10 Secrets to a Successful DITA Implementation

Short for "Darwin Information Typing Architecture," DITA may be the most compelling innovation in technical publishing since the invention of XML. It promises to help companies deliver technical documentation that significantly improves their ability to meet their customers' needs for information that is accurate, up-to-date, relevant and easy to use, while also reducing the cost of creating and delivering that information.

But like all innovators, early adopters of DITA must learn how to gain all the benefits of this new technology while avoiding pitfalls that could reduce those benefits or, at worst, create more problems than it solves. If you're thinking about implementing DITA, then this white paper offers our top ten best practices for ensuring success.

What is DITA?

DITA is an XML data model that requires you to take a modular (that is, a topic-oriented) approach to creating information. Since modularity is a requirement of DITA, and not an option, it's an important innovation. But DITA goes further; it also defines an architecture for building and maintaining publishing systems. The most significant and innovative capability of this architecture is "specialization," which allows adaptation of DITA to new requirements while maintaining compatibility with existing systems and data. This specialization capability of DITA brings a previously unknown level of flexibility to XML-based publishing.

DITA originated at IBM and arose from the need to simplify development and maintenance of XML-based systems for creating and publishing technical documentation while also making it easier for writers to learn how to author XML-based information. In 2004, IBM contributed the design of DITA to the community under the auspices of OASIS (http://www.oasis-open.org), the Organization for the Advancement of Structured Information Standards. As a result, DITA is now an open standard that is available to everyone.

Properly implemented, a DITA-based publishing system should deliver the following key benefits to technical documentation:

Reuse
To eliminate redundancy, improve accuracy, and reduce the effort to update information, you should be able to reuse and repurpose information, so you can create "a single source of truth."

Sharing
You should be able to construct your information in a way that allows other groups both within and outside your organization to incorporate your information seamlessly into their own processes, adding further value to the information you create.

Relevance
You should be able to create your information in modules, and automatically assemble those modules according to the needs of each individual so that everyone gets everything one needs, nothing more, nothing less.

Automation
To achieve these objectives cost-effectively, automation holds the key; you should be able to automate your publishing processes so that you can produce more information in more variations more quickly than current labor-intensive processes involve.

This white paper discusses our top ten best practices for achieving these benefits with DITA. You may notice that many of these best practices apply whether or not you use DITA, but that's part of the point of this white paper: DITA embodies many best practices, so you don't have to re-invent them, but it's not a shortcut for all aspects of implementing an XML publishing system.

single-sourcing
solutions, inc.
www.single-sourcing.com

Secret #1: Choose DITA, but Make it Your Own

It's easy to become so enthusiastic about DITA that one loses perspective. We're enthusiastic about DITA too, and for most technical publishing systems, we recommend DITA. But we also recommend that you make sure your needs match what DITA can provide, and more important, that you adapt DITA to meet the specific needs and attributes of your organization.

To appreciate the implications, imagine buying an accounting system and installing it without any setup or configuration to meet your own business's requirements. While some benefits may be achieved with the out-of-the box functionality, the true value of such systems is achieved only when you configure the accounting system to match your needs. Similarly, with DITA, it is possible to achieve some success by implementing DITA straight out of the box. As long as you intend to use DITA for technical documentation (and not for some other document types, such as a data sheet or catalog), then DITA already approximates what you need, although you may have to make some compromises. The cost-benefit tradeoff may make it acceptable for a small company to use DITA without modifications, but any medium-sized or large organization should adapt DITA to its own needs.

So, how do you make DITA your own? The key lies in the next secret.

Secret #2: Know Your Destination and Plan Your Route

The happiest and most optimistic point of a project often lies at its beginning – wouldn't it be better if you reached that point at the end of the project instead? You can, but only if you plan properly, to make sure that your project gets off to the right start. There are three areas you should focus on:

• Leadership

• Objectives

• Implementation Plans

The rest of this section looks at each of these in turn. You may find the advice familiar since it applies to any significant project, but it bears repeating here since it's critical to success.

Leadership
We have all heard that executive sponsorship is vital, but why does it matter for XML publishing, which seemingly should affect only a narrow part of the organization? The answer lies in the transformative nature of XML publishing, to achieve its full potential, its impact will be felt far beyond the publishing group.

Because of that impact, you may need to involve departments or even separate divisions that have not worked together closely in the past. You may be surprised to learn that we have often seen that the greatest problems in a project have been political rather than technical. Without having leadership involved, who can resolve political differences quickly, projects can take two or three times longer to complete.

Objectives
While it may be obvious that you have to know your destination before you hit the road, we have seen many instances where the objectives of an XML publishing project changed repeatedly during the course of the project. These changes occurred because, as the project unfolded, further opportunities for improvement became apparent as the team became more familiar with the potential of XML publishing.

