaire de fichiers</term>
        <term xml:lang="ja-JP">ファイル管理</term>
        <term xml:lang="pt-PT">Gerenciador de arquivos</term>
      </p>
    </body>
  </topic>
</dita>
```

Ciesielski's solution consists of creating a topic with the "title" and "body" tags. The "p" elements are used as containers for the terms themselves, whose language is determined by the xml:lang attribute. To add a new word, the technical writer can add a new "p" element with a new ID.

- Translation workflows describe the sequence of tasks involved in translating and reviewing files, thus ensuring a systematic and efficient localization process.

As DITA can be integrated with translation management systems (TMS), some companies are turning to these systems to optimize localization, especially when managing large volumes of files. Translation management tools improve translation efficiency by automating workflows, contributing to the overall effectiveness of localization efforts.

Overall, DITA's structured approach, content reuse features, single sourcing, and integration with translation management systems all contribute to better source management. As a direct result, organizations can efficiently

create, update and manage their content. A coin has two sides similarly DITA has its advantages but also its downsides.

Downsides of DITA

While DITA offers advantages such as facilitating content reuse and ensuring consistency in different documents, it is important to acknowledge its limitations. Several key comments in our survey and answers during interviews shed light on the drawbacks associated with DITA.

To begin, we would like to address the issue of non-flexibility that has been highlighted by participants in our survey:

- One respondent expressed their concern by stating: "While DITA ensures the consistency in different documents, we think it does not offer enough flexibility."
- Another participant mentioned: "DITA is limiting, persnickety, and too expensive to maintain. By not using DITA, we would gain flexibility, ease of use, and monetary savings."

In support of these observations, Michael Andrews, Content Strategy Evangelist at Kontent.ai, explains in his blog *Why You Should Ditch DITA and Adopt Headless Instead that DITA* applies a rigid framework to content structuring.

Companies often require a variety of content structures that may not fit within DITA's predefined information types. DITA's restrictive framework determines what information can or cannot be included within a particular topic type. As a result, they face the dilemma of either forcing their varied content into limited categories that don't align with their needs or going through the complicated process of creating custom information types within DITA's complex document typing system.

Another comment in our survey reinforces the challenges of customization within DITA.

When asked about the difficulties experienced during the migration, a participant underlined the following point: "DITA stylesheet may be complex to define for some specific layouts."

Although DITA allows content to be published in different formats, such as PDF and HTML, the default output options provided by DITA Open Toolkit (DITA-OT) are limited and simplistic in terms of visual design. Companies willing to achieve a more appealing and customized appearance that aligns with their brand identity, need additional investment.

This involves creating templates using XSL-FO for print formatting and CSS for web output which therefore means that companies have to invest time in the process.

Creating stylesheets requires a programmer or technical writer with programming skills to handle output formatting and automate processes. If there is no technical expert within the team who can develop DITA-OT plugins and templates, the company may need to hire a person with the required skill set. Although this approach can be expensive, it is necessary to ensure high-

quality output formatting that meets the company's specific requirements and visual guidelines.

In short, to get the output they want, companies need to invest not only money but also expertise and time.

Another significant drawback of DITA that emerged from our interviews with a DITA information architect and a technical writer was its steep learning curve.

Acquiring a comprehensive understanding of DITA demands a significant investment of time. DITA's learning curve is significant.

While theoretical courses in college, in webinars, or elsewhere can provide some initial understanding, it is through practical application and hands-on experience that one truly grasps the intricacies of DITA.

Writing in DITA differs significantly from composing a Word document. In DITA, each page functions as a standalone entry point. So, this requires technical writers to consider the user's search behavior and the use of search engines like Google. Unlike traditional linear browsing, users rely on search mechanisms to land on relevant pages and navigate accordingly.

The interviewees emphasized that learning DITA is an ongoing process that continues throughout one's career, and mastering DITA demands a diverse skill set.

Consequently, the challenges primarily lie within the realm of DITA, with its extensive learning curve being a significant obstacle to overcome.

Considerations for Adopting DITA

When considering the adoption of DITA, companies need to carefully weigh its benefits and downsides. According to the *2020 DITA Satisfaction Report*, 83% of respondents faced challenges during DITA adoption. Common challenges reported by technical documentation teams include incoherent content strategy (61%), cost of adoption (31%), inability to prove return on investment (41%), and technology/software selection and implementation (38%). Let's take a closer look at each of these challenges.

