

Society for Technical Communication

www.stc.org



PROCEEDINGS

5-8 May 2013

Atlanta, GA

Notice

The papers published in these Proceedings were reproduced from originals furnished by the authors. The opinions and security of the information are the responsibility of the authors and not the Society for Technical Communication.

STC grants permission to educators and academic libraries to photocopy articles from these Proceedings for classroom purposes. There is no charge to these institutions provided they give credit to the author, the Proceedings, and STC. All others must request permission.

Society for Technical Communication
9401 Lee Highway
Suite 300
Fairfax, VA 22031
(703) 522-4114
(703) 522-2075 (fax)
www.stc.org

The STC 2013 Technical Communication Summit Proceedings were created with Doc-To-Help.



© 2013 Society for Technical Communication



<http://summit.stc.org/proceedings/>



The Society for Technical Communication's 60th Annual Conference focuses on important trends in our profession. This publication contains papers submitted in support of the 2013 Summit conference sessions. This year's conference is the result of the efforts of many individuals, including the Conference Manager, Program Advisory Committee, and staff of STC.

Conference Chair

Paul Mueller

2013 Program Advisory Committee

Chris Hester, Program Chair

Education and Training/Academic and Research Topics

Dr. Pam Brewer, Track Manager

Content Strategy and Design

Mark Lewis, Track Manager

Content Development and Delivery

Bill Swallow, Track Manager

People, Project, and Business Management

Richard Hamilton, Track Manager

People, Project, and Business Management

Richard Hamilton, Track Manager

Professional Development

Louellen Coker, Track Manager

Web Design and Development

Michael Opsteegh, Track Manager

STC Summit on Lanyrd (for program, session and conference information): <http://lanyrd.com/2013/society-for-technical-communication-summit>

STC Summit on SlideShare: <http://slideshare.net/tag/stc13>

Save the Date!

STC 2014 Summit

18-21 May

Phoenix, Arizona

Hyatt Regency Phoenix and Phoenix Convention Center

Contents

Content Development and Delivery	1
Baby Steps: Automated Publishing Using Arbortext and SharePoint	3
Collaborating in DITA: Lessons Learned from the 2012 State of Collaboration Survey	9
Collaborative Single-Source Development of Documentation and Training	27
DITA Proof-of-Concept Publishing System	37
Doc Sprints: The Ultimate in Collaborative Document Development.....	43
Embedding User Assistance with DITA.....	49
EPUB and Techcomm — Are We Ready?	61
Managing and Optimizing Unstructured-to-Structured Conversions	67
Planning and Creating Engaging Infographics	79
Using Semantic Technology to Create Process-Driven Documentation.....	87
Content Strategy and Design	91
Plain Talk: Get the Marbles Out of Your Message	93
Education and Training	95
Creating Learning That Lasts with Interactive Documents	97
The Traditional Executive Summary Versus the Video Game Proposal One-Sheet	101
User Assistance, Technical Communication, and Learning (Instructional Design)	105
People, Projects, and Business Management	119
A Manager’s Guide to Acquiring Technology	121
Bending Without Breaking: Info Dev Flexibility in Agile	123
Our Processes Are a Mess!	131
Supporting Customers: Onsite and Online.....	135
Professional Development	139
Building and Maintaining Your Personal Brand.....	141
Interviewing SMEs: Covering the Bases and Hitting Homers.....	145
Shockproofing Your Use of Social Media	155
Staking Your Claim on the Social Media Frontier.....	159
User Experience and Usability	165
Addicted to Meaning: Mental Models for Technical Communicators.....	167
Formalizing the Technical Communication and User Experience Relationship.....	173
Gamification is Here: Build a Winning Plan!.....	179
Useful, Saleable, and Buildable: The Role of UX	189
Web Design and Development	195
Every Page is Page One	197

Progressions	205
Cultivating In-House Sources of Documentation	207
Is Your Content Ready for Localization?	211
Managing Technical Knowledge: Contributing Factors to Knowledge Loss	215
SME — ME — E Screencast creation and collaboration	221
Teaching a College-Level Editing Class.....	229
The Retirement of Technical Communicators — Making Retirement an Adventure.....	235
There’s Nothing Simple About Simplified Technical English	239
Using Mentoring Programs to Collaborate With Industry	241
Why Technical Writing Is Always About Training	245
Index	249

CONTENT DEVELOPMENT AND DELIVERY

[Baby Steps: Automated Publishing Using Arbortext and SharePoint](#) page 3

[Collaborating in DITA: Lessons Learned from the 2012 State of Collaboration Survey](#) page 9

[Collaborative Single-Source Development of Documentation and Training](#) page 27

[DITA Proof-of-Concept Publishing System](#) page 37

[Doc Sprints: The Ultimate in Collaborative Document Development](#) page 43

[Embedding User Assistance with DITA](#) page 49

[EPUB and Techcomm — Are We Ready?](#) page 61

[Managing and Optimizing Unstructured-to-Structured Conversions](#) page 67

[Planning and Creating Engaging Infographics](#) page 79

[Using Semantic Technology to Create Process-driven documentation](#) page 87

BABY STEPS: AUTOMATED PUBLISHING USING ARBORTEXT AND SHAREPOINT

Brett McCorkle and Rita Briody

The introduction of DITA and XML-based documentation has changed the way we can deliver content to end-users. XML makes possible the integration of a multitude of applications to improve the delivery of content to users.

Erie Insurance's goal was to provide users with full-text search of multiple document types, presented as web search results. An automated publishing solution that included our authoring tool, content management system, and SharePoint 2010 would act as the infrastructure to meet this goal. This would allow the company to move away from document-centric publishing to deliver content in discrete topics that provide targeted results for users, thereby saving them time and improving overall efficiency.

Our objectives are to explain:

- Where we started (mainframe BookMaster tool with limited publishing capabilities).
- The changes we implemented slowly over the course of several years to set the stage for improving our publishing process.
- The alternatives we evaluated.
- The approach we took to automate publishing our Arbortext content to SharePoint.
- The struggles, mishaps, mistakes, and challenges that occurred along the way.

THE PROBLEM: 90S (AND 80S) TECHNOLOGY SERVING CURRENT NEEDS

The Technical Communications Services Section at Erie Insurance has used ERIEbooks® to deliver content to the company for more than 15 years. ERIEbooks was a “temporary solution” at a time when technical writers created documents on the mainframe using IBM BookMaster, an authoring tool developed in the 80s.

ERIEbooks has several major limitations. It:

- is a pointer system where we store references to the documents we publish; it does not store the actual content.
- can only reference BookMaster documents (viewable as HTML) and PDF files published using a different tool.
- does not provide full-text search. ERIEbooks searches the document number, title, and five keyword fields (20 characters each) filled in by the author.
- does not have feedback mechanisms for users, making it difficult to identify issues with document content and system behavior.
- does not track system or document usage, so we must contact server administrators to obtain very limited usage statistics.

ERIEbooks' poor search capability causes users to waste time finding information they need to do their job or completely prevents them from finding it. Many users lose confidence with the system and avoid using it after a few tries. Not only is this frustrating for users, but it can also have legal consequences for the company.

ALTERNATIVES CONSIDERED

We review several purchased tools and in-house options as potential solutions. Here is a summary of the options we considered:

- Do nothing and continue using ERIEbooks in its current state.
- Purchase and implement an off-the-shelf solution.
- Build an adapter to connect SharePoint to our source files in IBM Content Manager.
- Develop a solution to publish our content to SharePoint 2010 using Arbortext.

ALTERNATIVE CHOSEN

We developed a proof of concept to publish our content to SharePoint 2010 using Arbortext. This solution:

- Eliminates many of the current system's problems, e.g., full text search, usage tracking.
- Uses tools Erie Insurance already owns.
- Integrates our content management system and publishing engine.
- Does not require outside consulting services.
- Provides Agents and Employees with improved search functionality.
- Provides a foundation for additional features to improve the user experience, e.g., user feedback, personalized pages.

APPROACH AND BACKGROUND

Baby steps! Faced with budget constraints and being trumped by high priority projects, we often had to be creative in finding simple solutions to our complex problems.

Over the past eight years, our section made significant progress in our approach to documentation. We evolved from creating and maintaining monolithic books in a mainframe environment, to applying minimalism and structured authoring approaches to documentation written using DITA. Unfortunately, that's a tale for another day.

To put it briefly, we implemented small improvements over the course of several years to get to where we are today. We started with research and training on new methods of writing, progressing to a new publishing tool and taking advantage of a content management system we already had in-house. After this, we spent several years manually converting and rewriting our legacy content from the mainframe.

With all of these improvements to our processes and content, our biggest problem was our ability to deliver the content to users effectively. ERIEbooks' limitations made it clear that we needed to replace our last legacy system.

Since our documentation is now XML-based, we can deliver content to users in a number of different formats, including HTML, PDF, online help applications, and EPUB, if we had the proper delivery mechanism. Our goal was to provide users with full-text search of our content published as HTML and PDF. Because HTML documents involve a large number of files, we also wanted to automate the publishing process. Before we could gain approval for a project to do this, we needed to prove this was possible.

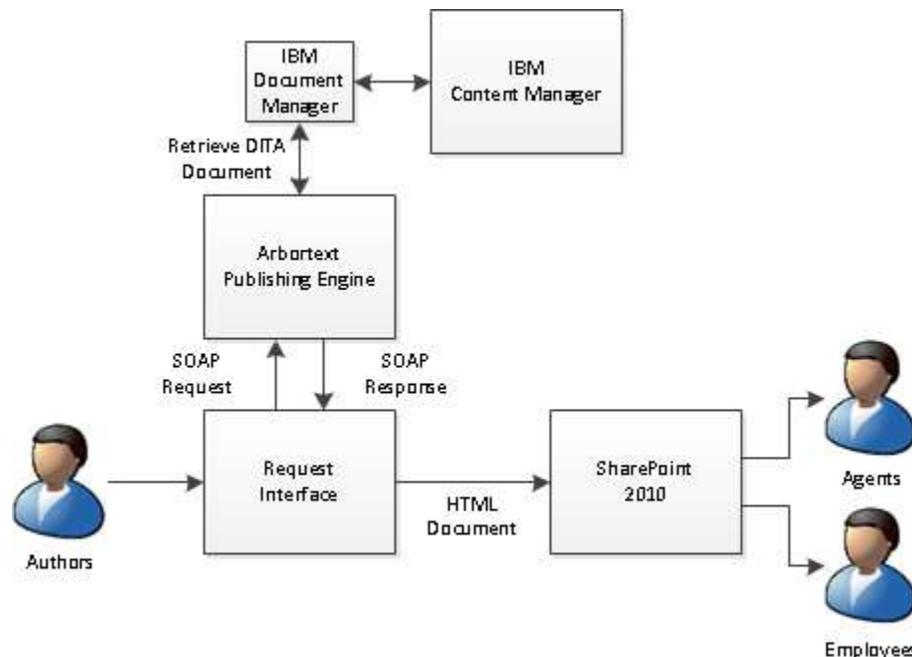
PROOF OF CONCEPT

To prove we could automatically publish HTML content to a repository that users could search, we evaluated several solutions, ultimately creating our own design using in-house systems. Do not let the brevity of this statement fool you. This took over two years and involved several “ah-ha!” moments sprinkled between far more “oh-no” moments.

At a high level, our design had authors submit requests to publish a document and the requested document was published as HTML to SharePoint.

The details of our design include the following steps:

1. Authors send simple object access protocol (SOAP) requests to the Arbortext Publishing Engine containing the document ID and any required parameters.
2. Arbortext connects to our source repository, IBM Content Manager.
3. Arbortext creates an HTML version of the requested DITA document.
4. Arbortext sends a SOAP response with the HTML document attached.
5. The document from the SOAP response is loaded into SharePoint, replacing the previous version.



For the proof of concept, we needed to prove we could:

- Connect the Arbortext Publishing Engine to our source repository, IBM Content Manager.
- Create an interface for authors to send SOAP requests to the Publishing Engine.
- Send the document attached to the SOAP response to SharePoint, replacing the previous version.

- Eliminate the use of out-of-support software applications, i.e., BookMaster.

To connect the Arbortext Publishing Engine to our source content, we implemented the Arbortext adapter to IBM Document Manager on the Publishing Engine. This adapter comes bundled with Arbortext and allows Arbortext to access files stored in IBM Content Manager, through IBM Document Manager.

We previously implemented this adapter for Arbortext Editor, allowing authors to store and manage their source files in Content Manager. Arbortext's API allowed us to create JavaScript to remotely log the Publishing Engine in and out of Content Manager.

Once we established a connection to Content Manager, we were able to use the web service (WSDL) hosted by the Publishing Engine to send remote document composition requests using SOAP. This WSDL allowed us to request a specific source document, output format, and other processing parameters, and in response return a composed document. Next, we needed an interface to submit these requests and send the output to SharePoint.

We worked with Erie Insurance's IT Lab to create a small application to provide authors with a user interface to submit these requests. The IT Lab had an established process and available resources to assist us in building and proving this interface. Without their help, it would have been extremely difficult to build this interface ourselves. After all, we are "just" technical writers.

After several months of gaining approvals, working through roadblocks, and overcoming issues that nearly cancelled the proof of concept, we had an application that sent SOAP requests, processed the response, and sent the attached HTML document to a SharePoint site. We completed testing and validated SharePoint was providing full-text search of our HTML content and providing discrete topics to users in the search results. This provided users with far more accurate and useful search results than our current system.

The proof of concept concluded with minor SharePoint customizations including:

- Correcting a character encoding glyph displaying in the HTML files.
- Removing unwanted files from search results.
- Ensuring search result links open in a new window.
- Proving we could replace an existing document with a new request.

Our experience with this customization showed us the platform is extremely flexible and we can customize it to meet many different needs. The largest hurdle was getting our content into the system in an efficient manner.

This solution would allow us to sunset our use of BookMaster, the original ERIEbooks application, and other version control systems we had in place.

CHALLENGES FACED

Throughout this entire process, management repeatedly asked us to provide metrics on the current system. We attempted to obtain metrics to support our efforts but this proved to be very challenging as no applications monitored use. The only supporting information we had was from the Usability Team that performed studies on Agents and found that they accessed ERIEbooks often.

This was our second approach to replacing ERIEbooks in two years. Originally, we created a "holy-grail" list of requirements and tried finding a solution to meet them all. This was met with much laughter and no success.

We lowered our expectations. We were just looking for a date to the prom; she didn't have to be pretty, just as long as she could dance.

After we learned to manage our expectations, we were able to determine our true goal: helping users find our content. Our next challenge was finding someone to help us meet our goal.

We needed to generate interest and found allies in the Auditing and Compliance groups in our Law Division. These groups rely on finding corporate policies from a specific date when legal action is brought against the company. Failing to do so can have serious legal consequences. These groups helped support our justification for a new system.

A limited budget prevented us from purchasing a solution, and we did not have the skills within our group to develop an application from the ground up. We had to find another group within the company that could provide the skills we lacked. Fortunately, IT had created a group to foster innovation and prove new technologies, the IT Lab.

During the development process, we encountered performance issues that caused the request application to time out. The Publishing Engine took too long to compose our large documents. There were no performance benchmarks for the Publishing Engine, so we had to work with our server administrators to analyze the problem. Consequently, increasing the server hardware improved performance to an acceptable level.

Even with help from other groups, we faced day-to-day challenges we had to address to keep things moving forward. This often took time away from our main assignments and caused us to work long hours to get the job done. In the end, we were rewarded with a successful proof of concept.

RESULTS

Management overwhelmingly approved a project to implement our design. They saw great value in what we had produced using existing tools and limited resources. We started the project and completed an estimate. Unfortunately, the project's need for SharePoint development resources caused it to lose priority to higher ranked initiatives. At the end of 2012, management placed it on hold indefinitely.

Our effort to replace ERIEbooks is still a work in progress. While Erie Insurance values the innovation and creativity we demonstrated with this solution, the company's focus is currently on higher priority initiatives. After all, we are an insurance company; the documentation we provide is an internal service.

In keeping with our Baby Steps approach, we have been able to continue moving forward. We returned to the drawing board and created a simpler SharePoint design. Much like the current system, this design uses metadata and document links, but removes many of the current limitations we experience. This design will take advantage of SharePoint's flexibility and allow us to do most of the work ourselves.

ADDITIONAL RESOURCES

Hackos, JoAnn. Minimalism: Creating Manuals People Can Use. Comtech Services.

Hackos, JoAnn. Structured Writing for DITA. Comtech Services.

REFERENCES

PTC. Configuring Arbortext® Publishing Engine™ (Needham, MA: Parametric Technology Corporation), 2009

PTC. Arbortext® Publishing Engine™ Programmer's Guide (Needham, MA: Parametric Technology Corporation), 2009

PTC. Arbortext® Content Management Guide (Needham, MA: Parametric Technology Corporation), 2009

Hackos, JoAnn. Introduction to DITA: A User Guide to the Darwin Information Typing Architecture - Arbortext Edition (Denver, CO: Comtech Services), 2007.

Hargis, Gretchen. Carey, Michelle. Hernandez, Ann Kilty. Hughes, Polly. Longo, Deirdre. Rouiller, Shannon. Wilde, Elizabeth. Developing Quality Technical Information: A Handbook for Writers and Editors, 2nd Edition (Upper Saddle River, NJ: IBM Press), 2004.

Self, Tony. The DITA Style Guide (Research Triangle Park, NC: Scriptorium Publishing Services, Inc.), 2011.

AUTHOR CONTACT INFORMATION

Brett McCorkle
Senior Technical Writer
Erie Insurance Group
100 Erie Insurance Place
Erie, PA 16530-1104
814.870.3075

Rita L. Briody
Technical Writer II
Erie Insurance Group
100 Erie Insurance Place
Erie, PA 16530-1104
814.870.7082

AUTHOR BIOGRAPHIES

As a senior writer and the system administrator for Erie Insurance's authoring tool, Arbortext, Brett has played a major role in implementing the tool, establishing our use of a content management system, and guiding the design and implementation of our SharePoint publishing solution. With eight years of technical writing experience, Brett has worked on a variety of complex projects providing development and support documentation for developers and other IT personnel. Brett holds a BS in Management Information Systems from Gannon University.

Rita Briody has been a Technical Writer at Erie Insurance Group for over four and a half years. She has focused on projects creating documentation for new software and technology as it relates to initiatives in ERIE's IT Lab - an environment for testing new technology (hardware and software) or new applications of existing technology. This position enabled her to see an opportunity for collaboration, and to play an integral role in the inception and completion of the ERIEbooks replacement Proof of Concept and the resulting project approval. Previously, she worked as conferencing manager at Gannon University, and information systems manager for the Erie Art Museum and The Erie Maritime Museum and U.S. Brig *Niagara*. Rita holds a BA in Political Science from Gannon University, and she is working towards a Master of Science in Computer and Information Science (Software Engineering Track).

COLLABORATING IN DITA: LESSONS LEARNED FROM THE 2012 STATE OF COLLABORATION SURVEY

Paul Włodarczyk, Jorsek LLC

In 2012, easyDITA conducted a survey of the current state of collaboration in information development organizations. Despite widespread DITA adoption, most review happens in PDF. Our STC Summit 2013 presentation discussed DITA-based collaboration best practices and demonstrated emerging technologies that improve the speed and effectiveness of review and collaboration.

Since the introduction of XML document technologies in the 1990s, technical communicators have adopted them for the advantages of single-sourcing, component content management, content reuse, and improved control over the localization process. DITA (the Darwin Information Typing Architecture) standardized best practices for XML document technologies and to a large extent provided for easier adoption by introducing an extensible content model, standardized publishing tools, and interoperability standards. DITA adoption is growing steadily, and most software tool vendors support this standard today. Communities of end-users have formed around DITA as well, which no doubt has contributed greatly to its widespread adoption.

Despite this, most organizations that use DITA continue to collaborate with subject matter experts (SMEs), reviewers, and approvers in non-DITA formats. The primary barrier to collaborating in DITA has been the absence of collaboration tools – including DITA editors – that can be used by non-professional authors.

In 2012, Jorsek Software, makers of easyDITA, conducted a survey of technical communicators to better understand current practices for collaboration – whether the source content is authored in DITA or in other formats, including unstructured formats like desktop publishing software (DTP).

This paper reviews the findings of that survey, discusses the advantages of DITA-based collaboration, and discusses technologies that exploit the capabilities of DITA for improved collaborative content development.