Improvements often cost much more to implement on-the-fly than if they had been part of the original plan because of previous decisions and work that were completed without knowledge of the new objectives. While some shifts in objectives during the course of a project are inevitable, we find that, in many instances, the impact could be vastly reduced by involving someone with previous experience in similar projects.

Most people involved in planning a new technical publishing system only know word processing and desktop publishing software, so they usually see the new system as an incremental improvement over the existing system.

But in reality, using XML for publishing can dramatically transform the resulting information while also having a significant impact on current processes. In other words, implementing XML is a true paradigm shift, so unless someone experienced with XML publishing is involved in establishing objectives, it's likely that project objectives will repeatedly shift, deadlines will be missed, costs will escalate, and the resulting system will be ill-prepared for future improvements.

One word of caution: outside experts should not be used simply to define your objectives for you, their role should be to help your internal team discover and prioritize your organization's objectives.

Implementation Plans
Even if you do a great job of setting objectives, further pitfalls await. Everybody knows that you have to plan the work before you work the plan, but we have been surprised many times by projects that have gone wrong because this step was short-changed.

The decision to launch a project always takes too long, thus leaving too little time for implementation, but you must resist the temptation to squeeze down the planning and leap right into the doing.

The greatest impact of XML-based publishing is on internal processes; it is much more than substituting technologies and file formats. That's why many (but not all) of the best practices in this white paper focus on internal process changes and not on technology. So, as you develop your plans, keep many of the following secrets in mind.

Secret #3: You Can't Eat the Elephant in One Bite

Once you're bitten by the XML publishing bug, you will see opportunities for dramatic improvement everywhere in your organization. So much waste! So much redundancy! So much inaccuracy! How could we be so blind?!?

But you have to resist trying to change everything at once. There are too many people, too many processes, and too many document types to tackle everything at once. Instead, you should start with one group, one process, and one set of related document types.

But make sure you take the long view during your initial planning, so that phase VII of your project works well with phase I. You don't want to have to go back and do a lot of rework every time you proceed to a new phase.

Secret #4: Embrace Change

Here's a surefire way to fail: start with the aim of "minimum disruption." Sounds good – won't work.

You expect to leave the same tools and processes in place and get a different result? You don't expect to affect anyone or change anything, but you want to achieve great benefits?

There is no magic "black box" that you can connect to your existing system to achieve your objectives. If you want to achieve dramatic results, expect to make dramatic changes. Since people naturally resist change, however, you'll need to sell them on the organizational and individual benefits of the changes.

Anyone involved in creating or updating your content will be affected, and since their support will determine your ultimate success, you must be willing to invest the time to help them move from resistance to enthusiasm. Our best advice: expose your team to people who have already gone through the changes that you're just starting. You need to involve end-users in the planning stage to ensure they are fully committed to the success of the project.
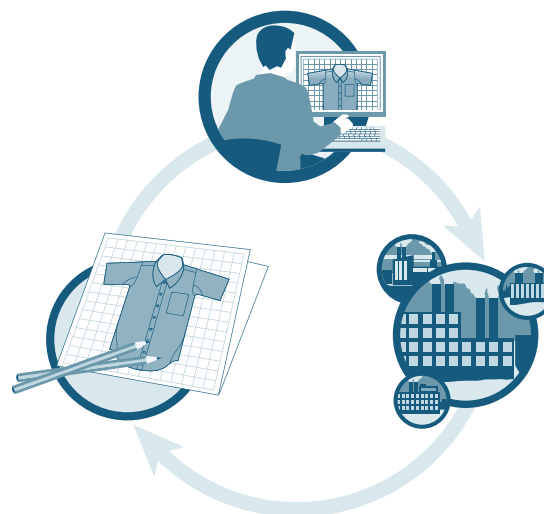
Secret #5: Limit Complexity

If you implement a complex system with software products from many different vendors, you will incur higher costs and risks compared to dealing with one or two vendors, because you have to do a lot more work to choose, integrate, test and keep everything working.

In traditional publishing processes, which involve a lot of manual work, this is usually not a problem. There may be many moving parts, but human intervention integrates them and keeps the whole machine working. For example, contributing authors may use word processors while the technical publications department uses desktop publishing software and manually imports the word processor files as needed.

In an XML publishing system, however, one of the goals is to eliminate human intervention, make everything work together automatically, and have a single source of truth (see Secret #9 for more about single sourcing). Fulfilling this goal requires tight integration among the various software products.

XML publishing systems must also deliver more functionality and productivity than the traditional systems they replace, so one of the key project requirements usually includes the implementation of a content management system as well.

No single vendor offers a complete system that delivers all of the functionality needed in support of every type of content. That leaves customers with the task of selecting vendors for each piece of functionality that's needed.