When migrating to DITA, a company's content strategy should focus on aligning content creation, organization, reuse, customization, localization, and governance with its goals and requirements. It plays a strategic role in managing content within the DITA framework, necessitating meticulous planning and coordination, especially when transitioning from a different format and structure to DITA. Although this process can be time-consuming and resource-intensive, documenting content strategy guidelines is essential for successful implementation.

Cost is a further challenge to the adoption of DITA. The steep learning curve requires staff training in the structure and new tools. Additional expenses may include acquiring DITA editors, hiring professionals to manage information structure types or stylesheet customization, and potentially investing in a content management system (CMS) for content management purposes.

It is crucial to consider the Return on Investment (ROI), which calculates the monetary value of an investment compared to its cost. Companies should assess the long-term ROI by evaluating the benefits and costs associated with DITA adoption, taking into account their specific documentation requirements. This analysis helps ensure informed decision-making regarding the adoption of DITA.

Finally, choosing the right tools and software is essential. With many DITA editors available, organizations need to carefully assess their options. Key factors to consider are collaborative authoring capabilities, localization support, and integration with CMS.

A documentation tech lead from a cloud computing company commented in our survey: "Please don't restrict yourself to DITA and XML-based solutions. Docs-as-code is an alternative to all these; it might be a bit lesser known than DITA, but it can be much more powerful – and less cumbersome – than DITA." Some of the participants also mentioned the use of the markup language Markdown. This brings us to the question of whether DITA is challenged.

DITA Challenged?

Markdown is used in software documentation, and the preferred output formats are readme files, blogs, or wikis. A particularity of this markup language is its compatibility with various coding software, online websites, and even opening and editing a Markdown file from a mobile phone.

Additionally, Markdown can be used in open-source editors, a feature that is attractive to companies with limited investment capacity for migrating or deploying an authoring environment.

Markdown seems to have gained significant popularity due to its effortless implementation. In a LinkedIn post *Intro to Markdown, Benefits & Drawbacks*, Sully Vickers, a content creator, and developer, describes Markdown as "simple" and "intuitive". This language separates content from formatting through the use of special characters and provides the writer with a readable content.

Sufyan Shaikh, a front-end developer, wrote in his article: *Markdown: What Is Markdown, and How Do You Use It?* "tags [...] make it quite difficult for humans to read HTML and understand how the text will look after the computer renders it". Markdown presents a user-friendly interface while offering comparable formatting capabilities to HTML. With a swift learning curve due to the use of simple formatting, implementing Markdown within a team is fast and efficient.

Developers may be disinterested in mastering traditional authoring tools, leading to less engagement with authors as they are less interested in interacting directly with the documentation. However, developers possess undeniable expertise in coding languages and frequently serve as the primary point of contact for technical writers. Concurrently, Markdown empowers technical writers to seamlessly author content using code editors. With Markdown, developers no longer need to step outside their IDEs - their comfort zones - to consult, interact with, or add to the documentation. This fosters

communication and collaboration between teams. Furthermore, deploying a developer-friendly environment has the added benefit of obtaining support from programming language experts. As one professional in the survey enunciated: “They don't need to invest deeply in hiring data technical expertise they can actually do with their own set of developers who might also write markdown”.

As a result, Markdown appears to be an apt choice for small technical documentation teams. Its tag-based content is also easy to understand and implement in software-oriented environments, where developers can thrive and collaborate. Nonetheless, while Markdown excels at handling simple structures and facilitating the creation of smaller documentation with less reused content, it falls short when it comes to managing complex content.

In an interview with a 20-year-veteran information architect at a multinational pharmaceutical company, another shortcoming of Markdown was brought up: “You cannot have complex tables depicted in a Markdown file where you might need to merge or span across multiple cells to make it a bigger field”. Markdown is a simple formatting language, and it does not cover complex content such as tables.

The most characteristic drawback of Markdown, in comparison to DITA, is its inability to enable the use of attributes and content reuse. Some Markdown advocates might argue that content reuse can be achieved through snippets-small chunks of code stored in a separate file that can be referenced within multiple codes-, but regardless, this markup language does not include a system of single sourcing as wide and complete as phrase reuse, which is a key aspect of single sourcing. The lack of attributes correlated to the single sourcing of each piece of content can impede the ability to maintain a cohesive message throughout the documentation and create translation management challenges.

