Documentation in a Collaborative World: What We've Learned

STC Summit
May 2011
Larry Kunz

Abstract

Two recent trends, commonly known as *agile* and *Web 2.0*, have brought fundamental changes in the way technical communication projects are planned and managed.

By trying new methodologies and by engaging in dialog, members of the technical communication community are discovering best practices that will help us make the most of these trends We're and also learning more about what we still don't know—so that the dialog can continue.

What is agile?

Agile, or "Just in Time" development, began as a software methodology but is likely to gain acceptance in other industries as well. Emphasizing flexibility and responsiveness to customer requirements, agile is characterized by short product cycles, made up of *sprints*, and frequent product updates.

Products and documentation are often shipped before being thoroughly tested and edited, a practice that is based on two assumptions: customers are eager to get the new materials as quickly as possible, and any errors or bugs can be fixed easily in the next update.

What is Web 2.0?

Web 2.0 refers to applications that facilitate collaboration and information sharing, often involving customers along with in-house writers, subject-matter experts (SMEs), marketers, and support personnel. While traditional websites represent a one-way communication channel--the site owner broadcasts content to readers--Web 2.0 establishes a two-way channel in which the "crowd" or "community" contributes content as well.

In Web 2.0, traditional product documentation is augmented by things like blogs, forum posts, and user comments. With the surfeit of new documentation sources comes the need to organize and prioritize the content—a task known as *content curation*. Organizations are beginning to develop *content strategies* to provide an overall framework for the development, presentation, and governance of their customer-facing content.

How are agile and Web 2.0 affecting our world?

As a result of both agile and Web 2.0, technical communicators are experiencing shorter cycles for developing content, and documents are published and republished more frequently. It's a two-edged sword: while we can quickly respond to customer requirements and fix problems in the documentation, we also have to adjust to a "good enough to ship" mindset that says a document can be published before it's been fully edited or even before all technical updates have been included.

A new role for the doc plan

From a project manager's standpoint, the documentation plan (or doc plan)—which traditionally provides a detailed description of deliverables and workflow—can no longer function as the authoritative guide to a project. Instead, the doc plan becomes subordinate to the content strategy[1], which defines overarching principles and provides general guidance but doesn't get down to the level of detail that a traditional doc plan does.

Other challenges

As I demonstrated in my presentation at the 2010 STC Summit, *Managing Documentation Projects in a Collaborative World*[1], agile and Web 2.0 bring several challenges to technical communicators:

- Technical reviews are often ad hoc and very limited in scope.
- Pre-existing or legacy information is easy to overlook, increasing the risk that outdated material is included in the final documentation.
- Editors have less time to edit documents, and they frequently have to do piecemeal edits rather than seeing the documentation as a unified whole.
- Translation is harder to schedule because of shorter development cycles, and translators, like editors, often don't see the documentation as a unified whole.

Over the past year I've monitored, and in many cases initiated, conversations within the technical communication community in hopes answering these questions. Conversations have taken place in several social media including blogs written by others and by me, "water cooler" chats conducted by the STC Technical Editing SIG, and conversations on Twitter. These conversations are helping move the profession toward a set of best practices for dealing with agile and Web 2.0.

Best practices

When considering the daunting prospect of managing projects in the constantly changing world of agile, I'm both amused and reassured by Stephen Gracey's wry observation that it's really no different from managing traditional projects: "Agile and Scrum accept the ultimate truth that no system can be controlled or predicted as long as human beings are involved."[3]

Tools and processes for effective collaboration

In a collaborative authoring environment, it's essential to have tools and workflows that facilitate the process rather than creating barriers to participation. Ideally, the process you set up will make it easy for contributors to provide feedback, communicate with SMEs, and review the documentation.

For collaborative or community-based writing to work, we need tools that make it easy to for SMEs, customers, and other non-professional writers to contribute content while simultaneously giving the writing professionals a way to assimilate the content into their documentation without needing to do a great deal of reformatting. Two kinds of tools are rapidly gaining acceptance:

- Wikis enable groups of contributors to create and update documents using a simple and well-understood set of formatting tags.
- Several text editors now enable contributors to write content in WYSIWYG (what you see is what you get) fashion while storing the content in a structured format like DITA. They can also be integrated with the company style guide to enforce correct spelling, terminology, and mechanics.

