

Managing implementation of structured authoring

Moving a desktop publishing–based workgroup into structured authoring requires authors to master new concepts, such as hierarchical content organization, information chunking with elements, and metadata labeling with attributes. In addition to these technical challenges, the implementation itself presents significant difficulties. This paper describes Scriptorium Publishing’s methodology for implementing structured authoring environments. This document is intended primarily as a roadmap for our clients, but it could be used as a starting point for any implementation.

This white paper assumes basic familiarity with XML and structured authoring. For more information about these concepts, refer to our *Structured Authoring and XML* white paper.



Scriptorium Publishing Services, Inc.

P.O. Box 12761, Research Triangle Park, NC 27709-2761

919-481-2701 or sales@scriptorium.com

<http://www.scriptorium.com>

Implications of structure

In a structured authoring environment, you improve information management by grouping information into semantic units (elements) with meaningful names and associated metadata. For many authors accustomed to a desktop publishing environment, working in this structured context is difficult. In a paragraph-based template, writers have the option of ignoring or overriding settings; in a structured template, content must conform to the required structure.

It's possible to develop and enforce templates in an unstructured environment. In our experience, writers accustomed to these environments generally have little trouble with structured templates. But for writers who have previously worked in a “template-optional” situation, the sudden implementation—and *enforcement*—of a particular structure can be shocking. This transition is comparable to the difficulties experienced in moving from longhand or typewriter-based work to word processing.

The structure that you develop for your workgroup needs to meet two—often conflicting—requirements. It must be sufficiently flexible to accommodate all of your information requirements but not so wide-open that no organization is imposed on the documents.

Costs

Establishing a structured environment is expensive. The costs break down into two main categories: the change in the mindset of authors and the actual technical implementation effort.

Changing hearts and minds

Since desktop publishing came along in the 1980s, most content creators have focused on page-based layout. Provided that the look and feel of the final printed document was correct, the underlying document organization was irrelevant. In some environments, a heavy focus on templates requires writers to do more—they had to deliver a document that met certain standards of *technical quality*.¹ Technical quality refers to a document's internal setup; for example, does it use a template, are styles applied consistently, and are overrides minimized? In a document with high technical quality, the implicit structure of the document is expressed by the formatting tags. That is, main body paragraphs are tagged with a Body style, headings are tagged with the appropriate heading style, and so on.

1. I believe that Michael Mueller-Hillebrandt of cap-studio.de first coined this term.

In a structured environment, technical quality is enforced automatically because authors lose the ability to “tweak” documents. The inability to control the display of information page by page (or line by line) is frustrating to some writers. A select few will spend a lot of time trying to find a way around this limitation. The cost to the organization is a loss of productivity.

Another issue that affects productivity is metadata. Exact requirements vary, but in most structures, authors must provide at least a few metadata items. In some structures, authors are compelled to provide metadata on every element. Excessive metadata interrupts the writing flow, which can make writers less productive.

Staffing considerations

Any change in workflow causes at least a short-term reduction in productivity as staff adjusts to it. The staffing considerations, however, go beyond process changes. You must take into account the possibility of staffing turnover during a structured implementation. Reactions to structure fall along a continuum from delighted enthusiasm to outright hostility. Most of the writing staff will react neutrally or slightly positively.

Scriptorium Publishing has found that the following factors affect initial response:

- *Communication:* Good communication about the project alleviates fear, uncertainty, and doubt. Lack of communication does the opposite. The staff needs to understand the reasons for moving to structure and the benefits it provides.
- *Training:* Writers need training to understand the new workflow. Without training, they will take longer to learn the new process and may resent the steep learning curve.
- *Quality of implementation:* The new structure should closely match the requirements of the content that’s being developed. Imposing a structure that does not accommodate writers’ legitimate requirements will lead to disgruntled writers.
- *Leadership:* Within the workgroup, the attitude of leaders—whether positive or negative—will influence reactions of the entire staff. Without support from leaders, you will encounter heightened aversion to change and perhaps even outright hostility. Leaders may or may not be managers—look for the employees to whom others go for advice.

Implementation effort

Scriptorium Publishing describes the implementation effort as a twelve-step process. The steps are as follows:

1. Identifying implementation goals and metrics
2. Defining roles and responsibilities
3. Establishing timeline and milestones
4. Performing structure analysis
5. Creating structure definition files and application files
6. Creating a legacy document conversion process
7. Setting up output paths
8. Developing documentation
9. Delivering training
10. Converting legacy documents
11. Creating change management process
12. Providing transition support and validating implementation against success criteria

The sections that follow describe each of these steps.