SURVEY OVERVIEW

2013 was the premier year of the State of Collaboration Survey. The objectives of the survey were to:

- Benchmark the current state of collaboration practices in technical communications organizations
- Better understand current approaches to content contribution, review, and approval
- Understand the extent to which organizations incorporate user generated content and social media methods for engaging end users in content-centered conversations
- Understand common challenges and unmet needs

The survey was promoted to technical communicators through LinkedIn, Facebook, and Yahoo groups, as well as through Twitter and other social media channels. The target participants were managers of information development teams, individual technical communicators, stakeholders in the collaboration process with technical communicators, and consultants and software vendors who serve these communities.

The survey opened on July 6, 2012, and has remained open. This report summarizes the findings through September 20, 2012. There was a total of 123 respondents, and 78 provided answers to all of the survey questions.

The survey will be repeated annually to track trends and to better understand emerging best practices and challenges.

The following sections summarize the findings of the survey. We note the number of respondents in the caption of each chart.

SURVEY FINDINGS

SURVEY DEMOGRAPHICS

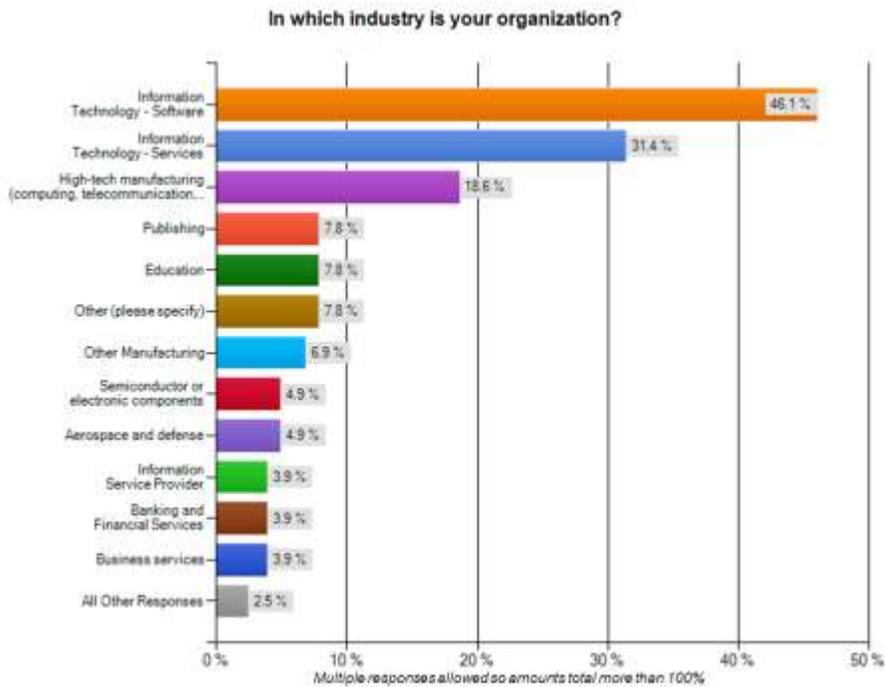


Figure 1. n=102

Figure 1 shows the majority of survey respondents were in the IT and High Tech manufacturing sectors. This is a stronghold for DITA adoption, and may represent a selection bias from the channels used to promote the survey. We see from Figure 2 that the plurality of respondents – 42% - were individual contributors that self-identified as writers or information architects. 18.6% were managers of technical communications groups.

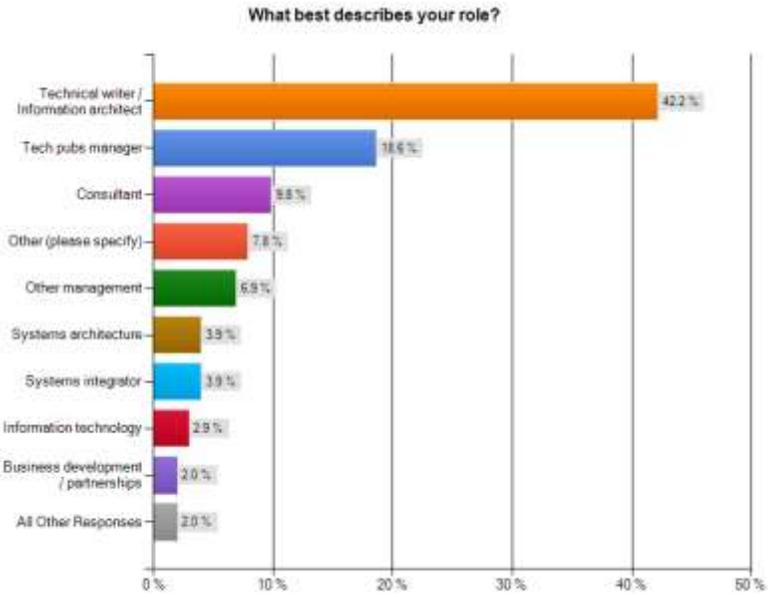


Figure 2. n=102

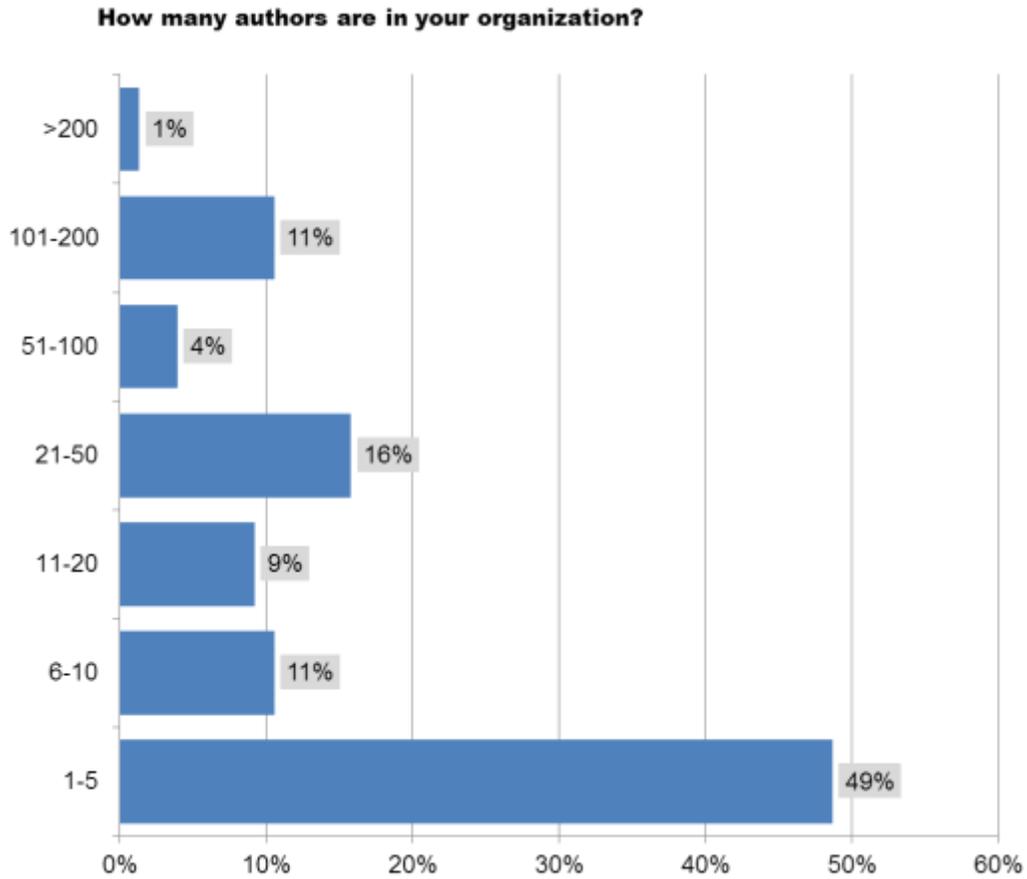


Figure 3. n=76

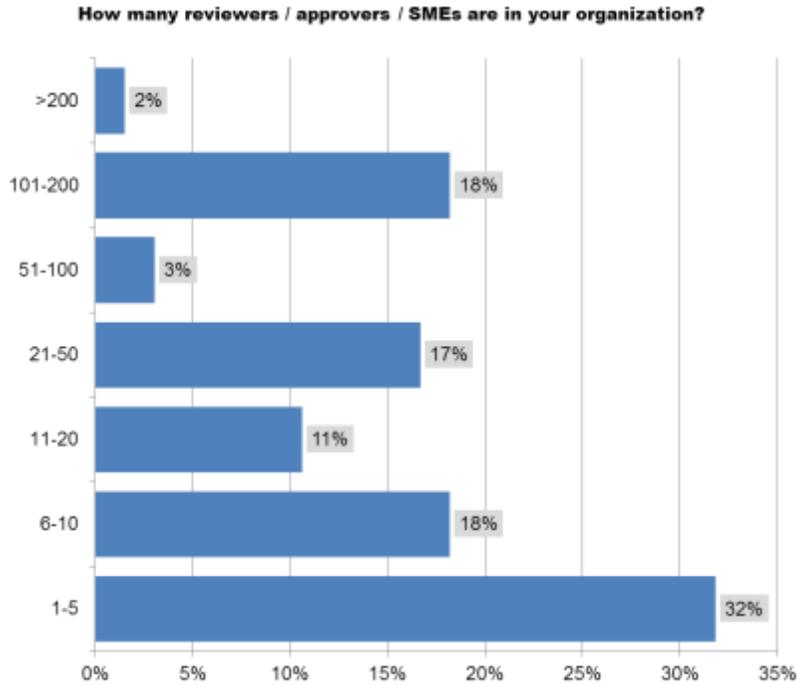


Figure 4. n=66

Organizations of a variety of sizes participated. The median organization had 6 authors, with just under half of all organizations reporting 1-5 authors (Figure 3). We also surveyed the number of reviewers (Figure 4) to understand the scope of collaboration. What we found when we analyzed the ratio of Reviewers to Authors (Figure 5) was interesting. In nearly half the organizations, there are 1-5 reviewers per author. In 23%, review is concentrated into a small group of SMEs (fewer SME reviewers than authors), suggesting review is centralized. In 30%, the ratio is greater than 5:1, highlighting that collaboration is far-reaching and spanning a number of functional areas.

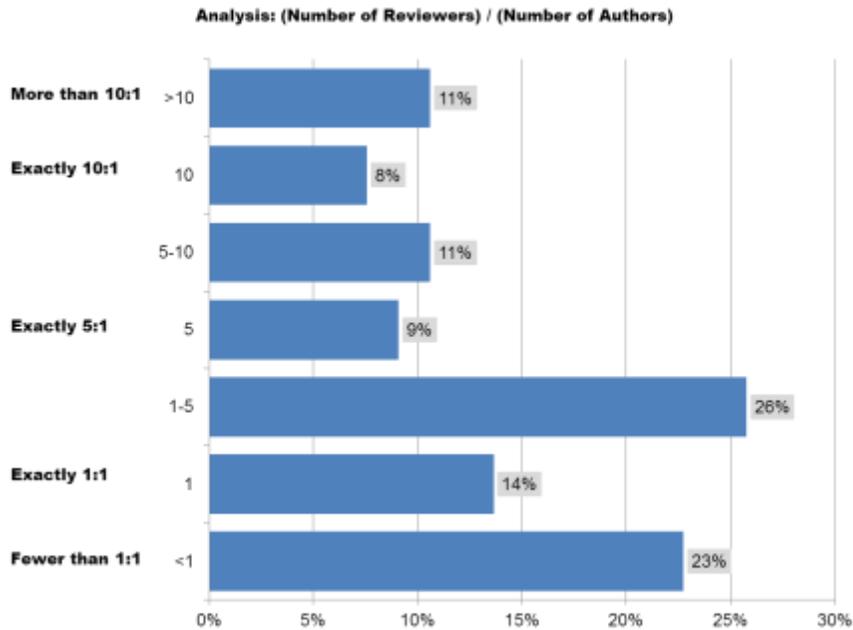


Figure 5. n=66

DELIVERABLES

We asked participants about the number and types of publications they were responsible for. About half of the respondents told us they create fewer than 50 publications annually (Figure 6). While User Guides were the most common deliverable type cited (Figure 7), we saw a variety of deliverables including web content, marketing, sales, and software engineering documents (requirements, QA).

We see in Figure 8 that 40% of respondents still deliver printed documents and 95% support PDF, however electronic delivery formats for mobile devices are growing in importance. We will continue to track the growing popularity of mobile formats in future surveys.

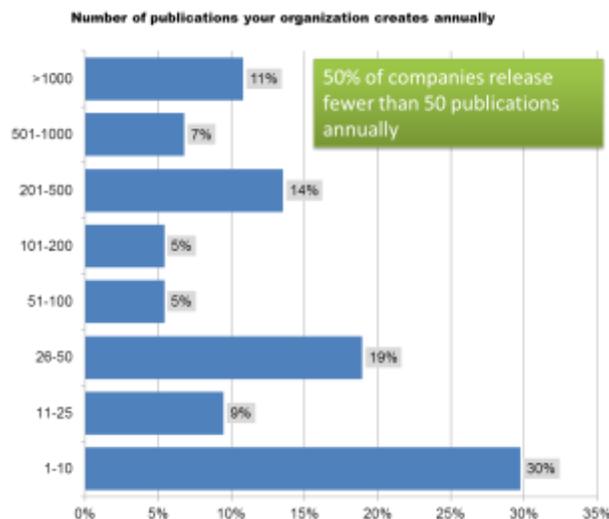


Figure 6. n=74

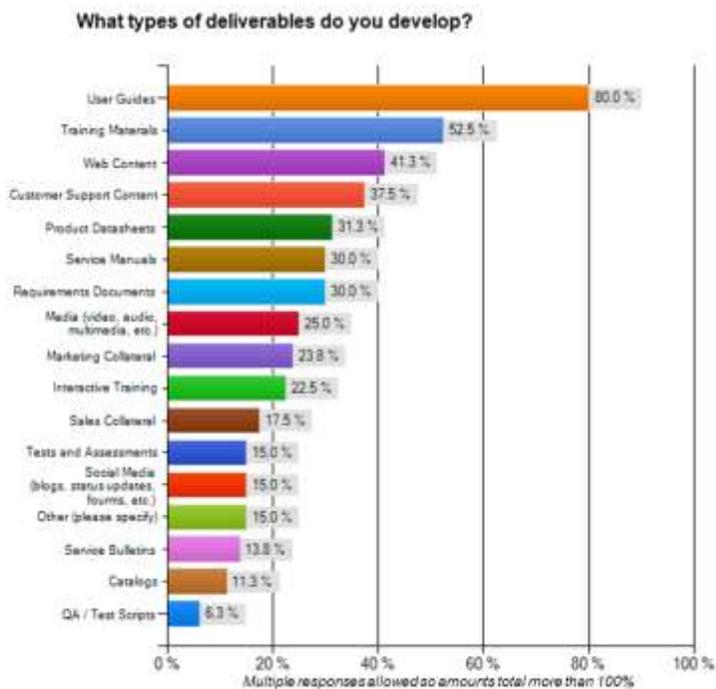


Figure 7. n=80

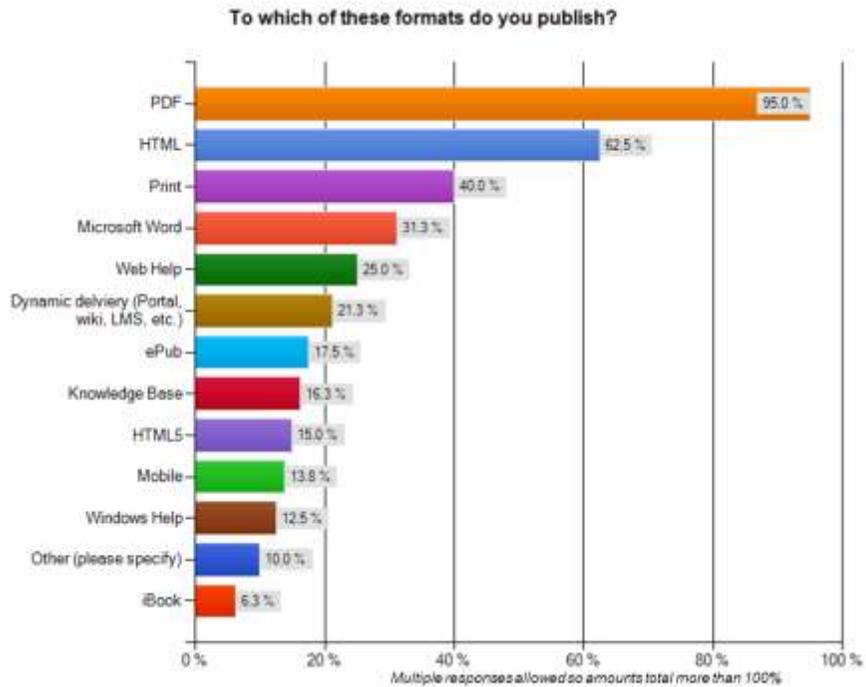


Figure 8. n=80

CONTENT MANAGEMENT AND LOCALIZATION

Over a third of survey respondents (Figure 6) do not use a CMS, and a third do. The remainder use a combination of methods for storing source documents, or use an “augmented file system” such as a source code control system for basic access controls and versioning.

While the majority of respondents do not localize (Figure 7), localization introduces unique needs for collaboration that we will examine in our upcoming 2013 survey.

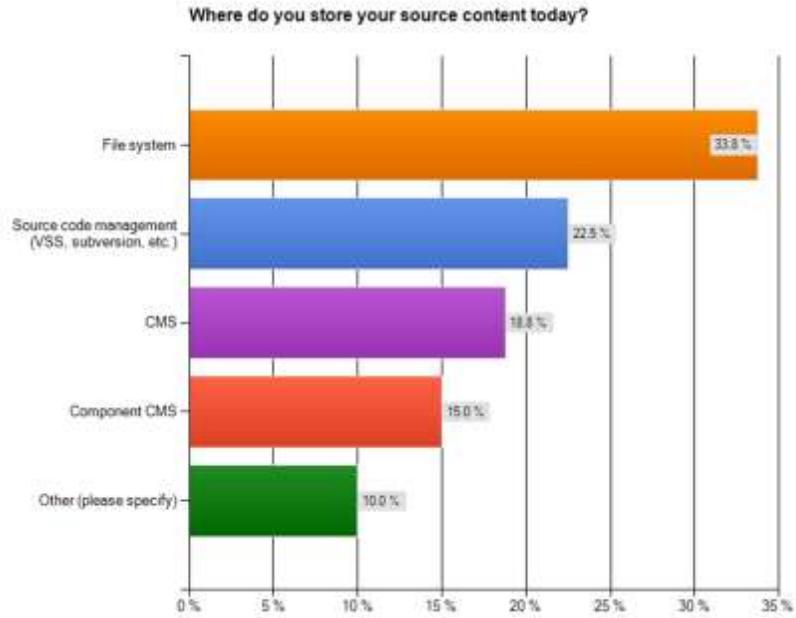


Figure 9. n=80

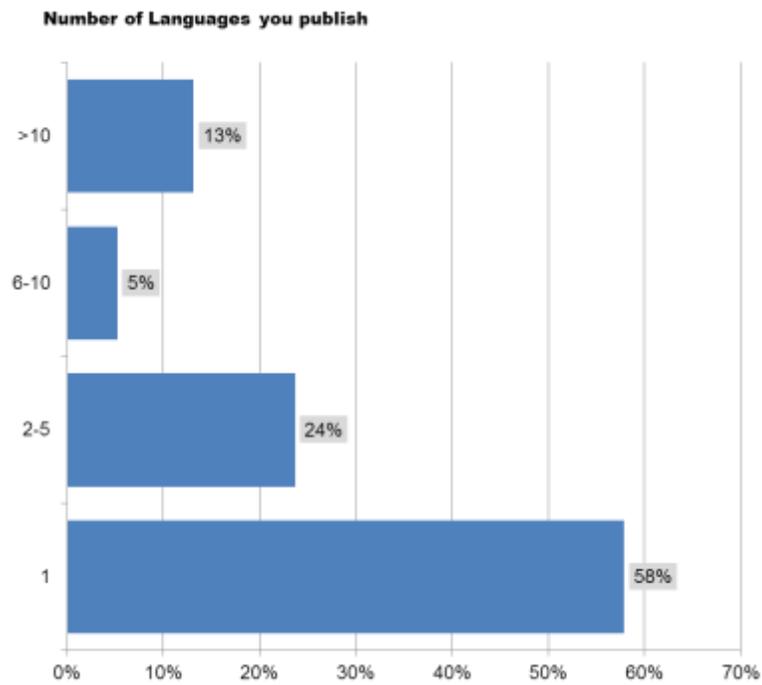


Figure 10. n=77

AUTHORING

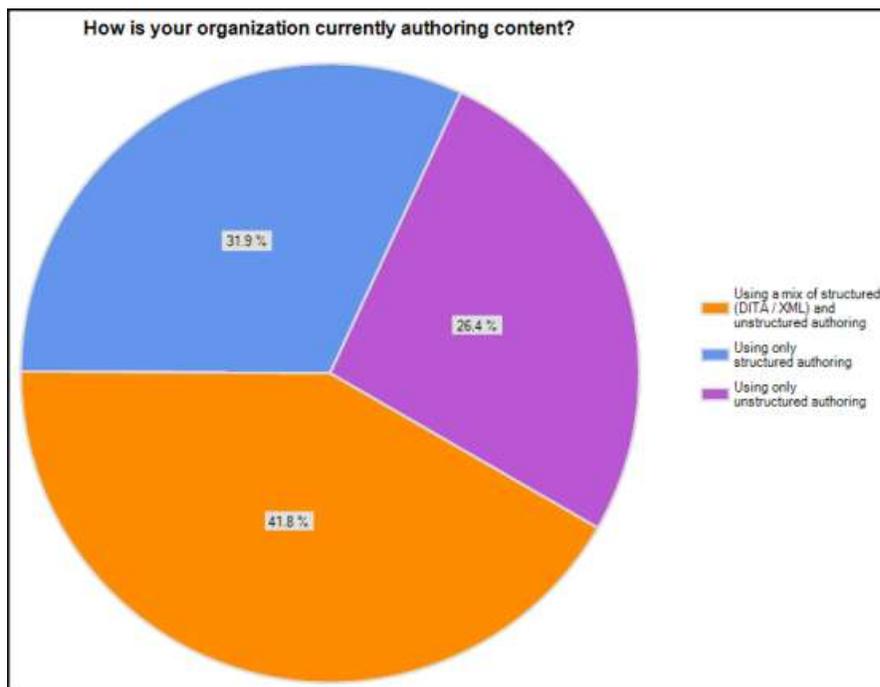


Figure 11. n=91

42% of organizations (Figure 11) use a mix of structured and unstructured authoring tools. Surprising, a full 32% use only structured authoring tools. We asked more detailed questions of the participants who use structured authoring, which we detail below.