Better, more frequent communication with SMEs

A participant in one of the editors' water cooler chats made a simple but profound statement: "Agile is about communication." She went on to say, "I think that's what saves the editing process."

Stuart Bargon of Atlassian, a company that's an innovator in collaborative documentation, shared a few pointers for working effectively on agile teams:[4]

- Using the development team's bug-tracking system to track documentation issues enables more efficient collaboration between writers and SMEs. Atlassian's Sarah Maddox describes an innovative way in which her documentation team uses JIRA.[5]

- "Get rid of the lag": develop and review documentation during the sprint, not in the next sprint. The developers are focused on the topics at hand, which results in better reviews, and the writers don't feel like they're always lagging behind.
- Insist that the developer teams finish all of the user stories in a sprint. Otherwise it's impossible to complete the documentation.

Bargon and his colleagues also discussed the idea of including the documentation in the “definition of done” for each sprint, and they found that there are pros and cons. While it would make the developers more aware of what the writers need to do their jobs, they might resent having additional work to do before the item could be marked completed.

Mary Connor[6] also recommends keeping the documentation synchronized with development activities during each sprint. Software must be coded, tested, and documented before the sprint can be considered done. Connor admits, however, that this practice isn't perfect from the writing team's point of view: the team tends to be very busy at the end of sprints, and less busy while they wait for things to stabilize during the early stages of code development.

Connor also recommends setting up “continuous, automated builds” of the documentation so that developers can review the documentation in a nearly-final format and so that the documentation can be shipped to customers on short notice.

Improving documentation reviews

Here’s some good news: many writers report that reviews go better in agile because subject-matter experts are reviewing documentation that pertains to the exact same code they’re currently working on. Reviews can be quicker and less formal, especially when the writers are full participants in the daily “scrum” meetings. A writer, for example, can talk with an SME and then update the documentation so that the SME can review the changes that same day.

The trick is managing these reviews, informal as they are, to make sure that all relevant material is reviewed with nothing being overlooked. Another difficulty arises in arranging for comprehensive reviews, which are needed to ensure that nothing is missing and that material carried over from the previous release is still accurate and appropriate. Connor recommends doing these comprehensive reviews during the final sprints, which are usually devoted to activities like regression testing and final bug fixes.

Book sprints

For the ultimate in both short development cycles and writer-SME collaboration, a few organizations have tried *book sprints*. These are carefully planned events in which a group of invitees, writers and SMEs, meet together and collaborate to write a technical book in a short period of time -- typically a few days to a week.

Anne Gentle and Sarah Maddox both have participated in and written about book sprints, and they describe their experiences in an interview with Michele Marques.[7]

No one has yet tried book sprints for conventional documentation projects. Still, Gentle's and Maddox's experiences teach us a few principles for effective writer-SME collaboration when short development cycles are needed:

- Detailed planning and a clear mission statement are essential.
- Getting the right mix of people is important.
- Remember the human aspects: the group works more effectively when a few amusements and diversions are built in.

Updating the localization process

In November 2010 I took part in a virtual conference at proz.com titled "Agile and Localization." Participants proffered a number of recommendations for translating content on agile projects. I believe that many of these recommendations apply equally well to user-generated content in a Web 2.0 setting.

Break up the translation into multiple deliveries, rather than submitting the whole job at once. The schedule for the deliveries, and the contents of each delivery, can coincide with agile sprints–except that you should hold off on delivering content that's likely to change in subsequent sprints. Although this approach—breaking the translation process into chunks—is fairly novel, Richard Hamilton argues convincingly that it's the right approach for any project—agile or not.[8]

Train the writers in the principles of reuse and single-sourcing, so that the overall volume of translated content is reduced. You'll also need to train the writers to resist the temptation to change content that's already been sent to translation. The team has to embrace the idea that content, once sent to translation, is final. (Reassure writers by reminding them that publishing cycles are shorter, so there's never a long wait to get changes into the published documentation.)