1. Identifying implementation goals and metrics

The first step in the structure implementation is to identify your success criteria. There are numerous reasons why you might consider structured authoring, such as:

- Increased productivity
- Better information classification and management
- Creating XML to support a content management system
- Reduction in production editing and manual formatting time
- Improved consistency of content
- Enabling content exchange via XML

Each organization will have a different set of goals and expectations. These goals should be discussed, prioritized, and documented before the project begins.

The initial project specification should also include the following:

- A list of required deliverables and deliverable paths (for example, HTML created using XSL transformation on XML files)

- Any tool-specific requirements that affect the elements and attributes to be defined
- A high-level description of the planned workflow

After defining goals, you can develop success criteria. This allows you to evaluate the project's success when it is completed. Your criteria should include specific metrics. These might include items such as the following:

- Number of deliverables per writer before and after implementation
- Percentage of information reuse achieved
- Time required to do “final polish” on deliverables

Once the goals for structured authoring are established, and you have defined measurable success criteria, it's time to consider who will do the actual implementation work.

2. Defining roles and responsibilities

Depending on the scope and complexity of your implementation, you may have several different people involved. We recommend assigning the following roles at a minimum:

- Education
- Development
- Review
- Approval

The education role is responsible for getting buy-in from all affected parties—especially managers who approve the effort and staff who will use the new authoring environment. Depending on the audience, you may use different approaches, such as presentations, informal discussions, and training.

If consultants are involved, they will most likely do the development work and then present it for your review and approval. There may also be an internal review on the consultant's side before you see any deliverables.

In larger projects, there may be a development team. For example, one person might be responsible for establishing taxonomy (element and metadata definitions), another for choosing and installing a content management system, and a third for creating XSL transformation stylesheets.

Scriptorium Publishing uses a collaborative approach. We strive to identify or develop technical expertise on the client side as early as possible so that our clients can provide meaningful reviews and feedback as we build their systems. By combining our clients' business requirements and expertise in

their own subject matter with our consultants' understanding of structured workflows, we can deliver a final product that improves on what either of us could do on our own.

Working on implementation and review teams will require significant time commitments from the participants. Any realistic resource plan must take into account other commitments, deadlines, and deliverables that could conflict with project requirements.

3. Establishing timeline and milestones

After defining the project's goals and resources, you can put together a timeline and milestones. These can be tied to business requirements as needed.

As in any project, establishing a schedule creates accountability. Without a schedule, the project will likely become a low priority and be delayed repeatedly.

If you are working with a consultant, project milestones will likely be linked to incremental payments. Scriptorium Publishing generally does structure implementations on a fixed-price basis. We do significant upfront analysis to ensure that we understand your project requirements before negotiating a contract. Structure implementation is expensive; even a medium-sized implementation can easily reach six figures with custom-developed documentation and hands-on training. Determining project scope before the project begins ensures that there are no unpleasant surprises for our clients later.

The timeline needs to include some slack for delays. The most common reason we encounter for lagging schedules is lack of reviewer availability, which causes delays in client reviews.

4. Performing structure analysis

Structure analysis is a critical portion of the implementation. During analysis, you identify your organization's requirements, develop a taxonomy (classification system) that meets those requirements, and consider where metadata should be allowed or required.

As you define the structure, you must balance precision and simplicity. Defining structure with precision leads to large, complex structure definitions. Keeping structure as simple as possible makes it more usable. Other workflow components, especially content management systems or single-sourcing plans, may put limitations on how you define structure and what metadata you create.

Scriptorium Publishing recommends using existing files to develop an initial structure. You will, however, also need to take into account future requirements. Consider how requirements might change in the future and build your system accordingly.

Do not overlook the importance of embedding metadata into the structure. Metadata is critical to making your content manageable—with or without a content management system.

The final deliverable in the structure analysis phase is a detailed document that outlines the proposed structure. This can be delivered in different ways; for example, as flowcharts or hierarchical tree diagrams (Figure 1). The delivery medium is less important than the ideas conveyed.