Over 71% of all respondents use Microsoft Word, and 37.5% use FrameMaker. The ubiquitous use of Word influences how organizations approach document-based collaboration, as we observe later in this document.

DITA AND XML ADOPTION

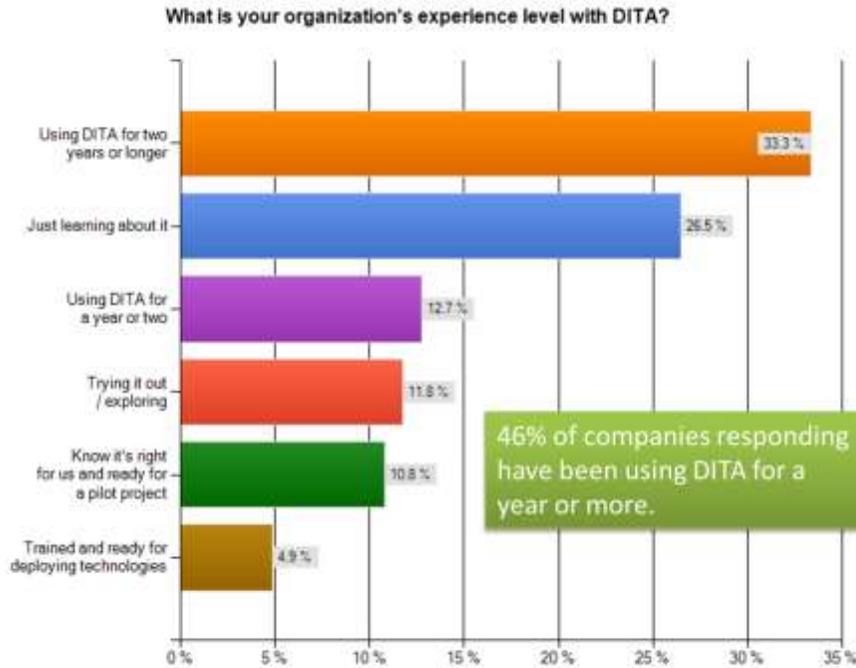


Figure 12. n=102

Our survey validated the widespread adoption of DITA (although as we noted earlier, our channels for promoting the survey reach a uniquely DITA-aware community, so the adoption rate is likely lower in the total population reflecting this selection bias). Nearly half – 46% - of respondents report using DITA for more than a year, with a full third using DITA for over two years (Figure 12). 25% are just learning about it, but the other 25% are either ready for, conducting, or completing pilots.

We asked all participants which standards they use for structured authoring (Figure 13). Again we see the dominance of the DITA standards, however 20% report using a customized XML format, and 11% use DocBook.

When we survey only those respondents using DITA for a year or more, 24% produce 100% of their deliverables in DITA. Another 26% produce more than 75% of their deliverables in DITA. This shows that in the organizations we surveyed, DITA is a primary production technology. Likewise, 46% of the organizations using DITA for a year or more report that 100% of their authors are using DITA>

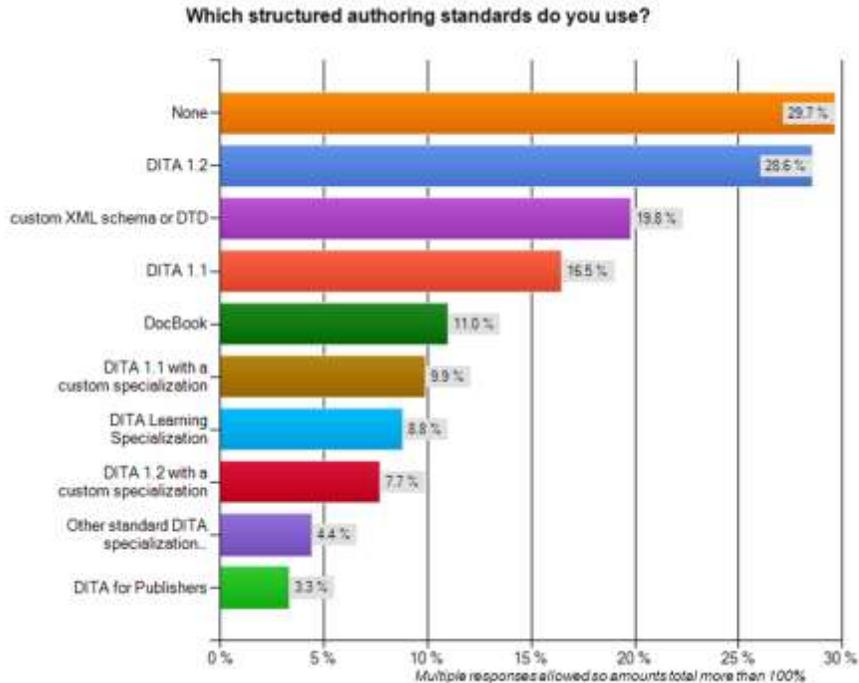


Figure 13. n=91

COLLABORATION: CONTENT CONTRIBUTION, REVIEW, AND APPROVAL

The core purpose of our survey was to better understand the state of collaboration (hence the clever name). We asked respondents how they gathered input from SMEs, and this is what they told us (Figure 14).

Over half of all respondents – 54% – report that SMEs contribute draft content to the technical communication process in the form of unstructured content (primarily Microsoft Word). For the organizations that use DITA extensively, this means copying and pasting between document formats. When we consider that nearly half of the organizations we surveyed use DITA extensively, it is surprising to see how few use DITA for co-authoring (17.5%) or contributed content (6.3%).

Our belief is that the ubiquity of Microsoft Word and email formats make them the tools of choice for collaborating with SMEs.

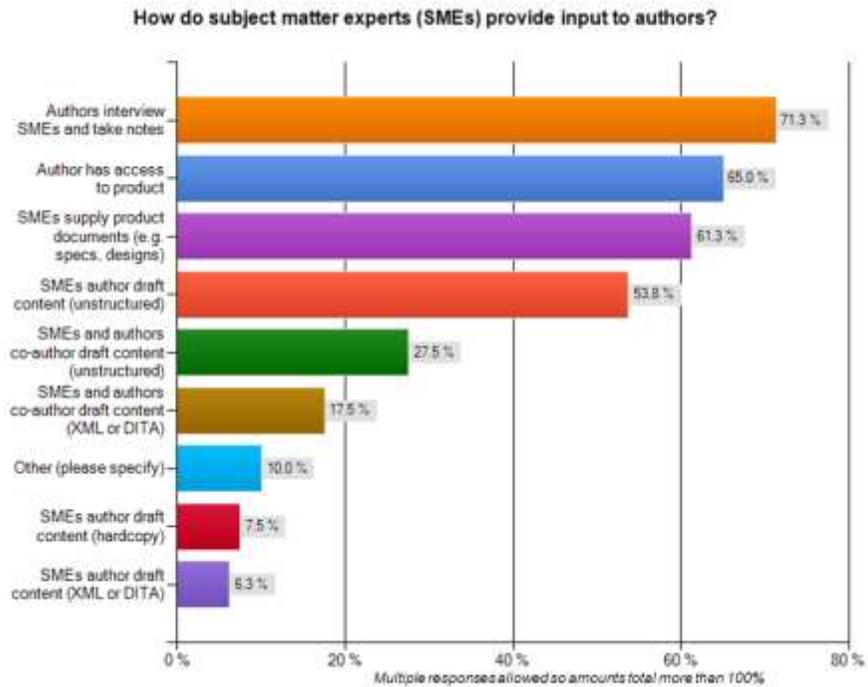


Figure 14. n=80

In order to better understand how adoption of DITA or XML structured affects collaboration, we conducted a split-sample analysis. Samples were:

- Organizations using DITA for a year or more (labeled as “Using DITA/XML >1Y” and colored in green). We consider these to represent production instances of DITA, based upon their other survey responses (% of authors using DITA, % of deliverables produced with DITA, etc.).
- All other organizations (labeled “Other” and colored in blue). Note that “Other” includes organizations that are using DITA but not at full production scale. Some are trying DITA out, and several create up to 25% of their documents in DITA/XML.

Note that in both of these groups there is a mix of structured and unstructured authoring and collaboration tools (42% use both).

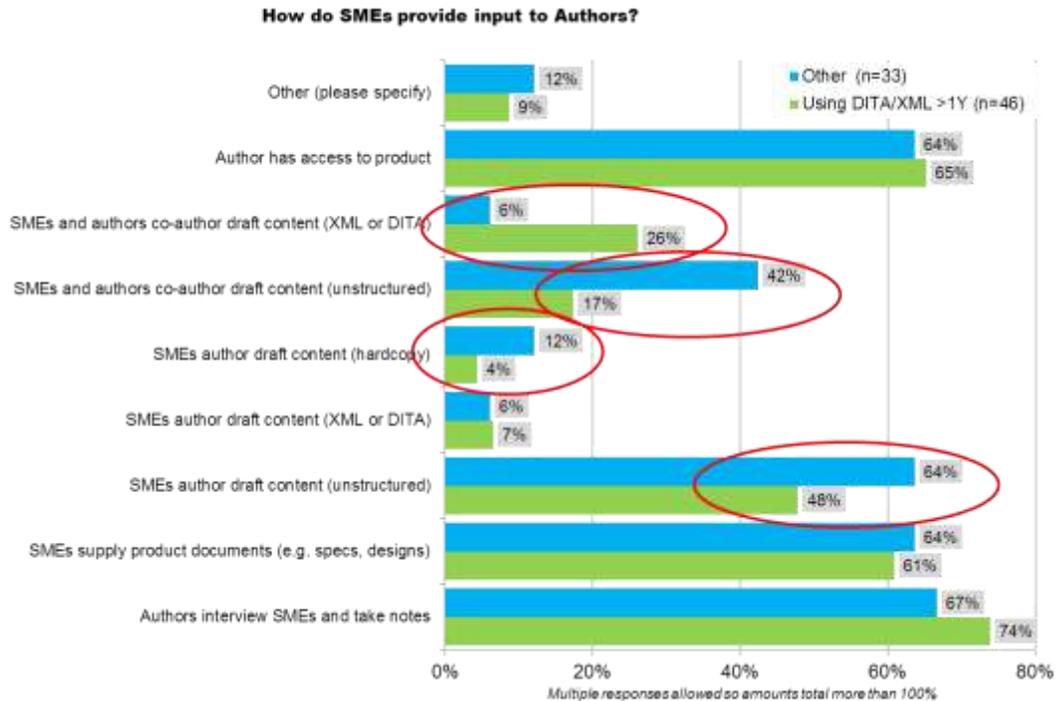


Figure 15. n=79

As we see in Figure 15, there is less co-authoring in organizations using DITA / XML for more than a year. 43% of organizations using DITA for more than a year report a combination of structure and unstructured co-authoring with SMEs. The other organizations report a combined total of 54% (with the amount unstructured being higher, of course).

Likewise, we see less content contribution from SMEs in organizations using DITA / XML for more than a year. For every format of SME drafted content (whether hardcopy, structured, or unstructured), the “other” organizations report either similar or substantially higher rates of content contribution for each format.

Does this indicate that structured authoring approaches such as DITA are a barrier to collaboration? Or does it indicate that DITA is more readily adopted in organizations that require less collaboration?

Figure 16 shows the methods used for SME review of content. As with content contribution, we can see that overall there is less collaboration going on in organizations that have been using DITA/XML for more than a year. 64% of organizations that lack a production DITA solution report that they use Word for track changes and redlines to collect reviewer comments and changes. Only 20% of those with production DITA solutions report using Word for review, probably due to the difficulty in rolling those changes back into DITA. However they have not replaced Word with DITA tools; only 26% report using DITA or XML tools to track changes and comments.

So once again we have to ask, does this indicate that structured authoring approaches such as DITA are a barrier to review? Or does it indicate that DITA is more readily adopted in organizations that require less SME review?

In either event, we must note the prevalence of PDF review for track changes, which we see adopted equally regardless of the level of DITA adoption.

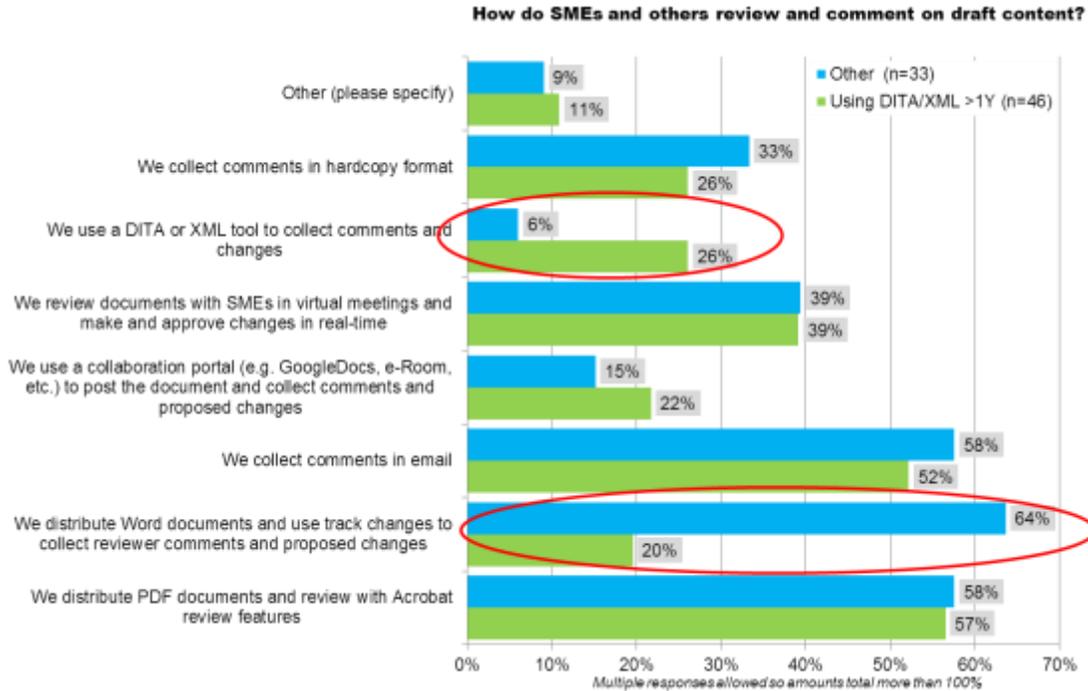


Figure 16. n=79

Review and approval of “final drafts” shows a dramatic difference in behavior between organizations with DITA in full production and those without. As we can see in Figure 17, over half of those respondents using DITA / XML for more than a year only review content that changed – whether as components or maps built just for review. This reflects the use of DITA’s component model to track components that changed and to put those into the review and approval workflow versus the entire document.

This “component-oriented” review can dramatically improve the speed and quality of content review. Reviewers and approvers require less time to review, and have less “old” material to take focus away from what changed. This in turn makes them more likely to participate in reviews and respond quickly.

Organizations with DITA in production still report that their use of PDF for review is still widespread for review and approval, though not as pervasively as for review and comment. This indicates to us the preference for Acrobat-based review is for collecting SME and reviewer comments, not just for the “immutability” of the PDF format itself.

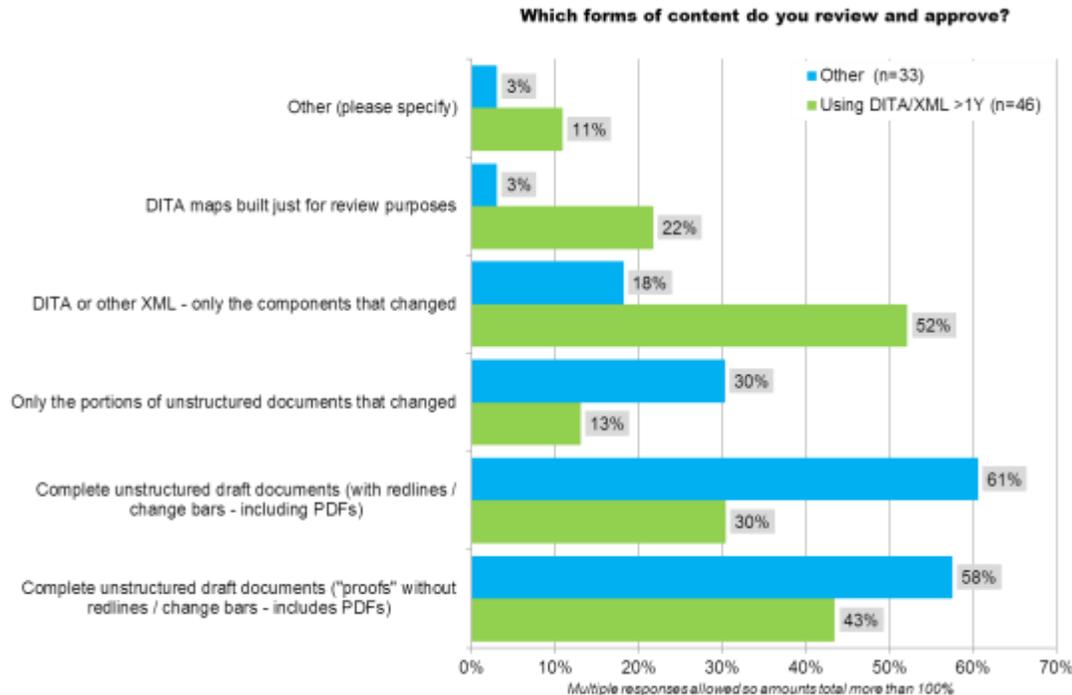


Figure 17. n=79

MAJOR ISSUES AND CHALLENGES

Continuing with our split sample analysis, we looked at how having DITA in production for a year or more affected the way organizations described the challenges they face. As we see in Figure 18, other than “rekeying SME proposed changes” (which had a slightly higher incidence in the DITA group), organizations with DITA/XML in production for more than a year report less pain overall in each of the dimensions we surveyed:

- Only 9% report record keeping is insufficient for audit purposes (versus 21% of the others)
- 54% report that reviewers do not respond in a timely fashion, versus 70% of others. We attribute this improvement to the use of component-oriented review and approval, which minimizes the amount of SME review to what changed.
- 37% report that SMEs and approvers comment on content that did not change, or was “out of the scope of the revision” – versus 55% of the others. We also attribute this improvement to the use of component-oriented review and approval, which focused the review activity on the changes.
- 46% report that SMEs complain about the time or effort to review, versus 64% for the others. Again we attribute this improvement to component-oriented review and approval.
- 54% report that review and approval delays publication, versus 70% of others. These numbers are identical to the question on responsiveness, so we can draw the conclusion that publication delays are likely due to non-responsive reviewers.
- A reduction of nearly half – 35% versus 64% of the others – in reports that authors have to reconcile conflicting change requests.

