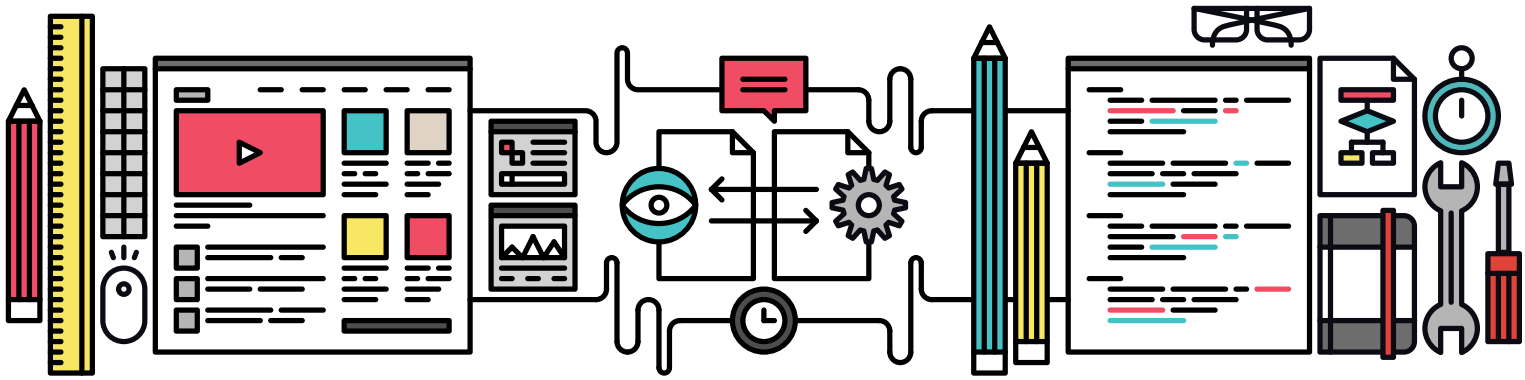
SEPTEMBER 2016

intercom

THE MAGAZINE OF THE SOCIETY FOR TECHNICAL COMMUNICATION



TOOLS OF THE TRADE

WHAT'S THE USE CASE FOR JEKYLL? 6

COLLABORATING AND CONTRIBUTING IN GITHUB 12

GIT HAPPENS 16

GENERATING API DOCS AUTOMATICALLY FROM THE SOURCE CODE 18

CREATING AUTOMATION TOOLS FOR WRITING TEAMS: A PROGRAMMER'S PERSPECTIVE 24

USING DATA-DRIVEN SYNTHESIS TOOLS TO AUTOMATICALLY GENERATE CONTENT 28

AN INTRODUCTION TO SPHINX AND READ THE DOCS FOR TECHNICAL WRITERS 34

TECHNICAL WRITING SUPPORTED BY A PRODUCT LIFECYCLE MANAGEMENT SYSTEM 40

THE BENEFITS OF DITA AUTHORING WITHIN A COMPONENT CONTENT MANAGEMENT SYSTEM 42

shutterstock.com/ K Kreto

Technical Writing Supported by a Product Lifecycle Management System:

A Better Approach for the Creation of Product Documentation

By CHRISTIAN BRAND

IN THIS ARTICLE you will learn the concept of PLM-supported authoring and single-source publishing, as well as the benefits compared to a stand-alone authoring solution or isolated content management systems.

Technical writing and the creation of quality product documentation is a complex field with many pitfalls that have been eliminated with modern tools. Unstructured authoring and mishandling of files on network drives are issues of the past. A modern Component Content Management System (CCMS) offers the functionality and controls necessary to support the author during the editing and publishing process.

You've just implemented a new CCMS…

Today's systems on the market bring these rich features:

- Database-supported content management
- Separation from content and form (XML authoring)
- An information framework which defines structure, content elements, and rules (DITA, S1000D just to name a few)
- Translation control
- Single-source publishing to different layouts and formats

Strengths and weaknesses are often revealed during the implementation process. The CCMS of your choice most likely comes with its own database, a proprietary file server, needs to be integrated with other systems, has its own user management, has its own support for release processes and workflows, etc. In short, you've just created another island-solution within the growing IT landscape of your company.