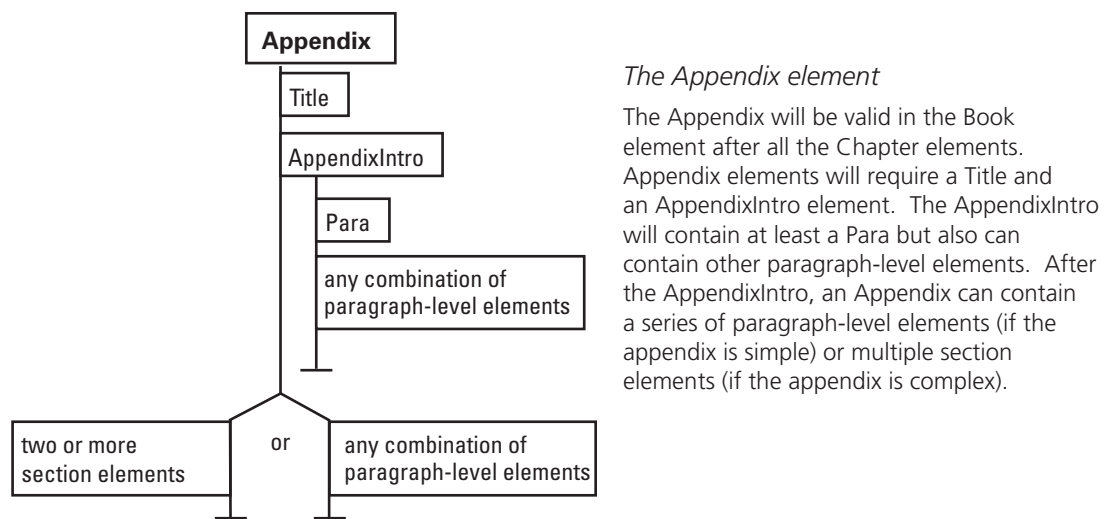


Figure 1: Excerpt of a structure analysis flowchart

After review and approval of the structure analysis, you can develop the structure files. A thorough analysis will allow you to avoid making changes later in the project. The farther downstream you change the structure, the more difficult and expensive it will be.

5. Creating structure definition files and application files

In the structure definition phase, you start with the results of your structure analysis and encode that information into a document type definition (DTD), schema, or other structure definition file. You will probably also need tool-specific configuration files. In a FrameMaker-based implementation, for example, you need an element definition document (EDD). The EDD contains structure definitions, which you can derive

from the DTD, and formatting information, which controls how elements are rendered on the printed page. FrameMaker also requires several configuration files that control how XML is imported and exported.

Creating structure definition files is mostly a matter of expressing in formal syntax the analysis you completed in the previous phase.

6. Creating a legacy document conversion process

After establishing the structure definitions, you need to begin working on the document conversion process. You can usually automate a significant portion of the document conversion process.

Establishing a conversion process early in the project serves two purposes:

- It provides a set of structured documents for testing.
- It may reveal inconsistency in the legacy documents that needs to be corrected or addressed in the structure.

7. Setting up output paths

At the beginning of the project, you listed your output format requirements. After establishing the structure definitions—which enable you to create XML—you can begin to implement output paths. XSL transformation is an obvious choice for the automated production of HTML and other markup languages. Creating printed output will present the most complex challenges.

If your output requirements are varied and complex, you may want to create a diagram that outlines how you get to each item (Figure 2).

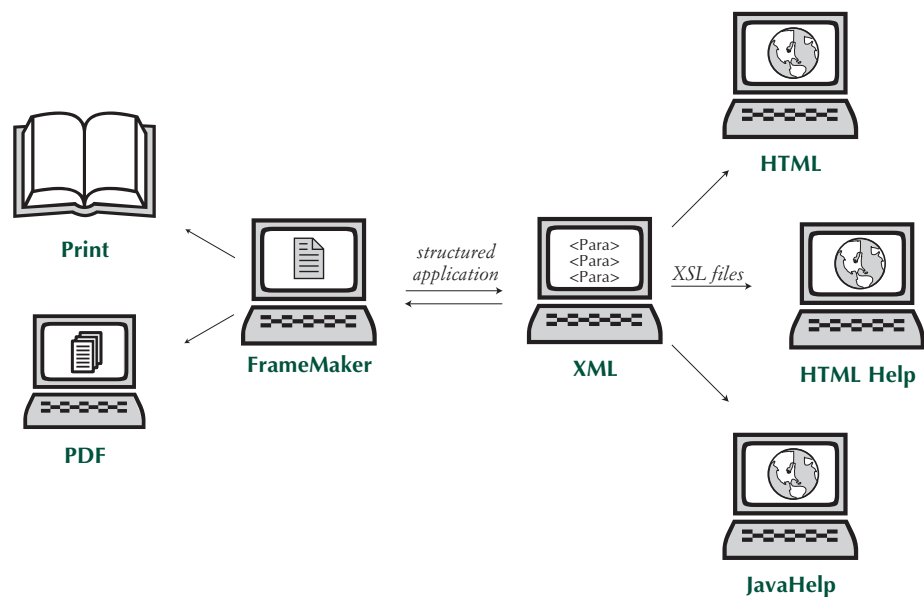


Figure 2: Sample output path diagram

XSL-FO is available for transformation of XML into print/PDF,² but at this time, our assessment is that XSL-FO is not ready for use in a production environment where print/PDF output is critical. Consider a commercial solution, such as ArborText Epic or Adobe FrameMaker, both of which offer powerful XML-to-print rendering engines. If your print requirements are simple, you may be able to use XSL-FO. Expect that the print-rendering effort will be by far the most complex output path you build.

