SDL Your Content Their Language

Managing Concurrent Product Releases and User Documentation

February 14, 2011

Today many development organizations work in rapid product release cycles pushing the limits on timelines for delivering customer-facing materials. This increase in frequency adds pressures on teams creating and managing these materials – user documentation, help files, Web sites, datasheets, training materials and so forth. In addition, teams frequently share content where different users may need to make variations on content for a specific deliverable, as for example when managing parallel software trains. How do you best manage the inevitable product variances with rapid development cycles while still maintaining quality and a consistent customer experience?

At SDL, we have seen our clients successfully develop and utilize best practices to manage this process using SDL Trisoft, a Component Content Management (CCM) system developed for handling just this sort of problem. SDL Trisoft provides out-of-the-box support for management and versioning of content using DITA as a content model. SDL Trisoft manages the baseline of the specific versions of topics, maps and images to include for each specific product release. The SDL Trisoft Publication Manager watches over the baseline, applying further context of which conditions, variables and output to apply to that specific release.

This document outlines several best practices we have seen our clients adopt to manage customer deliverables in a rapid development environment with multiple parallel releases. The requirements can vary by industry, types of products as well as workflows.

A key requirement of success is to plan and document your best practices as an organization in how you handle content reuse and how you want to manage variations. For example, a decision to version or branch a topic influences how you package it within a customer-facing deliverable and how to consolidate branch variations in a future release.

Assumptions made for establishing best practice guidelines within an organization:

- Adopt DITA as a content model embracing authoring in topics and assembling into DITA maps. This includes adopting DITA best practices of minimization, clear assignment of topic types, use of relationship tables and so forth.
- Use a CCM such as SDL Trisoft to manage versioning, branching and deliverables such as PDF, HTML, help files, etc. The CCM tracks, as a “baseline,” the various versions and branches of topics, maps and images as they package into a specific product release.
- Manage collaboration and concurrency with colleagues using a state-based workflow process within the CCM; status of topics is always clear to all users, typically in a status such as draft, review, released, in translation, and so forth. Workflow status provides users the ability to identify what content is approved vs. still work-in-progress.

- Adopt a reuse model including both topics (in maps) and components (Content reference). Clearly identify reusable components so either a user can quickly find them by searching or navigating to specific library folders designed to store the reusable objects.
- Use conditions and possibly variables or Keyrefs to manage variations in deliverables – differences in product features, product names, various images, platforms, audience, and so on assist in reducing new branches and versions.
- Determine if a variation of a topic is a real requirement or re-write to cover all uses? Identify what comprises a major change, thus requiring a new version vs. creating a smaller change as a branch.
- Use Metadata to assist in managing the reasons for new versions/branches; Metadata also helps manage the deliverable associated with a product release. Metadata is useful for reporting, statistics and providing a historical reference or audit trail.
- Develop and manage the localization process of deciding when to translate new versions of content. Ensuring that versioning and branching in the source language can be coordinated with the translation process is a critical process that shapes best practice decisions.
- Identify a final product delivery, typically using the DITA Open Toolkit to output PDF files, HTML, help and other deliverables.
- Identify when to consolidate several topic branches, typically with small changes, into a new version to provide clarity and consistency. Often a technical writer must compare the differences consolidating points into a new topic version. A Where Used feature within the CCM identifies the maps referencing the topic; the writer then updates the references (as appropriate) providing a consistent message to your customers across deliverables.

The roles identified below may vary within your organization; however, generally teams should base tasks on an agreed process.

1) **Information Architect / strategies for collaboration**
   - Develop a reuse strategy to handle variations is a collaborative process that requires a great deal of planning in a writing group.
   - Many groups identify “horizontal areas of expertise” that cut across product lines so that content can be written in a universal way to fit more product context and variations.
   - Other groups have cross team meetings to discuss ways in which what SDL calls “universal topics” can be written to be more generic
   - Versions and branching are key capabilities that supplement excellent information architecture and best practices approach to writing.

2) **Lead Technical Writer determines the impact of product or content updates resulting in the requirement to add, change or delete product features within a topic**

   - The Writer initiates a new or changed feature within a deliverable and must determine if he or she should:

    - version or branch a topic;
    - add a new topic;
    - identify impact on structure – potential changes required to a DITA map.
- Establish a best practice of when to version or branch a topic aligning with your development organization. Identify changes using the same tracking number if available – ECO, issue number, support ticket, etc.
- Examine conditions to manage product variances – explore best practices to minimize versioning and branching.
- Metadata identifies the specific version/branch of a topic with the product release intended and is important for identifying this change to a product release.

3) Lead Technical Writer determines if the DITA Map(s) requires versioning or branching

- A new version or branch of a DITA map can be used to address new or outdated topics for the product update. A CCM supports versioning or branching of DITA maps including associated metadata to track specific reasons about the change.
- Workflow steps in the CCM identify approval status of topics and maps – draft, review, approved, and so on. This metadata is viewable to assist other Writers providing feedback on the approval status of a topic or map.
- A CCM provides a Where Used capability to track all reference DITA maps for a topic. Users can quickly determine the impact of a new branch/version and plan for the change.

4) Lead Technical Writer plans what Publications must be updated including output requirements

- The SDL Trisoft Publication baseline provides the context of the collection of specific topic and map versions and branches required for a new product release.
- The Lead Technical Writer decides if the Publication requires a new version or branch to manage the topic and map updates. Various Publications can reference the same or different versions of topics and maps; planning can reduce the number of variations required and improve your customer's experience.
- It may be a requirement for simultaneously editing by users of different branches/versions of the same topic. A CCM supports this workflow and with metadata provides clear labeling to manage the needed variations.

- Writers generate output within the different Publication versions or branches providing a process to manage the variations. The output is stored within each Publication version or branch.
- As the product reaches a ship date, the Publication and baseline should be frozen and released to lock out any further changes for that version or branch.

5) Process for Writers to merge and consolidate branches into a new version

- Develop a process to consider consolidating several topic branches into a new version of that topic. Consolidation typically happens towards the end of a product development cycle or at the beginning of the next cycle.
- A CCM will still maintain all versions and branches providing a rich audit trail; consolidation is useful to reduce the number of variances developed within a product release cycle. However, often those variations must be maintained in parallel and cannot be consolidated.
- Writer compares differences between branches to determine consolidation points for a new version.
- A "Where Used" feature in the CCM identifies what maps are referencing the topic; the writer then updates appropriate references to consolidate variances and provide a consistent message.