Which issues does your organization face for review and comment?

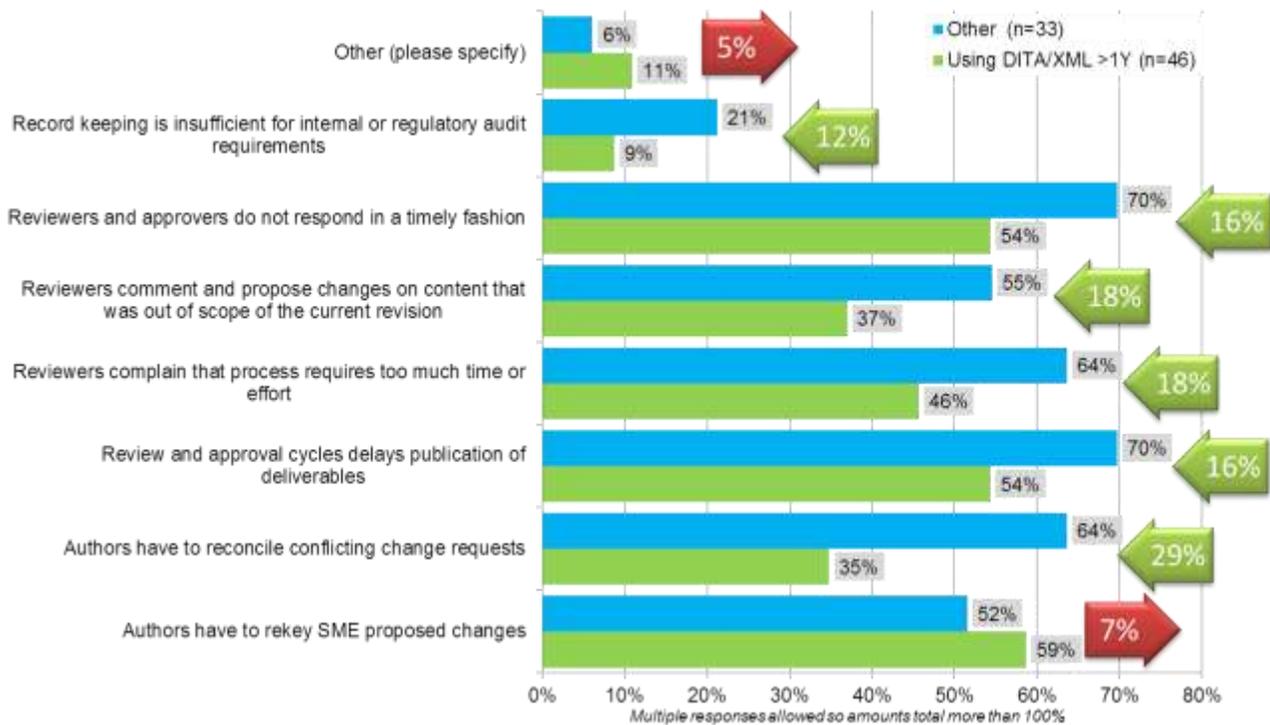


Figure 18. n=79

We attribute the higher level of pain around rekeying SME proposed changes as being due to – until recently – the lack of good tools for SMEs to contribute content in a native XML or DITA format.

WHAT IS THE ONE THING YOU WOULD CHANGE IF YOU COULD?

We concluded the survey with an open ended question, “If there was one thing you could do to improve collaboration in your organization, what would it be?” We sorted the verbatim comments into two categories – Process and Tools – and present many of them here as submitted, with the role of the respondent.

PROCESS IMPROVEMENTS FOR COLLABORATION

Product Lifecycle

- “[Info dev] included earlier in the product development cycle” – Tech Writer
- “Better notification from developers of product changes” – Tech Writer
- “Better communication about new features that are coming and how they affect current documentation” – Tech Writer
- “Make spec documents from SMEs clear enough that a common controlled vocabulary can be created, and that spec clearly indicates task flows etc.” – Tech Writer
- “Prevent customer engineering from being a roadblock in getting information about customer goals and needs” – Tech Writer

Review

- “Convince reviewers that modular documentation actually means not having to review entire books.” – Tech Pubs Manager
- “More timely responses from SMEs.” – Tech Writer
- “Train SMEs on collaboration techniques.” – Tech Pubs Manager

End-user Involvement

- “Get more customer feedback into the document development / improvement process.” – Tech Pubs Manager
- “We would like to know how the end users are using our documents.” – Tech Writer

Support

- “Have better coordination with Project Support (the people who actually interact with the customers!)” – Tech Writer

TOOLS IMPROVEMENTS FOR COLLABORATION

Common Tools

- “An affordable minimal-DITA browser editor for the masses.” – DITA Consultant
- “All parties using the same tools to create, review, and manage content.” – Tech Pubs Manager
- “XML/DITA implementation, everyone on same tooling/process” – Tech Writer
- “More collaboration that is viewable to all” – Tech Writer
- “Offer an easy [DITA] authoring tool ... Developers could draft procedures and QA could write test plans in DITA rather than unstructured documents.” – Tech Writer

Change Tracking

- “See a history of changes within a document, rather than by comparison of two versions in the CMS” – Other Management

End-user Involvement

- “Provide a way for customers to give direct input to our content” – Tech Pubs Manager
- “Move to a web-based content distribution model and enable feedback from end users directly.” – Content Production
- “Incorporate social media in the products and their help docs to capture the end-user’s feedback” – Tech Writer

SURVEY CONCLUSIONS

There are significant differences in collaboration approaches and challenges between organizations that have been using DITA / XML for more than one year – and have DITA in production – and others:

- There appears to be less direct content contribution from SMEs in organizations using with production DITA solutions. We suspect this is due to limited access to or usability of DITA / XML tools for SMEs

– and that DITA tools represent a barrier to collaboration. It is also possible that the lack of good collaboration tools has held back organizations that rely on SME input from adopting DITA. If this is true, the “early adopters” of DITA may well be those that collaborated less in the first place. We hope to tease out the answer in our 2013 survey.

- Most organizations do not appear to have found a suitable DITA / XML replacement for Word-based collaboration for contributed content or track changes. The feature set in Word has set expectations for how DITA or XML review and approval should take place.
- Use of PDF-based review is equally widespread for SME review and comment in all organizations. PDF is less likely to be used for final deliverables approval in organizations using DITA / XML for over a year. As Word has “set the standard” for collaborative authoring, so has PDF set the standard for a review and approval format.
- Organizations using DITA / XML for more than a year have overall lower pain in the areas of content review, probably due to the practice of component-oriented review of only the content that changed.
- Organizations using DITA / XML need better tools for collaboration as summarized in our write-in comments.

SO, HOW DO I COLLABORATE IN DITA?

So what would “better DITA tools” look like? We get a good sense of this from the “What one thing would you change...” question, as well as the pattern of current tool use for content contribution, review, and approval. Here are the conclusions we drew:

- DITA authoring tools that are easy to use by non-professional authors such as SMEs to contribute draft content.
- DITA co-authoring tools that behave like Microsoft Word, so authors that are collaboratively authoring can track each other’s changes.
- DITA review and comment tools that provide balloon or post-it-note style comments like Word, Acrobat, or Google Docs.
- DITA review and comment tools that enable changes to be reconciled and collected easily, and incorporated into the revised source document without rekeying.
- DITA review and approval tools that make it simple to route “final drafts” to approvers using DITA components and “review maps” to focus approvers on only what changed.

While these tools were relatively absent from the marketplace until recently, there are affordable tools available now that provide these capabilities. Most of the major DITA CMS vendors (including us) have released review and approval solutions. There are also new DITA editors coming on the market that offer easy-to-use Word-like (or even native Word) interfaces for content contribution, review, and approval. Some of these support the full feature set of DITA, others support a subset of the DITA vocabulary.

We can confidently say that there is a tool available today that will help your team collaborate in DITA. Look to the DITA CMS vendors for solutions, or if you already have a CMS or CCMS, look for collaboration tools – especially easy to use DITA editors with good collaboration features – that will integrate with your existing CMS.

RESOURCES

Jorsek Software (easyDITA), "The State of Collaboration - 2012 Survey." easyDITA.com - <http://ow.ly/cuyVU>

Jorsek Software (easyDITA), Recorded Webcast: "Insight Series: The State of Collaboration 2012 - Survey Findings." easyDITA.com - <https://www4.gotomeeting.com/register/983938351>

AUTHOR CONTACT INFORMATION

Paul Wlodarczyk
CEO
Jorsek Software – makers of easyDITA
125 Tech Park Drive
Suite 2160
Rochester, NY 14623
Direct 585.298.9538
Toll-free 877.492.2960

AUTHOR BIOGRAPHY

Paul Wlodarczyk is CEO of Jorsek Software, makers of easyDITA. Paul is passionate about content lifecycles - business processes and workflows that span the collection, collaboration, authoring, assembly, review, localization, publishing, reuse, classification, and search of unstructured content. Paul brings over 25 years' experience in content lifecycle operations, consulting, and software development, with expertise in the areas of enterprise content management, knowledge management, content classification, technical publishing, localization, collaboration, user interface design, learning technologies, and information worker productivity.

Paul supports clients and customers in the manufacturing, life sciences, high tech, aerospace, retail, energy, capital equipment, insurance, and publishing industries. Paul is a frequent speaker at industry events and contributing writer for publications by various organizations. Paul holds an MBA from the William E. Simon School of Business, and a BA in Psychology from the University of Rochester in Rochester, New York.

COLLABORATIVE SINGLE-SOURCE DEVELOPMENT OF DOCUMENTATION AND TRAINING

R. N. Homer Christensen

Training materials and documentation have long been considered separate, often isolated by departments and budgets, workflows, managers, and designers. Even the tools we use are often different. Because of this, the products are often produced through costly (and parallel) development cycles competing for the same resources. In many projects, there is a significant duplication of work to complete each deliverable. Every developer, for each deliverable, must learn the end user, discover the functionality, uncover the range of tasks, and determine how learners will use the product to accomplish work tasks.

Worse, as products mature, maintenance becomes an issue. As training ages, it begins to diverge from documentation, such as online help, user manuals, and other supporting material. In an era of receding budgets, often only one product line is chosen to be updated for subsequent releases. It is often a given that after the instructional developers and trainers walk away from a software release training effort, that the course languishes without updates.

Add in the challenge of writing materials under a tight development timeframe while the software is being developed and tested, and it is easy to see how quickly errors can creep into various documents and corrections or revisions to one set of materials are missed in others.

Training materials have a short life-span. They're intended to be used aggressively to support a roll-out of software or adoption of a new business methodology. But training doesn't end after the last webinar or an instructor shuts down the projector. Learning happens throughout time, whether that is to learn a new feature, refresh one's memory, or deepen mastery. Likewise, documentation materials need to be seen as effective, long-term training tools that support learning and promote mastery at a very low cost.

To help achieve this goal, our team created a development methodology that took advantage of each participant's individual strengths so that we could focus on our shared goal of assisting users as they learn and master software, hardware, and concepts. We used a single-sourcing workflow so that all client deliverables (except the slideshow) were produced from the same collection of topics. Our journey showed us that mindsets and habits of developers and reviewers were as much of a challenge as developing the methodology and setting up the tools to perform the work. Fortunately, we had client approval, management support, and a team of willing developers who not only graciously endured the starts and stops as we learned our way, but added their own brilliance to the workflow and methodology.

As a result, we were able to complete simultaneous development of an integrated document set that our client will be able to maintain with less work. Our development effort was reduced greatly—somewhere in the range of 20 to 30 percent—allowing us to complete the project on time with a reduced staff. This paper describes that journey.

BACKGROUND

This particular project was an enterprise-wide rollout of a web-based software system that fundamentally changed how state employees input and accessed information at the world's largest prison system¹. It was designed to be introduced into production in three massive waves. The new software replaced several legacy

mainframe systems. The audience for the training was about 50,000 state workers with a range of computer experience from significant (primarily analysts and administrators) to none (never touched a computer other than a smart phone or the ATM), and a learning attitude of excited to openly hostile and skeptical. (Several past programs failed prior to implementation or were scrapped after a brief time in service.) The hardware base was aging PCs on some desks in a locked-down internal network, and none within the prison institutions outside of the administration.

The state client initially wanted a blended solution of e-learning, instructor-led training (ILT), and documentation, including online help as well as a printed user manual. E-learning developers used an in-house proprietary LMS tool, the Instructional Designers (IDs) used Microsoft Word, and the documentation team used Madcap Software's Flare.

The development and delivery of the first wave went better than anyone had expected. Our materials were beautiful and well-written, the deployment team did an amazing job in the field preparing end users, and the state's trainers delivered webinars to a surprisingly receptive audience. The rollout was deemed an overwhelming success and was hailed in several state and industry conferences.

Among the notes from our postmortem review of phase 1, were the following:

- **Duplication of development effort.**

The IDs spent a considerable amount of time writing procedures that the documentation team had already written. The procedures for training, however, were nestled in demos, lessons, and exercises and were customized for the context. Likewise, the documentation team had introductions and process overviews that could have been leveraged from the ILT materials.

- **Communication during crunch time.**

As we neared the end of the development cycle, some final changes made to the software and communicated by our subject matter experts (SMEs) did not necessarily make it to all teams.

- **Reviewer exhaustion.**

Reviews focused on training materials, not documentation. So while each lesson received a full week or more to review, the entire documentation set was reviewed in 10 days. ILT materials had substantial revisions, while the 600-page documentation set received only 8 comments or changes.

- **Change Request updates.**

Budget constraints dictated that only the online help system be updated to include the numerous change requests and software updates that came in during end-user training or the initial months after release.

- **Lengthy production cycles.**

Merging multiple Word documents into a master without breaking numbered lists or introducing errors into the Normal template was challenging. Producing and proofing facilitator and student versions from those same documents was a chore. Exporting to PDF format was yet another step.

- **Monolithic training materials.**

Before we began development, the client determined the delivery method. Erratic budgets forced a change in the delivery method weeks before training began but after development was in final review.

- **Maintenance of delivered materials.**

After the client took possession of the training materials, they discovered gaps in their knowledge of working with Microsoft Word (especially templates) and struggled with incorporating updates to the

text and screenshots within lessons. They quickly realized that to keep the materials current or port it to their training academy (for recruits) would entail significant effort. They asked us to see if we could devise a way to reduce their maintenance.

CHALLENGES

I first proposed the idea of a collaborative single-source development approach (CSSDA) after the release of the first wave of training and documentation. The reaction I received was surprising to me, but perhaps not unexpected.

- **Mindset of Trainers/IDs.**

What surprised me most was the resistance of the instructional designers. After the release of the first wave, the client frantically cobbled together job aids for workers on how to complete tasks. When I showed them the information was already available in contextual online help, they were surprised. The online help had no value to the trainers and wasn't mentioned in class.

- **Topic-based writing.**

When we were finally given approval to explore single-sourcing, we faced the reality that the IDs were more familiar with working in a narrative, start-to-finish document. Word suited their development style (they could see the entire lesson in one document), but the idea of topic-based writing was difficult to get used to. Flare and other tools that use a single-source approach of reusable content use a more structured style that is a bit daunting and feels choppy and confusing to one not used to thinking in chunks.

- **Available tools.**

We were already using Flare for the online help and printed documentation, so it was a given that we'd continue with it. Madcap Software developed a tool for editing and reviewing materials (*X-Edit*, a free tool which was re-branded and given a price tag as *Contributor*), but it was not up to the task of acting as a writing client. We had to abandon our first development trial when the initial version of Contributor wouldn't do what we were asking. Our fallback was to continue to use Word, but to develop the TOC in Flare, providing the already-written procedures amidst the blank narrative topics and to re-import the topics in Flare for cleanup. This, too, was too much effort for what we saved.

Within a few months, however, I was assigned to lead both training and documentation teams, Madcap Software released a new version of Contributor that looked like it might just work, and a new management team approved and supported another trial. Even more fortuitous, the client requested that we look into a way that would merge the training and online help to make it easier for them to maintain the materials after we left. It was perfect timing because we also lost half of our staff due to budget cuts. Fortunately, we still had three Flare experts on staff.

WORKFLOW

One of the first tasks was to develop the workflow.

Our development methodology is aligned with the ADDIE model (analysis, design, development, implementation, and evaluation), and when we begin development of a new software module, we first complete the task analysis (what can be done with the software), audience analysis (who will use the software), and the needs analysis (what training is needed to ensure that the right people learn the right

tasks). A technical writer (TW) performs the task analysis and this becomes the basis for our single-source project.

Our collaborative workflow

Once the task analysis is complete, the TW moves quickly from design (creating the TOC) into development and begins writing the online help topics, making an educated guess as the topics most likely to be included in the first lesson. It's an iterative development cycle, however, as the ID meets with the client to determine the audience and needs analyses.

Because the needs analysis is the basis of the training design, the time to reach consensus allows the TW to make significant progress in authoring the procedures before the lesson structure emerges. As the design takes shape, the TW knows which topics will be needed and in which order and can author them before the ID begins development of the lesson that contains the topics.

All deliverables are managed from the Flare project. After the training design is complete, the TW works with the assigned ID to develop a table of contents. The TOC includes new topics specifically for training and the already-written procedures that support them. The TW then exports one lesson's topics to the ID for development.

When the ID has finished development for the lesson, the ID returns the topic package to the TW, who imports it back into the Flare project and exports the review documents. The TW also sends the ID the next lesson's topics.

This cycle continues until all lessons are developed. A few items deserve mention:

- The TW writes procedures; the ID authors introductions, summaries, and narratives.
- Small review comments are put in by the TW directly into the Flare topics; more extensive review comments require the topic package to be sent to the ID for revision.
- All deliverables are exported from Flare.
- Clients find the Word export easier to review (with track changes), and the PDF export more difficult. This adds some risks and additional work during reviews, but reviews were accomplished more quickly.
- We set up review packages based on the training design, with the lesson, exercise, job aids, and PowerPoint going to review together. We also included all related topics that were written for online help but were not used in the lesson. This gave reviewers confidence that all topics for that area were covered somewhere, and it allowed for efficient reviews.

TOOLS

The tools we used were a big part of the implementation and its success.

- **Flare and Contributor.**

We were already using Flare and I continue to be impressed with its ability to do what it promises to do. We used the new release of Contributor, which worked but was a little problematic. Any issues centered around the fact that it was designed as a reviewing tool, not an authoring client. We were asking it to stretch and bend, and it did that admirably, but we still had some pain points. Madcap Software's technical support helped when we got stuck and gave some tips that helped. That said, it allowed our IDs to author in XML in an environment that resembled their beloved Word.

- **Cascading Style Sheets (CSS).**

Getting away from Word's problematic implementation of styles was one of the primary goals I had. (I was responsible for the final reviews and production of the hand-off deliverables.) CSS allowed me to lock down the styles and set up a clean and easy-to-use style sheet that was used for all deliverables. We kept it simple. However, we could also customize the way the output appeared where needed by using media definitions, such as for online help, lessons, exercises, job aids, and so on.

For the most part, developers used the ubiquitous <p> tag most of the time. Much of the styling was determined by the <div> block that enclosed it. We set up <div>s for notes, cautions, procedures, before-you-begin text, and so on. This made it easy to clean up the ID-created content as they became familiar with using the styles.

- **Tables of contents, targets, and page layouts.**

We set up unique TOCs for each lesson, exercise, and job aid, and we linked those TOCs for each deliverable document. Delineating the TOCs to this level allowed us to quickly produce review packages for each lesson, so that reviewers could look at a complete set of materials for a topic set while we developed the next set. This also allowed us to quickly shift the order of lessons, and we could remove or include TOCs for customized documents for a specific set of learners.

Flare uses target files to define each document. Target definitions specify which TOC to use, what the output will be (Word, Online Help, PDF), as well as variables, CSS, page layouts and so on. We created targets for each deliverable and additional, separate targets for reviews, lessons, exercises, and OLH-specific topics.

Variables used in the page layouts supplied the version number, publish date, and other header and footer text. Because we had only one client for this document, we did not use variables within the text of the topics.

- **Condition tags.**

The ability to set conditions to hide or show specific blocks of text was key to making this work. We decided early on that for the student and exercise guides, we would hide the introductions that began each online help topic. This allowed us to write targeted narratives that introduced each procedure within the context of the scenario.

We set up condition tags for each deliverable, as well as exclusionary tags. This gave us a way to include or exclude text that was specific to a deliverable as well as that which was patently not applicable.