Sim-ship (delivering both English content and translated content to customers at the same time) is possible if the development team:

- Devotes the last few sprints to bug-fixing rather than developing new features, thus allowing time for translation to occur after the documentation is stable

- Selects a subset of languages, and a subset of document deliverables, for sim-ship—with the expectation that less critical languages and deliverables can wait until later

Managing and curating content

User-generated content is raw material in need of refinement. Someone needs to organize it, integrate it with other content, and decide what to emphasize and deemphasize. This process is often referred to as content curation, using the metaphor of a curator in a museum. Ideally, there's a clear definition of who performs these activities, and they're performed under the rubric of the organization's overarching content strategy.

Ahava Leibtag[9] outlines the kinds of guidance that a content strategy must provide for user-generated content:

- Encouraging people to write content: Should the site require registration and login? (That requires a little more work administratively, but it also facilitates interaction as well as ranking of users according to how popular their contributions are.) Should the site support multiple languages?
- Roles and workflow for handling content as it's submitted: What gatekeeping, if any, is applied to submissions? Do you vet all submissions, or do you publish them immediately–perhaps in an informal medium like a support bulletin—and then move them into the "official" documentation later, after you have a chance to fix problems? When questions of appropriateness arise, who has the authority to answer them?
- Managing the content after it's published: How do you handle content that undermines your company's brand image or marketing messages? What kind of legal problems might arise, and who will police the site for those problems? When is it time to archive or delete the content?

In any event, as Kristina Halvorson points out, the content strategy will prove inadequate if it doesn't address this fundamental question: what's our objective for publishing this content in the first place?[10]

Changes to the editing process

Editors are adapting. During a water cooler chat, one editor commented on the need to edit content that's "somewhat fictional" during the early stages of a sprint. The content will need to be edited again later, but the early edits still play a vital role by ensuring that organization, terminology, and other issues are addressed before the content has a chance to evolve.

Editors handle the issue of fragmentation—of having to edit topics rather than whole documents—by scheduling early editorial reviews of outlines.

Finally, as is the case with translation, the writing teams try to "lock down" the content as much as possible.

The content management system

A discussion among members of the Content Strategy group (contentstrategy@googlegroups.com) in February 2011 revealed some key things to look for in a content management system. Systems are available that can:

- Accommodate frequent editing and rework and are flexible enough to publish the documentation package for internal consumption even when incomplete—in other words, when links are broken or files are missing. This is essential for agile projects.
- Support simultaneous updates on the same file, so that two people—a writer and SME, or a writer and editor—can collaborate in real time. When one person makes updates in the file, the other person can see the updates and modify them.
- Synchronize with text editors that separate content from form—for example, accepting input in a WYSIWYG format but storing the content in XML or DITA.

Web analytics

Web analytics can help you answer questions about your documentation website:

- What are my visitors most interested in?
- Which portions of the documentation succeed? Which do not?
- Which contributors of user-generated content have the most influence?

Rachel Potts[11] defines web analytics as “the use of data such as the number of people viewing pages on a website, how they get to those pages and what they do next, with a view to improving the website in some way.” Because web analytics grew up primarily around marketing websites, Potts notes, it can be hard to apply the measurements in a technical communication context.

For example, does a high bounce rate (percentage of visitors who enter the site and then leave) mean that people are leaving your site in frustration or that they're leaving because they quickly found the information they were seeking?

Some measurements, however, aren't hard to interpret. If you have pages that nobody visits, for example, the pages either aren't useful or are hard to find. Either way, you have a problem that needs to be fixed. Similarly, analyze the search terms that visitors are using most often and see whether the search results are guiding them to content that's relevant and informative.

Both Potts and Anne Gentle[12] emphasize that the key to fruitful web analysis is knowing how every page on your site is supposed to be used. You can compare the intended usage with actual usage, identify the disparities, make adjustments, and then track trends to see whether more adjustments are needed.

All of this takes time. Don't expect to start using web analytics and get instant answers to all of your questions.

Issues and controversies

The discipline of content curation is in its infancy. There are debates over what to call it, over how best to do it, over how much of it (if any) can be done by software rather than by human curators, and even over whether it's truly a new discipline or simply "what we've been doing all along." As the amount of content continues to increase, I expect that content curation will be recognized as a legitimate discipline and that standards and best practices will evolve.