The answer is to limit the number of vendors involved; choose enough to accomplish your goals (both immediate and future), but no more. You will also benefit from using expert assistance to help you match your current and future needs with products and vendors.

single-sourcing
s o l u t i o n s ,   i n c .
www.single-sourcing.com

Secret #6: Pack Lightly

One of the most prevalent misunderstandings in publishing is that existing processes and tools produce information that is sufficiently consistent to allow automatic conversion to XML. No matter how many times we have encountered that belief – and no matter how insistently it is expressed – it is always wrong.

Word processing and desktop publishing tools survive precisely because of the flexibility and freedom they provide to authors. These product attributes are diametrically opposed to the primary purpose of XML content creation, which is to constrain the author to create content according to a set of rules.

If you are converting printed manuals to DITA, then you will face a serious additional complication because DITA is not only topic-oriented, it also expects each topic to be of a certain type (concept, task or reference) that can stand alone – exactly the opposite of most printed manuals. So, converting existing content may involve not only tagging that content, but also reorganizing and rewriting it.

As a result, our advice is to "pack lightly" on the road to DITA and convert legacy content only when you will either be able to reuse that content several times over, or when you plan to improve that content many times in the coming years.

Secret #7: Rules are Good – Too Many are Bad

XML enforces rules through its use of a data model, which is either a DTD or a Schema. We have seen many implementations that start out with data models that have too many rules and have rules that are too restrictive. In such cases, authors spend too much time learning the rules, as well as struggling to find ways over or around the rules to accomplish their tasks. Eventually, such data models must be relaxed, which has several costs, including changing and testing the data model, retraining authors, and possibly changing existing data.

One of the advantages of using DITA is that the rules are based on lots of experience where others have gone overboard with rules and then pulled back to a sustainable level. The challenge for you is to adapt DITA to your own needs by expanding the data model (in other words, by adding rules) without going too far. Outside expertise can often help you avoid this pitfall.

Secret #8: Authoring Must Change

One of the great strengths of DITA is that it requires a modular approach to building documents where the building blocks are called "topics." Topics are small write-ups that cover a specific subject in a specific way (out of the box, DITA supports three specific ways of covering a subject: concept, task and reference). However – and here's a key difference compared to other approaches to writing documentation – each topic must be able to stand alone because it needs to be usable in many different contexts.
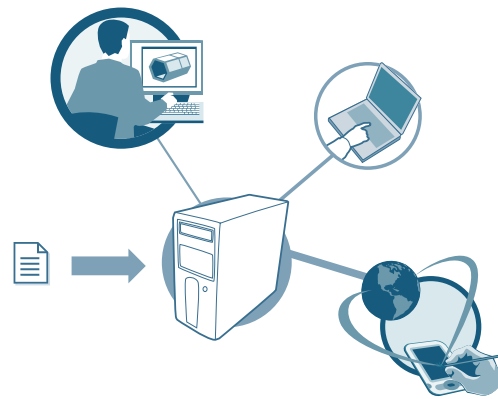
To support a successful implementation of DITA, your authors may have to unlearn years or even decades of incompatible writing habits by learning how to use:

- Topics – writing information in standalone modules is a big change from writing complete manuals, where modularity is often practiced in a haphazard fashion, if at all. Authors must learn to write a topic without knowing which information may precede and follow their topic. Not only can they no longer rely on readers having read a previous section, they can't even rely on the existence of that previous section.

- Information types – DITA specifies three standard information types – concept, task, reference – which requires that authors think more strictly about which information goes where. For example, the description of a procedure, which belongs in the "task" information type, must focus on the "how," and authors must learn to save the "why" of a procedure for a related "concept" information type.

- XML tags – as with any XML-based application, DITA's rules are embodied in a data model that specifies which tags may be used and how they may be used. For authors accustomed to word processing or desktop publishing software, this is a big change. Authors must transition from complete freedom to following rules that the software interactively enforces, and this change involves both learning the rules and learning how to use the software to follow the rules. In addition, authors must abandon responsibility for the formatting of the finished document, and instead rely on downstream automated publishing processes to convert their tagged text into appropriately designed and formatted documents.

If you must translate your documentation into other languages, then you should also consider training your authors to write fewer words – often called "minimalism." Many organizations that are implementing DITA are also training their authors in minimalism because research shows that properly trained authors can reduce their word count by up to 40%, which can have a huge impact on translation costs. In fact, even if you publish in only one language, the advantage of writing fewer words is that your customers have to read fewer words.

Secret #9: One Source of Truth Means One Owner

One of the benefits of an XML publishing system is that it allows you to create a single set of reusable information modules that you can use in a wide variety of document types. Ideally, your information repository would contain no redundant information at all, and thus represent a "single source of truth."