Getting down into the details

To further streamline and automate the creation of product documentation, data from other systems is needed. Because the CCMS is not connected to any other systems in the company, you need to consider building interfaces to push, pull, or synchronize data. Once this is done, you'll need to constantly monitor these interfaces to ensure everything is functioning smoothly until the next maintenance window.

Because the CCMS also has its own user management facility, you'll be adding another user credential system to the growing list or you'll have to once again enhance the configuration of your single-sign-on environment.

Consider your document approval workflow. The CCMS is not directly connected to your company's Document Management System (DMS). This often means you'll need to duplicate the output and keep copies in both your CMMS and in your DMS.

Let's summarize

The CCMS, without doubt, supports and enhances your authoring process. Moving from (unstructured) writing in a word processor and managing files without version control to structured authoring in XML and controlling files in a modern content management system is a quantum leap in efficiency and control.

However, decision makers should be familiar with the downsides of a stand-alone system, especially when evaluating a new company-wide solution for technical documentation:

- It increases complexity of existing IT landscapes
- It creates additional costs associated with setup and maintenance, including:
  - Additional servers
  - Additional databases
  - Additional storage space
  - Integrations with other systems
- It creates additional running costs associated with additional systems and tools needed
- It incorporates different processes and workflows which often minimize the new time savings
- It (still) creates redundant content

What would a different approach look like?

To create a new product, a company employs designers and engineers who create CAD models, drawings, bills-of-materials (BOMs), specifications, data sheets, manufacturing plans, and much more, to bring a product to life.

To manage all of this relevant product data, Product Lifecycle Management (PLM) systems are widely used in all different industries. Within the PLM system, everything concerning the product, from initial CAD model requirements to the latest specification and defect protocol, is stored under version control and managed by well-defined processes.

Is everything from the product environment really managed in PLM? What about product documentation? Wouldn't it be beneficial to use the existing PLM system for product documentation? What better place could there be to close the gap between product development and product documentation?

Imagine the benefits that could be realized by managing product documentation in your company's PLM system:

- Reduce the Total Cost of Ownership (TCO) by taking advantage of your existing hardware and software infrastructure

- Manage all product documentation assets in the PLM system
- Access existing PLM information such as BOMs and rendered product images and integrate them in your documents
- Integrate PLM metadata such as revision, author, document number, etc. automatically into your documents
- Use existing PLM processes and workflows to review and approve your documents
- Make product documentation assets easily available to marketing or presales departments
- Close the gap between product development and product documentation

PLM-supported authoring tools are not a theoretical approach that might be realized in the future but exist for almost all major/leading PLM vendors (Arbortext for PTC's Windchill PLM system, XPLM Publisher for Oracle Agile PLM, or TecPub Studio for Siemens Teamcenter).

Let's take a closer look at one of these solutions. XPLM Publisher is the authoring and single-source publishing solution available for Oracle Agile PLM. It provides all of the functionality listed above. XPLM Publisher is a DITA CCMS which is tightly integrated in Oracle Agile PLM, combining the best of both worlds—the power and flexibility from content management built into the PLM system and the professional authoring and publishing part of a standalone CCMS.

Conclusion

Product documentation is a critical part of the product record and is best managed in the system that contains the product record. By consolidating the product record into a single system, the high cost of additional IT silos can be avoided. With this approach, users can rely on standardized processes and controls provided by the PLM system to ensure efficient routing, approval, and control of the entire product record. When considering a CCMS, it is beneficial to first check the alternatives to standalone solutions on the market which integrate into your PLM system. You will be better off if the new solution complements your existing PLM system and will not be a new, redundant tool in your company's IT infrastructure.

CHRISTIAN BRAND *works as projecteer in the area of technical documentation, content management, and PLM systems. He's a trained technical writer with a flair for structures and well-phrased content.*