Agile's "good enough to ship" mindset is a stumbling block to writers and editors who have always insisted on high standards of editorial quality. While it can be hard to reconcile these two viewpoints, evidence suggests that many customers prefer getting their content fast to having it be perfect.

Where we go from here

By staying abreast of emerging best practices and staying knowledgeable about the tools available to support them, we can take advantage of agile and Web 2.0 to move closer to our goal of providing information that's accurate, timely, and relevant.

I look forward to another year of conversation within the profession as we continue to address the challenges of providing documentation in a collaborative world.

References

1. Kunz, Larry: *Content strategy for technical communicators: what happens to my doc plan?* Blog article, 25 October 2010. http://www.dmncommunications.com/weblog/?p=2215

2. Kunz, Larry: *Managing Documentation Projects in a Collaborative World*. STC Summit Proceedings, May 2010

3. Gracey, R. Stephen: *Scrummy Content in an Agile World*. Blog article, 17 Feb 2011. http://contentstrategy.rsgracey.com/

4. Bargon, Stuart: *Introduction to agile methodologies* (presentation for Atlassian). Summarized by Sarah Maddox, 26 February 2011. http://ffeathers.wordpress.com/2011/02/26/introduction-to-agile-methodologies/

5. Maddox, Sarah: *JIRA cloning to create a template for repeated documentation tasks*. Blog article, 12 April 2011. http://ffeathers.wordpress.com/2011/04/12/jira-cloning-to-create-a-template-for-repeated-documentation-tasks/

6. Connor, Mary: *How does Agile affect documentation*? Blog article, 25 February 2011. http://www.cleverhamster.com/clever_hamster/2011/02/how-does-agile-affect-documentation.html

7. Marques, Michele with Anne Gentle and Sarah Maddox: *What are book sprints really like?* Interview transcript, 3 December 2010. http://communities.bmc.com/communities/blogs/writeon/2010/12/03/what-are-book-sprints-really-like

8. Hamilton, Richard: *Managing Writers: A Real World Guide to Managing Technical Documentation*. Fort Collins, CO, XML Press, 2009. Page 98

9. Leibtag, Ahava: *Content Strategy and the User-Generated Content Conundrum*. Blog article, 28 Feb 2011. http://www.cmswire.com/cms/web-engagement/content-strategy-and-the-usergenerated-content-conundrum-010344.php

10. Halvorson, Kristina. *The Discipline of Content Strategy*. Blog article, 26 May 2010. http://ht.ly/1TDlN

11. Potts, Rachel: *What can web analytics do for technical communications?* Blog article, 19 November 2009. http://communicationcloud.wordpress.com/2009/11/19/what-can-web-analytics-do-for-technical-communications/

12. Gentle, Anne: *Web analytics for Tech Doc sites*. Blog article, 1 September 2010. http://justwriteclick.com/2010/09/01/web-analytics-for-technical-documentation-sites/

In addition, the following sources also contributed valuable background material:

Bailie, Rahel: Intentional Design blog. http://intentionaldesign.ca/

Halvorson, Kristina: Content Strategy for the Web. Berkeley, CA, New Riders, 2010.

Pulizzi, Joe: Content Marketing Institute blog. http://www.contentmarketinginstitute.com

---

Larry Kunz
Project Manager
Systems Documentation, Inc.
1005 Slater Road, Suite 220
Durham, NC 27703
lkunz@sdicorp.com
+1.919.354.1104

Larry Kunz is a project manager and information architect with Systems Documentation, Inc. (SDI) Global Solutions in Durham, NC, USA. In more than 30 years as a writer, manager, and planner he has experienced the transition from book-based documentation to today's integrated delivery of information from a wide range of sources using different formats and media.

Larry has managed and provided content for both technical writing projects and marketing projects. He holds a Masters Certificate in Project Management from the George Washington University and teaches a Managing the Information Development Process course in the Technical Communication certification program at Duke University. He is a Fellow in the Society for Technical Communication (STC) and in 2010 received the STC President's Award for leading the Society's strategic planning effort.