The advantage of this approach is that when you have to update your information, you can make a single change in one module that can be reflected in many documents at once through automatic republishing. This is much more effective and efficient than finding each document that must be updated, finding the specific information that must be changed, making the same change over and over again, and fixing up the formatting of each document if the change affects page breaks.

But the disadvantage of this approach is that you must define the rules for making changes, and that means that you must first declare ownership. This is a politically delicate decision, but it's far better to wrestle with this decision at the beginning of the project than to try to address it on the fly after you have developed and implemented the technology to support your new system.

If you do not determine ownership in advance, then you risk not only wasting a lot of precious time during the start-up of the new system, but also being forced to make changes to your new systems to enforce access privileges in order to protect ownership.

Secret #10: Train, Train, Train

As you saw in #8, implementing a DITA-based publishing system represents a huge change in your processes. To ensure success, you must plan the training as carefully as you plan the rest of the implementation.

There are many points during your planning and implementation where you should train your authors. Early on, you should train your authors about the theories and concepts behind DITA, as well as the business reasons for switching to DITA. You should then plan on a series of training sessions to help the authors understand the concepts more deeply. Finally, authors should learn how to use the new system just before they can start putting that training into practice.

But training shouldn't end when the system is launched. Experts and peers should review authors' initial work through the new system and provide immediate feedback, so that authors can move up the learning curve quickly.

Summary

XML provides an excellent path to automating the publishing process and achieving a leap forward in an organization's ability to produce and maintain volumes of high quality, tailored information at a fraction of the typical costs. Until recently, however, XML came with high start-up costs, which limited adoption. New standards, such as DITA enforce best practices and enable reuse of information and interoperability of publishing systems. Today, many organizations are adopting DITA because they are simply looking for faster implementation; they deploy the DITA open source toolkit with little to no specialization. These organizations are simply looking for an out-of-the-box solution that automates their publishing process with minimal implementation effort. As the technology matures, we will see an increasing number of new solutions, built on DITA, that take this idea one step further and leverage this excellent infrastructure technology to develop production-quality, turn-key applications for various publishing processes.

DITA has the potential to accelerate your adoption of XML-based publishing and to improve the long-term viability of your XML publishing system, but only with proper planning and professional execution. To take advantage of the transformation that DITA can bring to publishing, you must be willing to involve everyone in the organization who will be affected, replace many existing practices and processes, and bring in outside expertise to supplement your own knowledge and skills.

If you execute well, your organization will enjoy the fruits of your work by delivering information to your customers that is more accurate, more current, and more useful, while maintaining or reducing your costs.

single-sourcing
solutions, inc.
www.single-sourcing.com

About PTC

PTC was the first vendor to support DITA at IBM, the first vendor to support DITA authoring, and the only vendor to participate as a founding member of the OASIS DITA Technical Committee. As a result, PTC has led the way for DITA adoption offers the broadest and deepest capability and experience in DITA.

PTC is also the first to deliver a complete out-of-the-box solution for service manuals that's based on DITA. An integrated solution can be critical to realizing the key benefits of DITA, and a pre-built system designed around a specific document type – in this case, service manuals – can give you a low-risk way to jump start your implementation.

Because the PTC Arbortext solution for DITA is fully integrated, all of the components are built to work together out of the box. Everything is tested together, deployed together, and upgraded together. In other words, we've done the work of building the system, so that you don't have to.

PTC and its partners offer software and services that can play a pivotal role in your successful implementation of DITA based publishing system. With live installations of our DITA support, and with more on the way, PTC intends to remain unmatched for its breadth and depth of DITA support in the future.

PTC Global Services

Our Global Services ensures organizations realize the maximum value from their dynamic publishing investment. Deploying the right software is critical to automating the publishing process. But to truly realize the cost savings and time-to-market benefits from automation, customers need to ensure that everyone – from senior executives to end-users – adopts the change that comes with new technology and improved processes. In an environment where people are already accustomed to using desktop publishing tools, adoption can be a challenge.

At PTC, we recognize the importance of user adoption. Our Global Services team offers solutions that help you not only implement a DITA solution, but also drive adoption of the solution within your organization. After years of deploying new processes and technology across thousands of customer sites, our Global Services team is able to anticipate the cultural and geographical adoption challenges you'll face – and help you overcome them.

Each project begins with an examination of your current staffing, processes, and technology to determine the optimal deployment strategy and quickest path to value for your organization. We then guide you through each step of our standard Realized Value Methodology, which includes the steps required to facilitate adoption, including a unique training approach called Precision Learning.

single-sourcing
solutions, inc.
www.single-sourcing.com

To learn more about how Arbortext can help your company create and deliver high-quality product information, please visit our website at:
http://www.single-sourcing.com/

3131-DITA-WP-1107