EXECUTION

Our model used the online help as the basis for the training and documentation, and so we needed a way to integrate the help with the training in as seamless way as possible. We accomplished this in two ways:

- **Connector Topics.**

We developed connector topics, the narrative topics that would be used only in training and provide the context for the procedures. All text on the connector topic was used in the training materials, and each training deliverable contained their own, specific topics. These topics tended to be brief. They contained a short narrative (*Now that you've added an offender, you can move the offender to a*

housing unit) along with cautions, notes, and a before-you-begin segment that ensure that the learner was looking at the same starting screen.

Connector topics are almost always paired up with a related procedure from the online help topic.

- **Expanded Help.**

We made the help slightly more *lush*. Our online help was designed to assist a user coming from anywhere in the system to complete it without referring to another, unless it was one in a series of connected topics for a process. By design, that made it more generic.

While we wanted to continue the directness and brevity of help topics, we realized that we needed to ensure that there was some context as well as options listed in the procedure that would benefit a first-time user but not slow down an experienced one. For example, when a drop-down selection determined a significant change in the end result, we added a note to indicate the consequences of each choice, or created additional, specific topics that applied to the selection.

Our context-sensitive help delivered the viewer to a screen-level topic. That topic described the primary uses of the screen as well as had links to related conceptual and task topics. In all related topics, we applied condition tags to hide the title and introductory text, images, and (frequently) the navigation instructions that described how to view the first screen in the procedure.

We moved that information to the connector topics so that we could customize it. For example, if a task topic in the online help was titled *Preparing a Detainer Inquiry*, the title for the connector topic would be specific to the narrative: *Preparing a Detainer Inquiry to Law Enforcement*. This allowed us to use the same procedure topic multiple times and keep the narrative flow.

Our limitation was that we could not change the text of the procedure. We could only hide or add text that applied to all instances where we used the topic, whether in the student, facilitator, or exercise guides.

PRODUCTION ISSUES

When we started, we felt that we had thought through it fairly well. But of course there were some surprises. Some were in our control; others were not. These included:

- **Flare-Contributor Review Packages.**

The review packages turned out to be more sensitive than we anticipated. We ended up with a few cases where the ID had to cut-and-paste their changes into a new review package for a variety of reasons. Some undoubtedly were operator error; some just the nature of the beast. Madcap Software kindly supplied the help of one of their engineers on a conference call and walkthrough. We experienced fewer problems after that.

In addition, Contributor offered no way to preview a file or a collection of topics. This slowed down development until the developers became more accustomed to the format. In the short term, they returned their packages weekly (or more frequently) to have the TW import and compile the new topics.

- **Thinking in Chunks.**

It became apparent early on that the technical writers were much more accustomed to chunking information into small, reusable content. The IDs were less so. To compound the difficulty, the TWs controlled the Flare project and the IDs typically received a review package with skeleton topics. It

took a while to understand that each topic was a separate file, a topic file might be (although rarely was) as brief as a paragraph, the outline determined the order, topic files might be reused even within the same lesson, and that procedure topics were not to be touched, with few exceptions. We invested a decent amount of effort in developing this skill set for them. Once they were familiar with the concept, saw the benefit, and understood the limitations, they did very well.

- **Customization.**

Our IDs had to refrain from customizing the supplied procedures for the lesson. Their natural tendency was to craft it to fit the narrative and the scenario data. Since multiple documents used the same procedure, the IDs could only hide existing information in a procedure by using condition tags.

- **Staffing.**

Our staff was cut in half quite suddenly.

- **File Control.**

Our client operates in an ultra-secure, locked-down IT environment, which gave us numerous challenges in managing file check-in and check-out. We could not install any source control and, the SharePoint interface wouldn't work for our environment. So we instituted old-school management techniques and used an interim isolated project on a shared network server to synch our files within our assigned folders. We had to limit renaming or moving files to avoid clogging up the project with duplicate, unused files.

We created TOCs for each lesson, exercise, and online-help only review file, and for each of these, we also created TOCs for PDF and Word targets. So an 11-lesson course might have 66 targets just for the lesson reviews.

Our tracking sheets proved to be invaluable, and these also allowed us to report metrics of how many topics were written or remained to be developed, the average number of topics per week, and the number of development weeks left.

- **Review comments.**

Although Contributor is intended as a review tool, it was not practical for the client to install it and learn how to work with it. The client preferred to mark up Word documents, and so after comparing and merging multiple review documents into one, catching all comments, text additions/deletions, and revisions was a tedious effort.

- **Procedure Tables.**

Because we needed to re-use procedures and couldn't completely customize each, we ended up using a table that listed the fields and the entries the student would make to complete the lesson or exercise. We made use of a div that introduced the information with a caveat that not all steps in a procedure would need to be done, but everything in the table must be entered to achieve the desired result. This table included a column for notes, where we could explain where the information came from (if supplied as a default from an earlier or related action), where to find it (if entering information from a paper form), or why it was important.

The use of tables is a bit clunky for my taste, but it was the best solution we were able to find given our constraints. We have it down as an item to explore for improvement. The single-sourcing of topics was a client priority and tables allowed that to happen.

We experimented in a few places with placing part of a procedure into a *snippet*—a reusable chunk of formatted text—and then customizing the next steps to make the online help topic distinct from the

lesson or exercise. Where it worked, it worked very well, but employing snippets at a granular level reduces the savings and increases the effort to track and manage the information.

- **Client Issues.**

Just as we had to reorient our IDs and TWs to the new method, we had to devote some time to the client members who were not part of the decision makers who asked us to develop this method. Reviews often had comments about how a step in the procedure was not necessary for this particular task. We made some adjustments (including a clearer and more targeted note in the information table that preceded each procedure) and reminded them gently about the new look to the materials.

Eventually they accepted and even warmed up to the new approach, but it is clear that the benefit is more with the development, review, and maintenance of materials.

The delivery will no doubt require a similar discussion with the students.

DELIGHTS

A corollary to the *Law of Unintended Consequences* should be the *Law of Unexpected Delights*. The benefits of using this collaborative approach began manifesting immediately and increased as we used it more.

- **Efficiency.**

One of the easiest gains to see was that our team was much more efficient. Once we had a baseline of understanding of how to work with the files, the IDs quickly saw that the effort to develop the lesson was greatly reduced. Rather than discover the task steps, they could focus on developing realistic scenarios and construct a narrative flow that supported their learning objectives.

- **Flexibility.**

Working in chunks with reusable information increased our flexibility and resource leveling. We could assign a second ID to help complete a lesson if the review deadline was rapidly approaching. Because the structure of the exercise was already completed in Flare, we could copy the files and TOC from the lesson and focus on the changes to the narrative and information.

Exercises took days instead of weeks to produce. We were also able to create multiple related exercises that the students could use as practice in the time after training and before go-live, or any time in the future to learn a new skill or remind them how to accomplish a specific process.

- **Accuracy.**

We developed documentation and training on a software product that was still in active development and which exited user acceptance testing only weeks before going live to the client. This was always a bane for development because undocumented features and corrections would slip in at the last minute and were often undetected.

In this single-source model, each time we reviewed or validated the lessons or exercises, we simultaneously reviewed and validated the online help topics. The essential information was as current as the last review, which was a vast improvement over the previous method.

- **Unity.**

The collaboration between teams merged us from two *silos* into one cohesive team. Where before the IDs lead the effort, often to the exclusion of documentation team members, each now more clearly saw the relationship between the training products we delivered. The legacy of a previous

management team that had left our team fractured was quickly replaced by an internal model of cooperation and support. Both upper management and our client recognized that our team was a high-performing group of professionals.

- **Reviews.**

The client reviews were now much more thorough and timely. In the past, the reviews were scheduled at the end of the development cycle, when it was too late to make structural or content changes. Reviews focused on the training materials and all but ignored the online help.

Now we are able to deliver chunked reviews that contain the lesson, exercises, all related online help topics, and a slide deck in one review package. Because of the reduced size of the review package, we can turn the review and revisions around in one to two weeks. Early in the project the client can identify the business process topics they need to integrate and develop them at a less frenetic pace.

Our final client approval is also shortened because each individual part has been reviewed, approved, and accepted, and so the final approval focuses on ensuring that all parts were assembled correctly and delivered in the proper format.

VALUE-ADDS

Other benefits that were unexpected included these:

- The client experimented with creating videos for procedures within the online help. Because of the shared nature of our topics, links to those videos also appear in the final documents delivered as PDFs.
- Online help is finally part of the active training. In the past, the client's students unknowingly duplicated much of the online help by creating job aids. Training happened for one or two days, a month or two in advance of the software being live. Workers are held to high standards at a job with potentially severe consequences and in the prior releases they often felt unsupported. Now they know that the information is there for them when they need it. When they read it, it will be familiar to them as they will have seen it in class.
- We are able to refer to online help in procedures where tasks had appeared in the same lesson or exercise. This not only saved time and materials, but encouraged students to explore the online help.
- We can quickly create training and step-table job aids to use in class and for ongoing support.

SUMMARY

The collaborative single-source development approach (CSSDA) worked well for us for this client and project. We are in the process of refining it and hope to expand its use as we make it more seamless to the end-user and intuitive for the developers.

¹Due to reductions in personnel and releases and transfers of prisoners mandated by AB109, California is now ranked as the second-largest prison system. Texas now claims that honor of being first.

AUTHOR CONTACT INFORMATION

R. N. Homer Christensen
Senior Instructional Designer and Implementation Lead
Dynamics Research Corp.
Two Tech Drive
Andover, MA 01810
916.220.0141
rchristensen@drc.com
homer@homerchristensen.com

AUTHOR BIOGRAPHY

R. N. Homer Christensen has been a published poet, professional writer, and instructional designer for over 25 years. His client list includes Microsoft, Macromedia, Aldus, Asymetrix, and other leading technology companies. His work has been recognized by the Society for Technical Communicators, including an *International Award of Distinction* and two *Awards of Excellence*. In 2003 he was a Poet in Residence at Salmon Publishing, in County Claire, Ireland.

Homer is currently the Training Development and Implementation Lead at Dynamics Research Corporation (DRC), the training and documentation contractor for the SOMS project of the California Department of Corrections and Rehabilitation. In his spare time, he farms his yarden, studies permaculture, and enjoys repurposing found objects.

DITA PROOF-OF-CONCEPT PUBLISHING SYSTEM

Matso Limtiaco, Principal Technical Writer, Intermec Technologies Corporation

This paper describes our department's DITA Publishing System, developed as part of our effort to learn and implement DITA authoring processes. Because we knew our resources would be limited, we developed the system to use our existing authoring tools in conjunction with a few inexpensive software plug-ins. We've now published more than thirty manuals in ten languages...on a DITA project budget of less than \$8000 spread over three years.

In 2010, our Tech Comm group began work on a DITA-based proof-of-concept authoring project for a new line of hardware products. We wanted to learn how to single-source our content, and to save money on translation costs. But we didn't have the resources for new tools or extensive training, and only one writer on our seven-member staff had any DITA experience.

To learn more about structured authoring, we started using Structured FrameMaker (SFM), and three writers moved into full DITA authoring for the proof-of-concept project. We spent about a year developing the system using SFM and a few inexpensive software plug-ins.

To date, we've produced more than thirty user manuals (*technically* only four manuals, but most in ten languages, all 140+ pages each) with our system. DITA has saved us a substantial amount of retranslation cost. Our DITA-sourced computer manuals now share almost 70% of their content thanks to DITA single-sourcing. So far, our expenditures for training and software total a little under \$8000, spread over three years. This paper describes what we're doing and how we did it.

ABOUT INTERMEC TECH COMM

Our company makes mobile computers, industrial bar code label printers, and associated hardware and software. Tech Comm produces all user documentation, including user manuals and guides, instructions, regulatory documents, and API references. We use mature processes and common authoring tools:

- Template-based, task-oriented writing with print, PDF, and online Help deliverables
- Department style guides for all document types
- Standard processes for creating, reviewing, editing, and releasing documents
- Translations (up to twelve languages) of Quick Start Guides, small Instructions, regulatory information
- Tools: FrameMaker, InDesign, RoboHelp, Flare

When we first started thinking about DITA authoring, we had seven people on our team: a manager, one principal writer and two senior writers, and three junior writers. One of the junior writers had DITA experience from a previous job, and was expected to serve as project engineer for any DITA implementation we adopted.

WHY DITA WAS A GOOD IDEA FOR THIS PROJECT

Even after we decided to create a DITA proof-of-concept project, we were undecided about what form the project would take. We knew we faced a steep learning curve, and the sustaining work we did for existing

products didn't seem to lend itself to trial and error. Shortly afterwards, Marketing announced a new line of bar code label printers that would use a common firmware. We were asked to translate the new printer user manuals to nine additional languages, which we hadn't done before.

These factors made the new printer manuals a good choice for our DITA proof-of-concept work. Thanks to the common firmware, a large portion of the content could be shared between the manuals. By writing standalone DITA topics, we could save money on translation costs in the long term – when we made revisions, we would only need to send out revised topics for translation instead of the entire manual.

However, because the company was in the middle of a long downturn, there was little funding for new software, third-party consultants, or training. So our DITA project goals reflected our resource concerns:

- Use our existing tools to author DITA source files and publish to PDF.
- Save money on translation costs.
- Reuse as much as content as possible.
- Keep the system simple to reduce technical complexity.

We also knew that since we weren't going to spend a lot of money on the DITA project, we wouldn't be "locked in" to using DITA if we couldn't find workable solutions.

MOVING TO STRUCTURED WRITING

We hired a consultant to introduce Tech Comm to Structured FrameMaker (SFM) in November 2009. Our goals were to learn more about structured authoring as an introduction to the more rigorous DITA environment, and to get experience using SFM as an authoring tool.

Over the next year, we developed our SFM skills to where we understood how and where to use the various elements. But we weren't in DITA topic-based authoring mode yet; we were still writing manuals based on the "chapter paradigm," writing SFM files as we had always authored them. We also found that authoring in SFM isn't the same as writing topics to the DITA standard. Among other things, our staff was still unsure of how to write certain DITA elements (such as the short description), or how much metadata we needed to include in topics, or even how best to organize the DITA topics on a server. The junior writer with DITA background hadn't been involved to that level of detail in his previous experience. And the various sources we consulted provided many suggestions, but we couldn't decide on a specific path.

OUR DITA TEAM BUILDS THE SYSTEM

I unexpectedly inherited the DITA project engineering responsibilities in December 2010. The junior writer (and prospective project engineer) decided to leave the company, and my manager asked me, as the resident "tools guy," to take on the project. I would be working closely with our principal writer, who served as the DITA project manager, and had already completed a user manual in SFM. Although my SFM experience was minimal at best (I spend most of my time in RoboHelp writing API docs), I started experimenting with our templates and EDDs so I could learn how a change to the EDD would affect the output. Then I spent some time creating dummy topics and maps to learn more about how SFM created that output.

In May 2011 we hosted a Comtech workshop that taught us a lot more about implementing a DITA-sourced publishing environment. Afterwards, we divided our responsibilities between authoring and publishing:

Project Manager:

- Understand all aspects of DITA authoring using Structured FrameMaker (SFM)
- Create Information Model – document that lists the DITA elements we expect to use in our system, and how to implement them while authoring
- Train writers in DITA topic authoring and content organization

Project Engineer:

- Understand all aspects of the publishing system using SFM and additional software plug-ins
- Maintain and revise all publishing templates and DITA “engine files” as necessary, including EDDs, language-specific templates, book-build files, and snippets
- Define all publishing processes

When the first of the prototype printers arrived that summer (six months behind schedule), I started writing topics following the Information Model, and the first manual started taking shape. We revised the Information Model along the way, and I began compiling a “DITA Notebook” that explained how to use all the components and tools. The principal writer began authoring the second printer manual in the fall of 2011, and testing procedures from the Notebook.

Peer reviews of the manuals revealed items that needed clarification, such as how to determine if certain content should be included in a concept preceding a task, or included with the task itself. We conducted all SME reviews using our already-established process (one-week shared PDF reviews), so our engineering SMEs saw no changes to the process.

We wanted to keep our DITA implementation simple to minimize technical issues, so we chose not to use cross-references or content references. We did use some metadata, mostly the product attribute, in shared topics where appropriate. Because our only output was to PDF, we didn’t plan to explore other output types. We used the .svg format for all graphics, which made text callouts easy to edit for the translation vendor, and the format can be used for future web-based output.

As soon as I completed the English version of the manual (131 total topics) in December 2011, we sent a complete set of topics to our translation vendor. We asked the vendor to work with and return the content as .dita files; moving the content to SFM files would have incurred higher desktop publishing fees.

I released the first set of manuals in March 2012. The third writer joined the DITA team that month and began authoring the last of the three printer manuals, testing both the Information Model and processes in the DITA Notebook along the way, and creating an authoring handbook with suggested best practices. By the end of the summer, we had released the remaining two manuals, and I revised a few topics in September to include some minor firmware changes and new procedures. At the end of the year, the principal writer released the first computer user manual authored with the system, which was also our first manual to include topics written by multiple authors.

“AHA” MOMENTS AND TROUBLE SPOTS

We experienced plenty of “aha” moments in this project. The big ones included:

- Learning to modify values in the book EDD. When I inherited the project, there were still some page layout and style issues with our templates, but I had no idea how to move the templates in the direction we wanted. Studying the *Adobe FrameMaker 8 Structure Application Developer Guide* helped tremendously, and by testing the EDD with dummy DITA topics I was able to figure out almost everything we wanted.

- Discovering that although there are more than 170 different DITA elements available, we really only needed to use twenty-four of those elements for our initial work.
- Realizing we could mix and match DITA-sourced and non-DITA FrameMaker files in the same book. All front matter and TOCs are produced from our non-structured FM templates.
- Learning that short descriptions are not typically seen in print output – they are generally seen in web searches. Up to that point, topic content seemed awfully repetitive, as the short description and introductory text were often very similar.
- Figuring out how to use language-specific book templates to set up the correct fonts and header text for the generated SFM files. The templates automated several postauthoring tasks.
- Being reminded the hard way that you should never use spaces in the names of XML files, or in the names of graphics you place in those files.

As we developed the DITA Publishing System, most of the technical issues were solved with a question to a forum, or to the software developer at Leximation (our plug-in creator). Most of the trouble spots we ran into had more to do with process than anything else.

- Using SFM for structured authoring is not the same as using SFM to write strict DITA topics. You have to think differently about your content when you're authoring a standalone piece of information to the DITA standard. Strict topic-based authoring isn't hard to learn...but it was *one more thing* to learn.
- After we published the Information Model, I found that some of the element requirements didn't match the elements found in our pre-DITA, prototype SFM user manual. Upon further review, I found some "cheater" workarounds in the prototype that were used for semantic markup (such as a procedure heading), but were not allowed in DITA!
- It was hard to maintain the DITA Notebook (60 pages and counting) while I was authoring and publishing the user manuals. "Maintain" means: create or revise procedures, publish to PDF, post the revised Notebook at our Sharepoint site, and notify everyone by email that I had updated the Notebook. As each of the other writers worked on her individual project, she might discover that a procedure was actually obsolete (or broken!) because I hadn't published the updates in a timely manner.
- We use tables in many places throughout the user manuals. To complete each translated manual, we spent a lot of time checking and adjusting the column widths for every table since certain languages occupy a lot more space than English.
- The language-specific book template for Traditional Chinese never corrected the fonts as expected. When we revise that book, we have to replace fonts manually... *one element at a time*.
- We ignored good advice about using numbered callouts in illustrations, and used translated text in the callouts as we had always done. For some languages, the translated text caused distortion in the graphics when we generated SFM files, and each graphic had to be manually corrected. We have since adopted a numbered callout format.
- Completing multiple translated user manuals in a short period of time sounded great at first. Sending each manual through our department standard "quality check" took more time than usual because of the sheer number of manuals—we weren't used to seeing ten books in the span of a few days!

REACHING GOALS

We proved that we could handle the work from a technical standpoint. Our DITA server is now home to 500+ topics (300+ in ten languages, plus another 200+ in English only). What about the other goals?

In 2010, we guesstimated the cost of translating a manual as \$6000 per language, based on a one-off translation requested by our safety group. The translation cost for the first complete set of DITA topics was just under \$5000 per language for roughly the same content—an estimated savings of about 20%. We did our own page layout, contributing to the lower cost.

When we revised that first set of topics in September 2012, we estimated the cost at \$600 per language, based mostly on desktop publishing fees. The actual cost for translations (three new and three revised topics) was only \$200 per language. Of course our product team was happy to spend less money than expected!

With the original set of three printer manuals, we shared about 20% of the topics between the manuals since they used a common firmware. Our in-work computer user manual shares almost 70% of its content with the previously released computer manual.