8. Developing documentation

The planning and documentation phases are thankless tasks. If done well, nobody really notices them. But if you skip them or give them less attention than they deserve, you will likely create a structure implementation that is disorganized and impossible to maintain. We recommend that the structure documentation should include at least the following information:

- Structure explanations and recommended best practices for authors
- An explanation of all of the important elements and required components
- In-depth technical documentation for developers, which will be used to maintain the system
- Formatting specifications and algorithms

It would also be helpful to provide details about the project, background information, and the rationale for various design decisions.

9. Delivering training

Knowledge transfer is critical to a smooth implementation. Authors need training on the following topics:

- General structured authoring and XML concepts
- Rationale for implementation
- Structure definitions specific to your organization
- Using the designated authoring tool

We have found that authors accustomed to a template-driven environment where style overrides are discouraged (or prohibited) have an easier time making the transition to structure than less organized groups.

2. From a publishing point of view, print and PDF are effectively identical—both require you to do sophisticated pagination work.

If the staff that built the system will maintain it, little or no developer-level training is necessary. However, if the staff responsible for maintenance is new to the project, extensive training is required. In addition to the information needed by the authors, maintenance staff also needs the following:

- In-depth understanding of the structure
- How to implement and test changes
- Technical training on the various output paths and how to maintain them

10. Converting legacy documents

By this point, you should already have a document conversion process. The quantity of documents that need conversion will determine how much time and effort you put into automating the conversion and addressing issues in the legacy documents. You can expect several challenges:

- Conversion is usually based on formatting that's present in the source documents. Documents whose formatting does not conform to the standard will cause serious problems in conversion. Consistent use of templates in legacy documents will make conversion go faster.
- In some cases, the structure requires information that simply is not present in the source documents. This is common with metadata—the book element requires the author's name, but the author's name is not present anywhere in the source file. Another problem can occur when you implement information typing—the various section types, such as Procedure, Reference, and so on, correspond to a single Heading2 tag in the original source files. Assigning the correct information types requires manual intervention.

The best-case scenario for automated legacy document conversion will probably be approximately 95 percent accuracy. Significant human effort will be required to address the remaining five percent.

To reduce the cost of legacy document conversion, you may want to consider “as-needed” conversion. Instead of moving your entire document library over to structure immediately, convert documents only as required.

Scriptorium Publishing recommends against assigning tedious document conversion to your staff as their introduction to structured authoring. Any interest in structure will likely be eliminated by two months of boring document clean-up.

11. Creating change management process

In the ideal world, you could build a new structured authoring environment, deploy it, and wash your hands of the project. In the real world, however, changes are inevitable. Even in the best-planned, most-organized environment, you will be required to make small changes to the structure, add new output paths, and so on. You must have a plan to manage these changes, as in any software development project. This means developing a change control process and identifying and prioritizing bugs and enhancements.

Your process must address several competing requirements: it must minimize changes, ensure that changes are made in an organized manner, and be flexible enough to ensure that the environment meets the workgroup's evolving requirements.

Some workgroups implement changes on a schedule. For example, for the first two years, the implementation team rolls out new versions quarterly; after that, every six months. You might also schedule change implementation based on priority—changes with higher priorities are done quickly; lower-priority changes are rolled into a scheduled update.

12. Providing transition support and validating implementation against success criteria

After building, testing, and deploying the project, you need to shift resources from development to maintenance. We typically reduce our involvement with a project at this point. Scriptorium Publishing ensures that in-house resources have the knowledge required to maintain and extend the project. We offer follow-on support agreements to help with any new questions that might arise, and then we allow our clients to become self-sufficient in the new environment.

As you move forward, you can evaluate the implementation against the goals you identified at the beginning of the project. Are you seeing increased productivity and reduced production-editing requirements? Is content management improving? If your implementation was successful, you can tick off the items you listed at the beginning of the project (and perhaps a few you didn't anticipate) as accomplishments now.

Summary

A medium-sized structure implementation effort, with new structures, workflow changes, training, and documentation requirements, will take a minimum of six months. The project planning is an absolute requirement. The decisions made in the initial phases, especially in the structure analysis, will determine the success of the project. A close second behind the technical decisions is the education effort. Authors, managers, and others need to understand the rationale for moving to a structured environment. Once the environment is delivered, authors need training on how to use it.