Besides the need to keep the documentation simple, Markdown is impacted by the necessity to build an environment in which it can unfold. Markdown does not integrate DTDs or version control. Therefore, while Markdown is presented as an affordable solution, the remaining need to incorporate new software programs and products stands for potential additional licensing costs, depending on the tools chosen.

Migrating the documentation from DITA to Markdown can be a difficult process, especially with legacy products. Content reuse, complex tables, ditaval, and DTDs are all unmanaged features that need to be adapted and rebuilt to maintain a coherent structure and documentation. This migration task can be time-consuming, and the authors may struggle to reorganize content despite the simplicity of Markdown.

Conclusion

Our research started with the idea that DITA was a widely used standard and that its influence did not seem to flicker with time. First, the 2015 article *DITA: A Universal Solution?* corroborated our statement, and then the literature

review provided a larger view of the solutions used within the technical writing community, such as Oxygen, Framemaker, or even Word.

However, as our study went on, we were confronted with other options. 56% of the surveyed technical communicators affirmed they are using DITA with a majority of Oxygen users. Still, our survey showed that there is a variety of used authoring tools among the 44% of other respondents, including Confluence, Astoria, Paligo, Zendesk, Antora, MadCap, or Visual Studio Code. These multiple editors have been improved to fulfill the requirements of technical writers and provide competition to DITA editors, as they are easy to understand and use.

Our focus on Markdown shows that it is an ideal tag-based markup language to collaborate with developers. It is also easy to learn, understand and implement for simple documentation. Still, the limitations are numerous when writing complex content since Markdown does not manage complex tables, single sourcing or DTD the way DITA does.

Overall, DITA is a powerful framework for structured content, which enhances single sourcing and publishing across different channels and formats. Despite its gradual learning curve and the considerable initial time investment, the use of DITA ensures consistency, makes localization efficient and reduces translation costs.

To conclude our article, we observed that the technical writing field is constantly evolving with new features, new editors, and an unstoppable debate for the best authoring tool. Our survey showed that many aspects need to be taken into account before initiating a migration or choosing a first authoring tool. We believe that DITA, with its innovative features and strong user network forum, has a promising future and will not disappear from the market in the foreseeable future.

References

- Achachlouei, M. A., Patil, O., Joshi, T., & Nair, V. N. (2021). *Document Automation Architectures and Technologies: A Survey*. Retrieved from arXiv preprint : <https://arxiv.org/abs/2109.11603>
- Ajose-Coker, D. (2022). *Subject Matter Expert Role in CCMS survey*. Retrieved from Componize: <https://componize.com/subject-matter-expert-and-ccms/>
- Andrews, M. (2020). *Why You Should Ditch DITA and Adopt Headless Instead*. Retrieved from Kontent.ai: <https://kontent.ai/blog/ditch-dita-and-adopt-headless/>
- Bailie, R. A. (2021). *Content Operations for TechComm*. Retrieved from ComTech Paris Cité: <https://comtechp7.hypotheses.org/files/2021/10/Interactive-session-Content-Operations-Rahel-Bailie.pdf>
- Bébié-Valérien, G. (2022). *DocOps, in between figure to implement docs as code*. Retrieved from ComTech Paris Cité: <https://comtechp7.hypotheses.org/files/2021/10/docOps-in-between-figure-to-implement-docs-as-code-Gaspard-Bebie.pdf>
- Canart, C., Garnier, M., & Morceau, H. (2019). *Determining whether DITA is good for you*. Retrieved from ComTech Paris Cité: <https://comtechp7.hypotheses.org/731>