STAYING ON BUDGET

The largest amount we spent on any single part of the project was about \$4500, on the Structured FrameMaker workshop in 2009. The consultant ran the workshop and provided custom EDDs and other DITA component files. We were already using FrameMaker, so we only needed to purchase DITA-FMx (\$1500 for seven licenses, recommended by the consultant) to get started.

By hosting the Comtech workshop, we sent four attendees for the price of one, costing the department \$1500 instead of \$6000. We purchased StructureSnippets when we found it would automate some page layout tasks, bringing the total amount spent to just under \$8000 over the three years it took to complete the project.

OUR TECHNICAL SETUP

- Adobe FrameMaker 8
- Leximation plug-ins: DITA-FMx and StructureSnippets
- Adobe Acrobat 9 for shared reviews and PDF production

OUR AUTHORIZING/PUBLISHING PROCESS

Several steps in this procedure are specific to our authoring tool, Structured FrameMaker (SFM).

1. Author DITA topics and maps in SFM. Each map corresponds to the content of a single chapter in the manual.
2. Generate FrameMaker files (“build a book”) from the DITA maps.
3. Publish to PDF for reviews.
4. Revise, publish, review again.
5. Generate final FM files and perform final page edits.
6. Publish to PDF and perform department quality check.

7. Send all topics to translation vendor.
8. After translations are received, repeat steps 5-6 for each language.
9. Release the manuals.

WHERE WE'RE HEADED NEXT

Our current plan is to continue using the DITA Publishing System for all new computer and printer user manuals. We'll implement more content reusability (conrefs on the way) and more output types (an HTML prototype by the end of the year). We're also researching more platforms (style sheets for mobile phones and tablets), and possibly even new tools like a component content management system...although as before, there's still only a limited budget!

RESOURCES

Adobe Systems Incorporated. Adobe FrameMaker 8 Structure Application Developer Guide (San Jose, CA: Adobe Systems Incorporated), 2007.

Houser, Alan. What They Won't Tell You About DITA (Proceedings, Writers UA Conference for Software User Assistance, 2011).

Linton, Jennifer, and Bruski, Kylene. Introduction to DITA: A User Guide to the Darwin Information Typing Architecture (Denver, CO: Comtech Services), 2006.

Miller, Frank. Information Modeling for Topic-Based Authoring in DITA and XML (Workshop presented by Comtech Services, Everett, WA), 2011.

Self, Tony. The DITA Style Guide: Best Practices for Authors (Research Triangle Park, NC: Scriptorium Publishing Services), 2011.

Urban, Linda. Topic-Based Authoring: Getting a Handle on Topic Types (Proceedings, Writers UA Conference for Software User Assistance, 2011).

Vazquez, Julio, and SDI Global Solutions. Practical DITA (Raleigh, NC: lulu.com), 2009.

AUTHOR CONTACT INFORMATION

Matso Limtiaco
Principal Technical Writer
Intermec Technologies Corporation
6001 36th Ave West
Everett WA 98203
425.265.2383

AUTHOR BIOGRAPHY

Matso Limtiaco is a Principal Technical Writer for Intermec Technologies Corporation in Everett WA. Along with serving as the lead API reference writer, he is the DITA project engineer for Intermec Tech Comm, and published the first user manuals created with the DITA system. He created and maintains the department DITA Notebook, which includes procedures for using, maintaining, modifying, and troubleshooting the publishing system. Matso is also a published composer/arranger of music for jazz ensemble, and in addition to his "day job," maintains an active schedule as a saxophonist and jazz education consultant.

DOC SPRINTS: THE ULTIMATE IN COLLABORATIVE DOCUMENT DEVELOPMENT

Sarah Maddox

This presentation at STC Summit 2013 describes how to plan and run a successful doc sprint. The result? High-quality documentation, happy customers, and an enhanced reputation for your tech comm team.

A doc sprint is an event where people get together to write tutorials and, often, code. The sprinters work together for a given period of time, usually two to three days, on a specific set of documentation. They tap into each other's skills, get into the creative zone, have fun and write haikus. The outcome is a number of top-quality tutorials, plenty of learning, and a set of new acquaintances.

While we're delving into the how, we'll also explore the why, hear some stories, and see some results. The presentation covers these topics:

- Introduction to doc sprints, agile environments, and why a doc sprint is a good fit for technical documentation.
- Who to invite, when to start, and how to ensure that the sprint will produce the documents you need.
- How to get the best out of the sprinters.
- Collaborative tools for use during the sprint.
- Sprinting across the world: Handling multiple time zones, early sprinters, late sprinters.
- How to run a retrospective, and why.
- Reviewing and publishing the documents, and writing up the results.
- Other innovative types of sprints for documentation teams.

This proceedings paper introduces the above topic areas. It is intended as an introduction to the presentation, not as a replacement for the presentation.

During the presentation I'll also tell some stories from real doc sprints, and give tips from the experts. Those stories and tips aren't in this overview either.

AGILE METHODOLOGY AND DOC SPRINTS

Many technical writers find themselves in agile environments, where time is at a premium. It can be difficult to pry the developers and other subject matter experts away from their day-to-day deadlines. This session shows how we can walk the agile walk, talk the agile talk, and at the same time have fun developing the tutorials and documents our customers need. Time-boxing, efficiency and structure are key to agile development. We use those virtues to plan and run a doc sprint.

In a doc sprint, we use the following key aspects of agile methodology:

Timeboxing:

- Work for a set period.
- Begin with a goal.

- Re-assess the goal continually.
- Redefine the goal if necessary, to ensure we achieve something useful within the scope of the sprint.

Self-organising teams:

- The team is central. People collaborate. They examine their work, and continually improve the way they work together.
- We hold a planning meeting at the beginning of the sprint, and report-back sessions throughout.
- We hold a retrospective at the end, and use the resulting feedback to improve the next sprint.

Efficiency:

- Preparation.
- Planning.
- Helping people do their best work during the sprint.

Structure:

- Templates.
- Guides.
- Making sure people know the part they will play.

PLANNING A DOC SPRINT

A doc sprint takes a lot of planning and hard work up front, then review and tidying up afterwards. But it's worth it, because the result is high quality documentation written by experts in the field. Good preparation means that the sprinters get off to a flying start when the day of the sprint dawns.

WHEN TO START

Good advice is to start planning early, two or three months ahead, especially when organizing your first doc sprint. If sprinters will need to travel long distances, or if the sprint is longer than a day, they will need plenty of time to make arrangements.

WHO TO INVITE

Everyone! Don't be shy. To get your numbers up, you'll need to invite a large number of people. We'll look at the groups of people to invite, how to word the invitations, and the channels to use.

GETTING THE DOCUMENTS YOU NEED

How can you make sure the sprint will produce the documents you need?

Define a focus for the sprint. This is something short and simple. For example:

- Developer tutorials for product X.
- Quick-start guides for product Y.
- API documentation for version N of product Z.

Create a wish list of the tutorials or other documents that you would like to develop.

Create a template for each type of document required.

Consider writing a simple style guide.

Write detailed and careful technical guidelines on how to use the documentation platform.

If the sprinters will write code as well as documentation, they will need technical guidelines for the coding part too.

MORE PLANNING

There's plenty more to do, ranging from practical tasks like preparing the venue, to technical tasks like setting up the development environment.

SPRINTING

In some cases, you'll be sprinting around the world. The sprinters may be in different locations, different time zones, and are sometimes even not be native English speakers. To manage this, you need:

- Flexibility, to cater for sprinters who want to start a couple of days early or late, and other oddities.
- A schedule, so people know where they need to be and when they can find each other online.
- Mentoring and teaching sessions, to help new sprinters and give something back to people who are giving you their time.
- Technology, to hold it all together: a collaborative writing platform, a chat group, video conferencing and/or webinar technology, and a code repository.

OTHER USEFUL TYPES OF SPRINTS

This part of the presentation is about extending the metaphor and knowing when you're on to a good thing. Technical communication teams can employ some neat agile tricks to get what they need from their stakeholders and subject matter experts.

We'll take a quick look at these new types of sprints:

- Innovation sprints, giving busy technical writers some time to think about and implement innovative techniques or procedures. An innovation sprint is also known as a "tech writer funny hat day". See the presentation to discover why!
- Doc blitz tests, an opportunity for quality assurance engineers and developers to work with the technical writers on the documentation for an upcoming release. We spend an hour doing concerted and concentrated testing of specific aspects of the documentation.
- Doc blitzes, a way of getting other teams' help to fix a specific and large-scale problem in the documentation. Let's say the development team makes one of those small, dreaded changes to the product that necessitates sweeping changes to the documentation. Almost all pages and screenshots are affected. A small technical writing team will find it hard to apply the updates in the time allowed. On the other hand, the changes are fairly straightforward and follow an easily defined pattern. Just the job for an engineer.

RESOURCES

Why do people write free documentation? by Andy Oram: <http://onlamp.com/pub/a/onlamp/2007/06/14/why-do-people-write-free-documentation-results-of-a-survey.html>

Experiencing the Atlassian Doc Sprint - Aug 2012, by Swapnil Ogale: <http://icreatedocs.blogspot.com.au/2012/09/experiencing-atlassian-doc-sprint-aug.html>

We Did It: Zero to Book in Five Days, by Anne Gentle: <https://www.openstack.org/blog/2013/03/we-did-it-zero-to-book-in-five-days/>

Atlassian doc sprint wiki: <https://confluence.atlassian.com/display/DOCSPRINT/>

How to plan a doc sprint, by Sarah Maddox: <http://feathers.wordpress.com/2012/09/07/how-to-plan-a-doc-sprint/>

A big hairy task list for planning a doc sprint, by Sarah Maddox: <http://feathers.wordpress.com/2012/09/07/a-big-hairy-task-list-for-planning-a-doc-sprint/>

HipChat: <https://www.hipchat.com/>

Tea Party: <http://tea-party.appspot.com/>

Google Talk: <http://www.google.com/talk/>

GoToWebinar: <http://www.gotomeeting.com/fec/webinar>

Booki: <http://flossmanuals.net/booki-user-guide/>

Confluence: <http://www.atlassian.com/software/confluence>

Bitbucket: <https://bitbucket.org/>

GitHub: <https://github.com/>

Doc Sprint Mountain View, by Peter Lubbers: <http://blog.webplatform.org/2012/12/doc-sprint-mountain-view/>

Recap of Web Platform Doc Sprint Berlin, by Andre Jay Meissner: http://www.google.com/url?q=http%3A%2F%2Flists.w3.org%2FArchives%2FPublic%2Fpublic-webplatform%2F2013Feb%2F0120.html&sa=D&sntz=1&usq=AFQjCNGDrM5Zap7v4gWWyT1Xfu_lu5c_dg

Innovation sprint, by Sarah Maddox: <http://feathers.wordpress.com/2010/07/03/technical-writers-hold-an-innovation-sprint/>

Documentation blitz-test, by Sarah Maddox: <http://feathers.wordpress.com/2010/12/04/a-blitz-test-of-the-documentation/>

Documentation blitz, by Edwin Dawson: http://blogs.atlassian.com/2011/09/interviews_from_the_jira_screenshot_doc_blitz/

AUTHOR CONTACT INFORMATION

Sarah Maddox
Technical writer
Sydney, Australia
Twitter: @SarahMaddox
Blog: <http://feathers.wordpress.com>

AUTHOR BIOGRAPHY

Sarah Maddox is a technical communicator, author and blogger. Her focus is on technical documentation that gives the readers exactly what they need. Sarah started her working life as a software developer, but recognized after some years that what she really loves is writing. Fourteen years ago she moved into technical writing, with a lot of blogging and a bit of fiction on the side. Sarah has worked for various companies large and small, in South Africa, UK, Netherlands, and Australia. She is currently working at Atlassian in Sydney, where she has learned all about developing technical documentation on a wiki.

EMBEDDING USER ASSISTANCE WITH DITA

Ray Gallon

The complexity of modern software demands decision support as well as procedural assistance. Decision support principles also correlate well with JoAnn Hackos' recent restatement of the principles of Minimalism, as developed by John Carroll. To help people learn better and faster, we should not only be embedding user assistance into our interfaces, but embedding decision support and other conceptual information into tasks and procedures - an idea we call "Double Embeddedness." The idea is to deliver just enough information, at just the right time, in just the right place, to help users do real, productive work. And we call that Kanban information, after the system developed by Toyota and since adopted by many companies around the world. Many people worry about using DITA because of its seemingly rigid separation of information types: tasks, concepts, reference. However, we demonstrate here that it is possible to integrate conceptual information in DITA tasks and steps, without breaking its semantic structure. Illustrations from a real project are used to show bits of DITA source and the published results.

BEFORE PROCEEDING, DECIDE

The complexity, depth of features, and overall power of modern software can often leave our users perplexed. This frequently occurs at just the moment when they have some immediate, contingent need – “I need to get this done, and NOW!” User assistance that is limited to procedures cannot help people with contingent needs decide which, if any, procedures they need to use. Long-winded conceptual topics might provide insight into this question, but people with contingent needs are not going to spend the time to wade through long texts.

We can help users get real work done more quickly with a bit of decision support. In software, when we use that term, we can mean two things:

- Automated decision support aids, such as expert systems
- Information designed to inform the user's judgment, but not formalized into an automated system

In technical communication, we are most often concerned with the second of these. The National Research Council (NRC – United States) has defined the following core principles that apply to decision support of any sort¹:

- Begin with users' needs
- Give priority to process over products
- Link information between producers and users
- Build connections across disciplines and organizations
- Seek institutional stability
- Design processes for learning

A user who needs to get something done quickly, when faced with complex applications, needs to know what a task or procedure does, and why s/he would be interested in doing it. This information needs to be communicated concisely, in a way that helps the user retain this information the next time s/he needs to do this same thing. It must also lead the user directly to the procedural assistance, if, and only if, s/he desires.

COMING TO GRIPS WITH MINIMALISM

Interestingly, the NRC core concepts correlate nicely with the core principles of Minimalism, as restated in a recent blog post by JoAnn Hackos²:

- Focus on an action-oriented approach
(Users' needs, process)
- Ensure you understand the users' world
(Users' needs, links and connections, institutional stability)
- Recognize the importance of troubleshooting information
(Users' needs, links and connections, learning)
- Ensure that users can find the information they need
(Users' needs, learning, links and connections)

Note that users' needs correlates with all of these core principles.

I have recently been advocating that we treat users as learners, a notion validated by John Carroll, the father of Minimalism, in a recent interview with Nicky Bleiel where he states that he prefers the term "learner" to both "user" and "customer."³ Carroll has also suggests that most people learn best by doing something rather than reading about it – that, in fact, his research, and others', states that the kinesthetic memory function is among the most efficient for embedding new procedures and concepts in our brains.

Minimalism, in fact, does not suggest that concepts are excess baggage, it merely suggests that the best place to learn a concept is while performing a task, and that permits user/learners to perform meaningful, realistic work immediately.

User assistance that is limited to procedures that one memorizes by rote is not, in fact, minimal if we need to constantly come back to the UA to get the step-by-step instructions for a number of different procedures based around the same principles.

It is not only minimal, it is also meaningful if, embedded in our procedural information, we have just enough conceptual information to be able to generalize what we learn in the current task to other tasks we may not yet have tried.

It is also minimal and meaningful if one quick look at the UA helps us understand that we don't need to bother with the screen or information object in front of us – or, indeed, that we do.

The solution, then, is *Double Embeddedness*: embed the user assistance in the interface, to make it available in the time and place that it is most needed, and embed concepts into tasks, to help learners acquire the understanding that will allow them to perform new tasks without having to consult the UA another time.

KANBAN INFORMATION

The Japanese term, *Kanban*, is often assumed, in English, to mean "Just in Time" – as in "just in time delivery" of spare parts, lowering warehouse costs. In fact, the original Japanese term refers to a complete set of principles, developed by Toyota to render manufacturing more efficient⁴. If we apply Kanban ideas to information and UA, we come up with these core principles:

- We give the learner all the information s/he needs and only the information s/he needs.

- We want to deliver that information when s/he needs it – which implies, at the moment s/he has real work to do.
- The logical conclusion is that user assistance needs to be embedded in the software itself, in such a way that:
 - The user can find it immediately, without excessive searching, if s/he needs it.
 - If s/he doesn't need it, it stays out of the way.
- We want the learner to understand the information in a way that s/he can apply it to other situations, without needing to call repeatedly on the user assistance.

DOUBLE EMBEDDEDNESS USING PROGRESSIVE INFORMATION DISCLOSURE

- In user experience design and information architecture, the term *Progressive Disclosure* is used to describe a set of design principles based on the idea that people understand a system better when you help them prioritize features and spend more time on the most important ones.
- Progressive disclosure specifies:
 1. Initially, show users **only a few** of the most important options.
 2. Offer a **larger set** of specialized options upon request. Disclose these secondary features only if a user asks for them, meaning that most users can proceed with their tasks without worrying about this added complexity.

In applying these ideas to user assistance, Andrea Ames speaks of *Progressive Information Development*⁵.

DITA AND DOUBLE EMBEDDEDNESS

At first glance, one might think that DITA's separation of topic types into concepts, tasks, and reference was incompatible with the idea of double embeddedness. This turns out to be incorrect. The DITA structures provide many opportunities to embed concepts within tasks.

What follows are examples from a real project that I have been working on, used with permission of my client, ValidIT, a small French IT company based in Orléans. The application does Business Activity Monitoring (BAM), programmed in Java. It is based on a client-server architecture, and the Java desktop client uses a progressive disclosure interaction model. A thin, browser-based client is planned that will duplicate the desktop interaction model as closely as possible.

We decided to create an embedded user assistance system that followed the same model, using principles of progressive information disclosure. Although the volume of information is relatively small, the nature of the software offers many opportunities for reuse. We decided to use out-of-the-box DITA for our structure, with oXygen as our editing tool. We are publishing to JavaHelp – not my first preference, but the choice of the development team.

At this time, there is no component CMS in this system. As mentioned above, the volume of information is relatively small, and the software is not yet in release – only two languages are planned for the first release, English and French. Plans call for eventual releases in several European and probably two Asian languages, and provision for this is already built into the interface components that will need localization, including the user assistance.

The current DITA repository is a dedicated area of the company's software version control system, SVN. It is anticipated that this will need to be migrated to a CMS in the future, so the design of the repository is similar to one that we would use in a CMS. Connection with the editor is provided by oXygen's SVN plugin, which provides easy checkin and checkout facilities.

THE CONUNDRUM

BAM installations are, by their nature, highly customized for each client. The minimalist dictum to document real user tasks, and not the interface, becomes almost impossible in this case.

Our first user assistance release is, therefore, more oriented to an IT manager than to the final users of the software, as this was the only way we could produce generic user assistance at this time. We retain a user task orientation, all the same.

One feature of this software is to partially industrialize the BAM installation and configuration process, lowering commissioning costs to potential clients. A future release of the user assistance will feature conditional builds, where the conditional material will be oriented to one of the main target industries for this software. We also plan to add scripts that will permit the user assistance to plug company-specific terminology (in the application database) into variables in the DITA source.

These possibilities for automation also figure in the decision to use DITA for this project.

PROGRESSIVE INFORMATION DISCLOSURE STRUCTURE

After an audit of screens and a task analysis, we determined that we needed the following structure:

1. All elements of the interface that trigger an action or reveal information have tool tips. Most of these are standard Java tool tips, but the most significant interface elements have enhanced tool tips.

All tool tips appear only after a delay, to stay out of the learner's way if not needed. In the case of enhanced tool tips, a small question mark is added to the cursor on hover, so the learner knows that an enhanced tool tip is available if needed.
2. Enhanced tool tips, level 1: on hover, after delay, pops up and indicates what the element is or does.
3. Enhanced tool tips, level 2: clicking anywhere on the body of the level 1 enhanced tool tip reveals a slideout that gives decision support information – what the learner can use this element for. It also contains a link to a task or concept related to the element.
4. If the interface element has only one task associated with it, clicking the related link opens a popup window with full procedural assistance for that task. If there are multiple tasks associated with the element, the link goes to a conceptual topic.
5. A floating window with complete task or conceptual information, and links to related tasks and concepts.
6. Once a floating window is open, it is possible to click an icon that opens full, classic tripane help, with full table of contents and full text search.

Full Disclosure: This structure does not conform totally to the double embeddedness model I have proposed in this text, though many of the task topics include embedded concepts. This is a work in progress, and I would probably do some of this differently were I starting today.

HOW WE BUILT IT

Here are some examples of elements from the user assistance, with DITA source to show how it was done. Needless to say, the creation of such a system also requires adjustments in the CSS and, at times, the XSL transforms. These were done by the development team as we went forward. Some of them had more to do with the deficiencies of JavaHelp than any inherent lack in the DITA Open Toolkit.

ENHANCED TOOL TIPS

We used the DITA generic topic tag – <topic> – to generate two levels of tool tip as follows:

- Level 1 is taken from the <shortdesc> element and displays information about the element: what is it, what it does. The displayed result is shown in Figure 1.

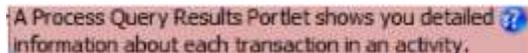


Figure 1: Level 1 Enhanced Tool Tip from <shortdesc> element only

- Level 2 is the entire topic, including <shortdesc> <body> and <related-links>. It reveals a slideout that adds decision support (when to use it) and related links. The displayed result is shown in Figure 2.

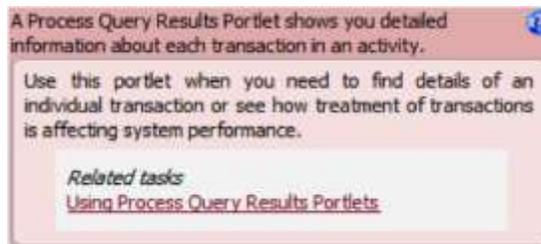


Figure 2: Level 2 Enhanced Tool Tip showing slideout - all elements in the topic

Figure 3 shows the DITA source that produced these. Note that both levels of the tool tip are generated, via the CSS, from one single topic.

<topic> <title> **Process Query Results Portlet** </title>
<shortdesc> Short Description: A Process Query Results </shortdesc>
<glossary/glossentry_portlet/abbreviated-form> </abbreviated-form> shows you
detailed information about each transaction in an activity. </shortdesc>

<prolog> <author> Author: Ray Gallon </author>
<resourceid> Resource Id:
id=dashboardExplorer.portalViewer.ProcessQueryResultsPortlet.caption
appname=javahelp </resourceid> </prolog>

<body> <p> Use this portlet when you need to find details of an individual
transaction or see how treatment of transactions is affecting system performance
</p> </body>

<related-links> Related Links:
<link> [L/main/BPResultsQueryTask.dita](#)
Using Process Query Results Portlets </link>
</related-links> </topic>

<shortdesc> element contains an
<abbreviated-form> glossary entry that is
rendered without popup definition here. It
is rendered differently in the task topic.
This is managed from the CSS

We used the <resourceid> element to
link the tool tip to the interface element.
The ID's were managed by the
development team with tech comm input.

Figure 3: DITA source for the Enhanced Tool Tip topic

Related links are hard coded in the tool tips, while elsewhere we use <reltable>..

TASK TOPIC

The tool tip in the example above links to a floating task panel:

The screenshot shows a task panel titled "Using Process Query Results Portlets" from the "VITAM Documentation". The panel content includes a short description, a definition of a portlet, and instructions on how to use the portlet. A callout points to the title, stating "Click to open tripane help with table of contents and full text search". Another callout points to the short description, stating "The <shortdesc> and first <p> are reused from the tool tip". A third callout points to the definition of a portlet, stating "This time the <abbreviated-form> is resolved on hover to pop up a definition from the glossary using <keyref>".

The DITA source for the top of the task panel is shown below the screenshot. It includes the title, short description, author information, and the context section.

```
<task title="Using Process Query Results Portlets">
  <shortdesc #reference="...">
  <shortdesc>Short Description: A Process Query Results [glossary/glossentry_portlet]
  <abbreviated-form #keyref="..."> shows you detailed information about each transaction in an
  activity.
  <prolog author="Ray Gallon">
  <taskbody context="Context">
    <p>Use this portlet when you need to find details of an individual transaction or see how
    treatment of transactions is affecting system performance.
    <p>Results are displayed in a spreadsheet-like table. You can arrange data display in this portlet
    as follows:
    <ul>
      <li>Select the columns you want to display
      <li>Sort data by column
      <li>Rearrange horizontal column order
```

Figures 4 & 5: Top of a Task Panel and DITA source for the top of the Task Panel

Although the first elements of the task panel are reused from the tool tip, display can be, and is, in this case, handled differently, as you can see in Figure 4, where the word “Portlet” is a <keyref> to the glossary, and pops up a quick definition. Clicking the highlighted word links the learner to the full entry in the glossary.

BLENDING CONCEPTS AND TASKS

All too often, and I am guilty of this, we use a concept as a way to provide general information that links to multiple tasks. Effectively, the concept serves as a repository for information that might better be embedded in tasks, but it also serves as a link page to multiple tasks when we cannot know what task the learner needs to perform.



Figure 6: This structure avoids embedding a long list of task links in the tool tip – but is it helpful for learning the application?

This is not double embeddedness, nor is it good Kanban. But how can we embed concepts in task information, given DITA's information typing structure?

The answer is that DITA tasks are not as rigid as many people think. Inside the <task> element, you can include a variety of elements that maintain the semantic structure of the topic, and allow you to embed conceptual material either at the top of the task, or inside an individual step. All this without specialisation, as diagrammed in Figure 7.

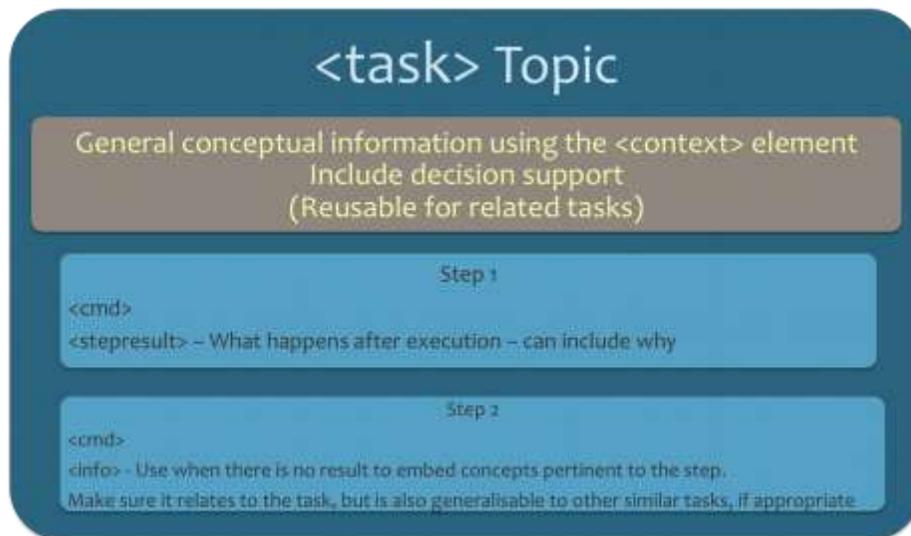


Figure 7: This diagram shows how the DITA <task> topic allows us to encapsulate local, conceptual information without breaking the semantic structure of the DITA tags.

You can manage this to a very fine granularity, if you need to. For example, it is possible to use <stepresult> uniquely for a substep, inside a <substep> element, and then add another at the <step> level that applies more globally (i.e., to all the <substep> elements embedded in it).

In the following example (Figure 8 and Figure 9), we use the <context> tag to include general conceptual information about the task. If there are other tasks that need the same information, we save it as reusable content and call it into each task that needs it.

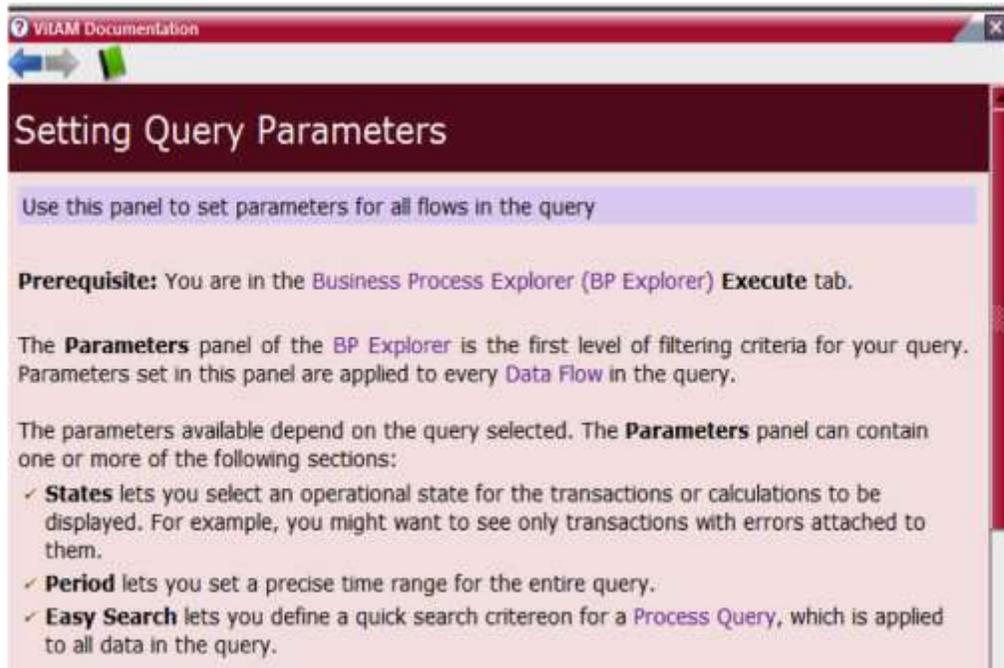


Figure 8: Task Panel with conceptual information embedded in a <context> tag (shaded area at bottom)

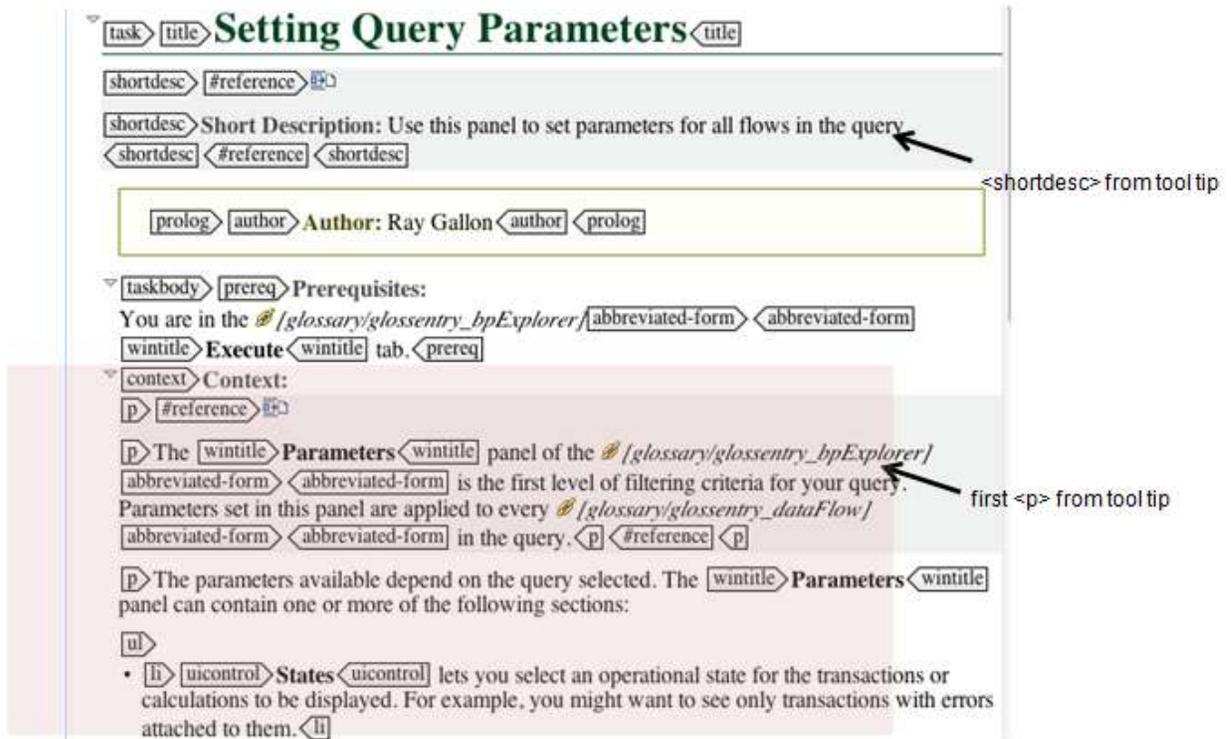


Figure 9: DITA source for the Topic Panel in Figure 8. The <context> element is in the shaded area.

CHOICE LISTS

In addition to <stepresult> and <info> the DITA <step> element allows you to nest auxiliary elements that can also be useful to communicate conceptual information. One of them is the <choices> element.

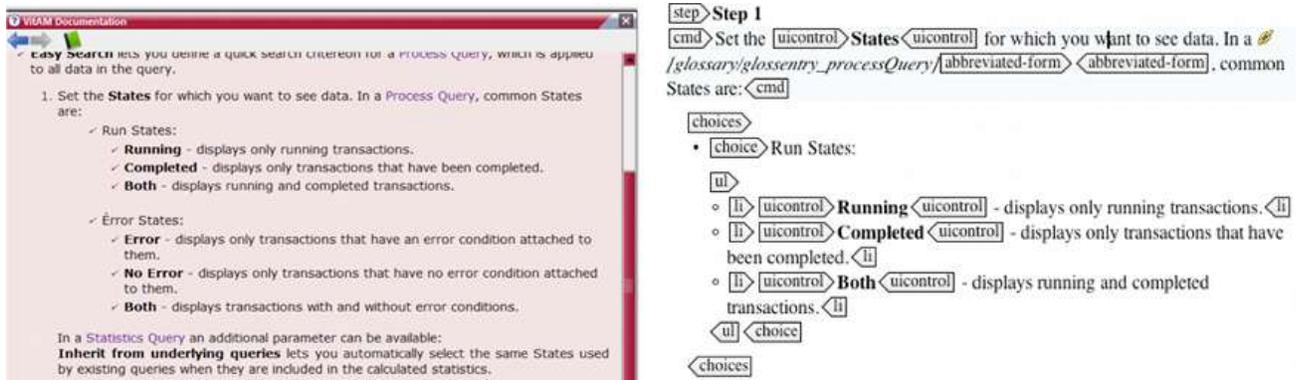


Figure 10: The <choices> element lets you list options in a given situation and explain them (shaded areas)

In this example, you see a simple explanation of what happens when you select the choice. For more complex concepts, though, you could add more conceptual information, for example, why you might want to use one choice or the other.

You can nest additional tables inside a <choices> element. This can be seen in Figure 10.

CHOICE TABLES

The <choicetable> element is very similar to the <choices> element, except it formats your data in a two-column table with "option" name in the first column, and "description" in the second. You can use the "description" column for conceptual data. You cannot add more columns to a <choicetable>.

Figure 11 shows how we used <choicetable> in the same Task Panel as Figure 10.

2. Set the **Period** for the query using the following parameters:

Option	Description
Drop down list	Selects where the date and time criteria are applied: <ul style="list-style-type: none"> ✓ Select Process Start to filter the query by the date and time the transaction was initiated. ✓ Select Process End to filter the query by the date and time the transaction completed.
from and to parameters	Check the from box to set the starting date of the period you want to set. Check the to box to set the end of the period. If you want the start or the end of the period to be open-ended, leave that parameter unchecked. For both parameters: <ul style="list-style-type: none"> ✓ The arrow in the date box opens a calendar from which you can pick the date. ✓ The time setting can be controlled by clicking the up or down arrows. You can also enter date or time directly from the keyboard.

Figure 11: The <choicetable> may be limited to two columns, but you can style its display as you like. In this example, column dividers and table borders are not rendered.

MULTIPLE POSSIBILITIES

There are, in fact, 11 elements that you can inside a <step> after a <cmd> element. You can add conceptual information related to an individual step in most of these elements without violating their semantic purpose. Be sure, however, that you don't overwhelm your learners. In the Kanban spirit, give just enough information to help the learner do what s/he needs to do.

As a final note, there is a "composite" topic, that lets you nest <concept> and <task> elements in the same topic. It is not a map. It is to be used with caution - I've used it exactly once in my entire career, and I still am not sure it was the right thing to do.

Fortunately, DITA is flexible and versatile enough that we can craft it to meet our needs, and help us help our learners, without resorting to this sort of bludgeon approach.

In the live presentation, I will be showing two alternative implementations of similar systems. These alternatives were not yet ready at the time of writing this document.

REFERENCES

- [1] Silverman, Howard. "Principles for Effective Decision Support | National Research Council." *People & Place*, Volume 1 issue 03 (14 March 2013).
http://www.peopleandplace.net/on_the_wire/2011/3/21/principles_for_effective_decision_support_national_research_council.
- [2] Hackos, JoAnn. "Minimalism Updated 2012." *CIDM* (14 March 2013).
<http://www.infomanagementcenter.com/Resources/eNews2012-12JHackos.pdf>.
- [3] Bleiel, Nicky. "Minimalism Revisited: An Interview with John Carroll." *Intercom*, February 2013 (14 March 2013). <http://intercom.stc.org/2013/03/minimalism-revisited-an-interview-with-john-carroll/>
- [4] Collective anonymous work. "Kanban." *Wikipedia* (14 March 2013). <http://en.wikipedia.org/wiki/Kanban>
- [5] Ames, Andrea L. "Enabling Progressive Information Disclosure With The "Stages Of Use" Model." (*Proceedings, STC Summit Conference, 2012*).

AUTHOR CONTACT INFORMATION

Ray Gallon
Owner/Consultant
Culturecom
3, impasse de la Gare
34260 St. Etienne Estréchoux, FRANCE
+33.6.72.40.37.60
c/ Bertran 117
08023 Barcelona, SPAIN
+34.620.222.931
infodesign@culturecom.net

AUTHOR BIOGRAPHY

Ray Gallon is owner of Culturecom, a consultancy specialised in technical information design, content strategy, and usability. He has over 20 years' experience in the technical content industries, including major companies such as IBM, Alcatel, and General Electric Health Care. Previously, Ray was an award-winning radio producer and journalist, working with broadcasters such as CBC (Canada), NPR (United States), France Culture, Radio Netherlands International, Deutsche Welle, WDR (Cologne, Germany). In the late 80's, Ray was program manager of WNYC-FM, New York Public Radio.

Ray is a member of the international board of directors of the Society for Technical Communication (STC) and past president of STC France. He is a two-time winner of awards in the trans-European technical communication competition, including Best in Show. He is a frequent speaker on communications topics at conferences and seminars around the world, and has taught communications subjects at New York University, The New School (New York City), Université de Toulouse Le Mirail (France), Université Paul Valéry (Montpellier, France) and Université de Paris Diderot.

Ray shares his life between the city of Barcelona, Spain, and the Languedoc region in France. He is currently a researcher at The Transformation Society (Barcelona), which he co-founded with Dr. Neus Lorenzo.

EPUB AND TECHCOMM — ARE WE READY?

Scott Prentice

You've probably been hearing more and more about EPUBs and ebooks, and how this new technology is changing the way people consume written information. Can the techcomm industry make efficient use of this technology? What type of content is best suited? Does it really make your documentation available on "every" device and platform? Is it really ready, and if so how do you get started?

This paper will bring you up to speed on the current state of EPUB tools and technologies and how they relate to the techcomm industry. It will discuss why, how, and *if* you should consider delivering your content as an EPUB and will explain how to make the most of this new content delivery option.

UBIQUITY

EPUB is the most widely accepted ebook format by ereader applications and devices. Because ereader applications are available for almost every platform and device (from mobile, to tablet, to laptop, to desktop), documentation published to this format can be read by all users of these devices. One documentation deliverable available to users of *all* devices. That's a big claim. Technically, it's true, but as you might expect, there's a catch (maybe more than one).

Let's back up, to be sure we're all on the same page. What exactly is an EPUB? Fundamentally an EPUB is a collection of XHTML, XML, CSS, and media files wrapped up in a "zip" archive (conceptually like a CHM file that you're probably familiar with). It's a nice neat little package of content and formatting that can be easily provided to users who can add your ebooks to their online libraries for reading whenever needed. As mentioned, an EPUB does require a reader application or dedicated reader device, and this is where things start to break down. Since these reader applications and devices are created by different companies with different requirements, the rendering of an EPUB will differ on each.

The EPUB specification is maintained by the IDPF (International Digital Publishing Forum). EPUB became an official standard in 2007, superseding the older Open eBook standard (from 1999). Since then, EPUB 2.0.1 was approved in 2010, and the latest specification, EPUB 3, was approved in October 2011. A "fixed layout" format was added in May 2012, and currently under development are Indexes and Dictionaries extensions to the specification.