- Ciesielski, J. (2020). *How to leverage multilingual terminology in DITA*. Retrieved from C-Jay International: <https://www.cjay-international.com/post/how-to-leverage-multilingual-terminology-in-dita>
- Content Rules. (2020). *2020 DITA Satisfaction Report*. Retrieved from Precision Content: <https://www.precisioncontent.com/wp-content/uploads/2016/10/2020-DITA-satisfaction-report.pdf>
- Johnson, T. (2023). *From DITA to docs-as-code and Docusaurus: Q&A with Mike Howes*. Retrieved from I'd Rather Be Writing: <https://idratherbewriting.com/blog/docusaurus-questions-and-answers-howes>
- Kasim, N. N., & Khalid, F. (2016). *Choosing the Right Learning Management System (LMS) for the Higher Education Institution Context: A Systematic Review*. Retrieved from International Journal of Emerging Technologies in Learning: <http://dx.doi.org/10.3991/ijet.v11i06.5644>
- Marion, L., Catherine, S., Jean-Michel, V., & Laureen, Z. (2015). *DITA: A universal solution?* Retrieved from ComTech Paris Cité: <https://comtechp7.hypotheses.org/27>
- Metapercept Technology Services. (n.d.). *Unlocking the Benefits of DITA-XML vs Docs as Code: What's Right for Your Project?* Retrieved from LinkedIn: <https://www.linkedin.com/pulse/unlocking-benefits-dita-xml-vs-docs-1f/>

- Mondélice, A. (2020). *How To Get the Best Out of Your Localization Processes*. Retrieved from IXIASOFT: <https://www.ixiasoft.com/how-to-get-the-best-out-of-your-localization-processes/>
- Palliyathu, V. G. (2021). *A Publishing Adventure: From Markdown to DITA and back*. Retrieved from CIDM: <https://www.infomanagementcenter.com/from-markdown-to-dita-and-back/>
- Palliyathu, V., Tiwari, D., Kothari, A., Upendra, V., & Gupta, R. (2020). *Dynamic Generation of an Ontology-Based AI Schema for Chatbots*. Retrieved from Technical Disclosure Commons: https://www.tdcommons.org/dpubs_series/3368/
- Pratt, E. (2020). *Future techcomm trends – differing opinions, and why everyone could be right*. Retrieved from ComTech Paris Cité: <https://comtechp7.hypotheses.org/files/2020/11/Ellis-Pratt.pdf>
- Roopnarine, O. (n.d.). *How are new technologies affecting the technical writer/audience relationship?* Retrieved from 2018: https://conservancy.umn.edu/bitstream/handle/11299/196497/Roopnarine_Capstone_Research_Paper_Audience_050918_v012.pdf?sequence=1
- Schengili-Roberts, K. (2016). *The Past, Present, and Future of DITA*. Retrieved from IXIASOFT: <https://www.ixiasoft.com/past-present-and-future-of-dita/>
- Schengili-Roberts, K. (2019). *The Future of DITA*. Retrieved from <https://www.ditawriter.com/wp-content/uploads/2019/12/The-Future-of-DITA-Boston-DITA-Users-Group.pdf>
- Shaikh, S. (2022). *Markdown : What Is Markdown, and How Do You Use It ?* Retrieved from LinkedIn: <https://www.linkedin.com/pulse/markdown-what-how-do-you-use-sufyan-shaikh/?trackingId=WgCqR31NSsyyh2ut0sM7jg%3D%3D>
- Swope, A. (n.d.). *Four Best Practices for Sharing Content Across Departments using DITA*. Retrieved from DITA Strategies: <https://www.ditastrategies.com/media/sharing-content-across-departments-using-dita>
- The Acrolinx Team. (2023). *The Power of Content Reuse, with the Founder of DITA Michael Iantosca*. Retrieved from Acrolinx: <https://www.acrolinx.com/blog/the-power-of-content-reuse-with-the-founder-of-dita-michael-iantosca/>
- Vickers, S. (2023). *Intro to Markdown, Benefits & Drawbacks*. Retrieved from LinkedIn: <https://www.linkedin.com/pulse/intro-markdown-benefits-drawbacks-sully-vickers/?trackingId=nq%2B5%2FeP7S4%2BqqGQUfcxSzw%3D%3D>
- Wang, V. (2005). *An Analysis of SCORM and DITA*. Retrieved from <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9ebf560c65f398d9dd21aace035b6d09226c772e>

- Weimann, S. (2019). *Open DITA: Making DITA Reuse More Accessible*. Retrieved from DitaExchange: <https://ditaexchange.com/open-dita-making-dita-reuse-more-accessible/>
- White, L. (2015). *The Elusive Promise of Content Reuse*. Retrieved from IXIASOFT: <https://www.ixiasoft.com/the-elusive-promise-of-reuse/>