CREATING AN EPUB

Because an EPUB is just a zipped collection of common file formats, it is possible to create one "by hand." While some people do this, it's not likely to be a viable option for most in the techcomm world. Most of the EPUB development tools, like Adobe InDesign, have been available for some time, but are designed for people who are authoring content specifically for EPUB. This is also not likely to be a good choice for technical publications departments, because you probably already have the content in another format, and you're probably interested in delivering your documentation in multiple formats, not just EPUB.

Conveniently, the traditional techcomm authoring tools have finally stepped up to the plate and many are offering EPUB as an output option. As with the ereaders, these development tools don't all support the same level of the EPUB specification and don't all create an EPUB using the best practices. In most cases, you'll

end up needing to “crack open” the generated EPUB and make some minor adjustments. This is unfortunate, but at this stage in the EPUB tool support, that's what you're faced with.

These EPUB creation tools will typically let you import a set of FrameMaker and/or Word documents then export to the EPUB format. The resulting EPUB will open in your ereader of choice, but the actual readability/usability will vary. There are things that you can do to improve the chances that the resulting EPUB will be something close to what you would feel comfortable making available to your customers.

FORMATTING ISSUES

Remember that each ereader application or device will render the same EPUB differently, so the first thing to do is to simplify the formatting. Less formatting and a simpler layout will result in a much more readable EPUB, and one that looks better on a wider variety of ereaders. Don't bother with special fonts, and keep the inline formatting limited to bold, italic, and monospaced.

Typically, when you import a document into one of these tools, it tries to retain the formatting from the source document. This isn't usually desirable for EPUB output. So, unless the tool provides a way to clean up all of the extra bloated code, your best bet is to clean it up before importing. If you're using named styles (you are, aren't you), apply a new template that simplifies the formatting as much as possible.

The most challenging “screen” is the mobile phone. With the limited screen space you don't want side-heads or special indentation. Remove all of those features from your document before importing. Trim out all but the most essential formatting, and you'll be happier with the result. As will the readers of your documentation.

Some tools will map your named styles via CSS; this is a good thing. Other tools will hard-code style properties to each tag; this is bad. If you can tweak the formatting via CSS you'll be much better off in the long run.

Keep in mind that one of the important aspects to ebooks is the ability to personalize the ereader interface. Most ereaders let the user specify the font, font size, font color, as well as the background color. If your EPUB specifies special fonts and formatting, it may not look good in the user's custom UI.

PAGE BREAKS

One area that can cause great consternation for the page-oriented technical writer is the frequent occurrence of odd page break situations you'll see in an EPUB. There are properties that can be assigned to styles in a CSS that are intended to control widows and orphans and to force a page break, but this doesn't work consistently across all ereaders. The only way to ensure that a page break happens is to break the content into separate content files within the EPUB. Each content file in an EPUB will always start at the top of a new page.

Some tools will let you define where this happens, others won't. Learn how to do this in your tool, or if necessary, go in and make this “fix” after the EPUB has been created.

As ereader support for page-oriented CSS properties improves, this problem will become less of an issue. But you'll still need some way to specify this property in your tools.

WHAT ABOUT TABLES?

Tables cause the most problems for EPUB conversions. Again, it's the small screen limitation where you run into trouble. Content in tables of more than 3 columns will often be lost entirely. If the ereader does squish it all into the available space, the columns will be so narrow as to being completely unusable. If your documentation makes heavy use of large, complex tables, that may be a deal-breaker for the EPUB option.

Or, you may want to design for a minimum screen size that does allow the content to render in a useful manner.

People have suggested converting tables to images. This is almost as bad as having no table at all. When the table is rendered as an image, you can no longer search for the content in that table. Yes, you can typically zoom and pan around in the “image-table,” but it’s really better to recast the table into smaller tables or as a list. Figure out a way to make it work.

METADATA

One area that’s often lacking is tool support for EPUB metadata. The EPUB specification allows for varied and often complex metadata assignments, and typically a tool will restrict the options considerably. This is one of the areas that you’re likely to need to resort to opening and fixing the EPUB after it’s been exported.

Depending on how you plan to publish the EPUB, you may be required to include certain very specific metadata for use by an EPUB aggregator service or other delivery mechanism. Even if you’re not required to add metadata to your EPUB, it’s a good practice to use thoughtfully, so your book can be easily located in a large library of ebooks.

EPUB 2 OR EPUB 3?

Most tools these days will include “EPUB 3 support” in the sales and marketing information. However, few (if any) really provide even minimal support for EPUB features that were added in the EPUB 3 specification. All they do is provide the new “nav” document, make sure their HTML is valid XHTML 5, and set the version number in the metadata to indicate that it’s an EPUB 3 file. So, in reality, there’s probably very little difference between the EPUB 2 or EPUB 3 files produced by that tool.

EPUB 3 brings a host of new features to the EPUB developer, but at this time, they will all need to be manually added or created through custom development. A big focus of EPUB 3 is interactivity; where EPUB 2 discouraged the use of scripting, EPUB 3 encourages it. The content documents are all XHTML 5, which provides support for audio and video, as well as adds support for SVG and MathML. It includes support for CSS 2.1 and CSS 3, and much improved font support (especially for Asian languages). Also, the Media Overlays feature allows you to integrate audio with text to provide “read aloud” ebooks. This is just a small sampling of the features offered by EPUB 3. Someday, these features will be supported by the tools.

Additionally, extensions are being added to EPUB 3. The “fixed layout” feature was added in 2012 which is more targeted at developers of highly designed “picture” books, as well as comic books and children’s books. Two upcoming features to be added soon are the Indexes and Dictionaries extensions. These are likely to be of possible interest to the technical documentation producer.

Note that if you do manage to add some great new EPUB 3 features to your EPUB, it’s going to limit the ereaders that can make effective use of your ebook, since most don’t fully support EPUB 3 yet. Tradeoffs, always tradeoffs.

EPUB AND KINDLE?

Amazon’s Kindle devices are the primary ereaders that do not support the EPUB format. Amazon provides a utility called *kindlegen*, that will generate a “kindle” (MOBI/KF8) format file from an EPUB. While there are other utilities that can perform this conversion as well, it’s best to use the Amazon-provided tool (it’s free, why not).

If your EPUB has been developed with the “keep it simple” approach, then the resulting Kindle file is likely to come through the conversion without trouble. If you use special formatting or other possibly EPUB-specific

features it may not work well in Kindle format. If this is the case you may need to have separate workflow paths for the EPUB and Kindle targets.

TECHCOMM EPUB TOOLS

The following tools are likely to be useful to the technical publications department for EPUB creation and maintenance. There are many other tools and utilities available which may provide useful functionality, but many are designed for the EPUB developer who focuses on single purposed, hand-crafted EPUBs.

Authoring/conversion tools:

- Adobe Technical Communication Suite (FrameMaker + RoboHelp)
- ComponentOne Doc-To-Help
- EC Software Help & Manual
- ElmSoft EpubFm (FrameMaker plugin)
- Madcap Flare
- WebWorks ePublisher

Conversion tools:

- DITA Open Toolkit + DITA for Publishers plugin
- DocBook to EPUB XSLT

Maintenance and cleanup tools:

- Oxygen XML editor
- Sigil
- BlueGriffon EPUB Edition
- Calibre
- epubcheck
- kindlegen

The “authoring/conversion tools” listed are familiar Help conversion tools that now support EPUB output. These will let you generate EPUB output from your current content. Making that EPUB output into something that you're proud of may take some effort, but these tools provide a good place to start.

The “conversion tools” listed are for those groups that are using a DITA or DocBook workflow. The DITA-OT combined with the DITA4Publishers plugin produces some of the cleanest and most user-ready EPUB files of any automated workflow. This is a great option for those using DITA.

The maintenance and cleanup tools are useful for all EPUB workflows. The tools from Oxygen, Sigil, and BlueGriffon are important for opening up a generated EPUB to perform cleanup. Oxygen is a fantastic tool for all sorts of EPUB fixes; highly recommended if you can afford it.

Calibre is the “swiss army knife” of EPUB tools. It does it all (plus a few other things). It's worth installing and learning what it can do.

All EPUB workflows should wrap up with a pass through the epubcheck utility to confirm that it is valid. Oxygen and other editors do include validation as a feature as well.

If you plan to provide both EPUB and Kindle, you'll need the kindlegen utility. Amazon also provides a Kindle preview tool as well as Kindle reader desktop versions.

CONCLUSION

The jury is still out as to the usefulness of EPUB for technical documentation. In general, ebooks are a huge hit. The volume of ebook sales increases every year, and it doesn't appear to be slowing any time soon. But does that mean your customers will find benefit in reading your documentation as an ebook?

It can be argued that certain types of documentation are better suited for the ebook model. Content that's written as "context-sensitive Help" is probably not right for an EPUB, since an EPUB can't be used in a context-sensitive manner (yet). But, documentation that was written to be read in a linear order (or by section), may probably be a reasonable choice as an EPUB.

Remember that an EPUB can be read on a mobile device (on the bus or train while commuting), or on a desktop ereader (possibly alongside the application being documented). It's even possible that you could ship an ereader with your product, and use that as the "online Help" tool.

It might be worth doing a trial run for one product or set of documentation. Produce some EPUB files (and Kindle too) and make them available to your customers. Set up an easy way for people to provide feedback, and be proactive and get back in touch with those who have downloaded the files. It's likely that certain markets will find that their customers are more interested in ebooks than others.

The EPUB format is here to stay. It's a good idea to figure out how your documentation fits in to this delivery option.

TOOL RESOURCES

Adobe Technical Communication Suite – <http://www.adobe.com/go/technicalcommunicationsuite/>

Amazon Kindle tools – <https://kdp.amazon.com/self-publishing/help?topicId=A3IWA2TQYMZ5J6>

BlueGriffon EPUB Edition – <http://www.bluegriffon-epubedition.com/BGEE.html>

Calibre – <http://calibre-ebook.com/>

ComponentOne Doc-To-Help – <http://www.componentone.com/superproducts/doctohelp/>

DITA Open Toolkit – <http://dita-ot.sourceforge.net/>

DITA for Publishers plugin – <http://dita4publishers.sourceforge.net/>

DocBook to EPUB XSL stylesheets – <http://sourceforge.net/projects/docbook/files/epub3/>

EC Software Help & Manual – <http://www.helpandmanual.com/>

ElmSoft EPubFm – <http://elmssoftonline.com/>

epubcheck – <http://code.google.com/p/epubcheck/>

MadCap Flare – <http://www.madcapsoftware.com/flare/>

oXygen XML Editor – <http://www.oxygenxml.com/>

Sigil – <http://code.google.com/p/sigil/>

WebWorks ePublisher – <http://www.webworks.com/Products/>

USEFUL INTERNET RESOURCES

Castro, Liz. – <http://www.pigsgourdsandwikis.com/>

eBook Architects. – <http://ebookarchitects.com/resources/>

ePrdctn on Twitter. – <https://twitter.com/#!/search?q=%23eprdctn>

EPUB Resources. – <http://epubtest.com/resources.php>

IDPF – EPUB 3 Specification. – <http://idpf.org/epub/30>

IDPF – EPUB 2.0.1 Specification. – <http://idpf.org/epub/201>

AUTHOR CONTACT INFORMATION

Scott Prentice
Founder and President
Leximation, Inc.
122 H Street, San Rafael, CA 94901
415 485 1892
www.leximation.com

AUTHOR BIOGRAPHY

Scott Prentice is the founder and president of Leximation, Inc., and has been in the technical publication field since 1991. His work focuses on custom online Help development, FrameMaker (plugin and structure application) development, as well as custom web application development. He is very involved with DITA and created the DITA-FMx plugin for FrameMaker. He is an invited expert on the IDPF EPUB 3 Index working group.

MANAGING AND OPTIMIZING UNSTRUCTURED-TO-STRUCTURED CONVERSIONS

Edna Elle

Converting unstructured content to structured content requires technical expertise and careful planning. This paper describes how to manage and optimize the process and produce valid XML that is ready for your publication processes.

This presentation is geared toward technical writers, managers, and information designers who have XML background, have or will soon have a well-designed structured authoring environment in place, and have a large amount of content to convert or a need to continuously convert content. This document discusses mostly Adobe FrameMaker conversions plus conversions from Microsoft Word.

PLANNING: HOW MUCH TIME DO YOU NEED TO CONVERT YOUR CONTENT?

Conversions have two time components.

- Developing the Conversion Plan and Tools
- Performing the Conversions

DEVELOPING THE CONVERSION PLAN AND TOOLS

Allow a minimum of one month for this component depending on the size and complexity of your conversion. You may create more than one set of mappings, use third-party tools, and develop post-conversion tools to adjust content. Key components are to use well-designed structured authoring templates and to use representative samples of your content when you create mappings. The amount of time you spend before actual conversion will vary greatly.

You need to:

- Analyze your current documentation content. You may need to delete, rewrite, or reorganize content as well as change style formats. Consider creating a spreadsheet of different templates and format styles you intend to convert.
- Plan your migration so that you do not miss important release dates. Start with small or non-critical projects.
- Map format styles to your current elements.
- Develop tools that may include using third-party or open-source software, creating utilities within your target authoring environment, and/or writing your own tools.
- Document your process from preconversion through post-conversion tasks.
- Test your process before entering a full-scale conversion.

PERFORMING THE CONVERSIONS

At Oracle, the average time to convert a 200-page unstructured FrameMaker book to a structured FrameMaker book is 2 to 5 person-days.

As you assess how much time it will take to convert your content, consider the following factors :

- **Volume.** Consider the number of documents, the number of files per document, and the total page count. Five documents of 200 pages each takes less time than 20 documents of 50 pages each.
- **Complexity.** Evaluate how many different formats are used to convey information. More formats may be helpful in conversion and assure a good 1-to-1 mapping.
- **Variety.** Count the number of unstructured templates and tools in use. If your content is in multiple templates and tools, it will take more time.
- **Format Adherence.** Assess the extent to which the documents use only the formats in the current unstructured templates, without overrides, changes, or additions. Documents on ad hoc formatting alone do not convert well. If documents adhere poorly to current templates, this can easily add one or more days to each conversion of a 200-page book.
- **Similarity.** Determine if the current templates have similar or radically different editorial structures. Radically different editorial structures can add 2-4 hours of rewriting for each 200-page book.
- **Sharing.** Assess the extent to which content files are shared across books and whether this meshes with your intended structured authoring model.
- **Internal/External Cross-References.** Keep in mind that cross-referencing within books and linking to other books may require redesign.
- **Graphic Formats.** Determine if graphics are created as separate referenceable files or by using a graphics editor embedded in the authoring tool. Embedded graphics, depending on the conversion method, must be imported by reference. If a document has 100 graphics, this can take two or more hours.
- **Conditional Text.** Determine how conditional text in your current unstructured authoring environment is used in your new structured authoring environment.

RESOURCES: CAN YOU DO CONVERSIONS IN-HOUSE?

If you have the in-house expertise to develop a structured authoring environment in one of the many popular XML tools, than you probably can develop an in-house conversion. **You need a developer or you need developer resources.** The more robust your conversion, the less time you will have to spend in post-conversion clean-up and restructuring.

If you have not developed your structured authoring environment, you will probably contract this service. Consider that you may require on going support and adjustments to your conversions over time. Also, consider the support you will need if you acquire new and different content to convert. Overall, it may be more cost-effective to hire one or more developers. Additionally, a developer may be able to create various utilities and plug-ins that turn hours-long repetitive work into 30-second fixes.

PRECONVERSION CONSIDERATIONS: HOW DO YOU PREPARE DOCUMENTS A DIFFERENT AUTHORIZING MODEL?

It is likely that your unstructured authoring model is different from your structured authoring model. Therefore, conversion is more than just structuring your current content. Simple differences might include the fact that you unstructured documents skip heading levels that are required in your new Document Type Definition (DTD). More radical differences may include "flattening" your content presentation or requiring elements such as tables or ordered lists in certain locations in your content.

To achieve a good post-conversion result, you may want to substantially edit your content, add "meta" information, or add formats to your current templates that are familiar to your writers.

The following list discusses some of the authoring differences and how you can address these.

- **Heading levels are skipped**
Consider editing your content, elevating sections, or demoting content to paragraph- or list-type content.
- **Moving to topic-based authoring**
Your current templates may not contain topic-specific formats for different topic types. Consider adding and applying new paragraph formats in your current unstructured templates or adding markers that can be changed to meta data in a post-conversion step.
- **Heading levels do not include content**
Some DTD require content before the next section- or topic-type element is added. Add this content prior to conversion or add boilerplate text to meet the DTD requirements so you can edit the document after conversion.
- **Lists may nests too deep**
While list are usually recursive, most templates provide formatting for list beyond a certain depth. Flatten lists to the number of levels supported before conversion.

PRECONVERSION PLAN: WHAT ARE THE COMPONENTS OF A PRECONVERSION PLAN?

A preconversion plan allows you to perform as many cleanup tasks as possible in your familiar, unstructured environment.

- **Correct Irregular Template Usage**
- **Address 1-to-Many Format Mappings**
- **Supply or Remove Boilerplate Text**
- **Use the Required File Naming Conventions**
- **Establish Required Directory Structures**
- **Change Variable Names and Variable Definitions**
- **Use Generated List Files to Optimize Results**

CORRECT IRREGULAR TEMPLATE USAGE

Examine your current template usage for the following issues.

- Format overrides or ad hoc formatting. Remove all format overrides or ad hoc formats. Determine if the information still makes sense as presented. If not, apply different formats.

Word documents more frequently use all or mostly ad hoc formatting. If this is the case, your conversion results will be very poor. Use well-defined format styles in Word.

- One-off formats. Writer-created formats that are not part of your standard templates must be removed or considered for conversion.
- Heading hierarchy. Ensure that the unstructured heading hierarchy matches your structured hierarchy. Skipping heading levels, in some DTDs, is not an option.

ADDRESS 1-TO-MANY FORMAT MAPPINGS

Determine whether your current templates use a single format in multiple situations. Often, this does not convert well. For example, a paragraph format called Body, used for regular body paragraphs, glossary definitions, paragraph in certain types of lists, and in informal sections, will be difficult to convert because typical conversion mappings allow only a small amount of context. Consider renaming certain paragraph formats for a better conversion that allows elements to wrap correctly in their corresponding parent element.

For the paragraph format called Body:

- In regular body paragraphs, leave it as Body.
- In glossary definitions, append Body to be glossBody.
- In certain lists, append Body with a list-type label, such as, variablelistBody
- In informal sections, append Body to be informalBody.

SUPPLY OR REMOVE BOILERPLATE TEXT

Your unstructured and structured authoring tools may or may not automatically supply certain text strings. For example, Note, Caution, Syntax, Step, Example, etc. may be supplied by the template. Before conversion, add or remove strings as necessary to reduce confusion after conversion.

USE THE REQUIRED FILE NAMING CONVENTIONS

Depending on your authoring tool and how you plan to use the XML, file naming conventions are more strict. File names must start with a letter and contain no spaces or special characters. This includes graphic file names. Additionally, graphic file names should use browser recognizable file extensions if you plan to create HTML output. Also, consider whether your authoring environment, content management tools, or archiving system has certain file naming and directory structure requirements.

ESTABLISH REQUIRED DIRECTORY STRUCTURES

Consider your required directory structure after converting. Often, conversion is faster and less error-prone if these structures are established prior to conversion.

CHANGE VARIABLE NAMES AND VARIABLE DEFINITIONS

Variables become entities in XML therefore, you must ensure that variable names are valid prior to conversion.

Variable names cannot use characters that are illegal in XML. *Spaces, special characters, and parentheses are not allowed in XML entity names and these are the most commonly used illegal characters.* Variable names must start with an alphabetic character and can only contain:

- letters
- numbers
- underscores
- hyphens
- periods

Variable definitions may contain certain character formatting that is not recognized in your Structured FrameMaker environment or is not preserved in conversion. Consider deleting all formatting from variable definitions and restoring the formatting after conversion. You will likely need to redefine variable definitions to reflect new editorial style guidelines.

Remove all unused, unnecessary variables.

USE GENERATED LIST FILES TO OPTIMIZE RESULTS

Use generated lists (in FrameMaker) of certain types of markers to optimize your conversion results and save time by reducing problems such as cross-references that become unresolved, duplicate markers, mislabeled markers, and so on. Losing cross-references in large documents or conversions is painful.

Use the following types of generated lists:

- **Cross-Ref.** Find and fix the following problems with cross-reference markers:

Duplicate unique id markers (numbers)

Markers without unique ids

Illogical markers, such as a Chapter-type cross-reference marker on a Glossary

Mislabeled markers that should have been index markers or other types of markers

- **Hypertext.** For message url-type Hypertext markers, do the following:

Delete the Hypertext markers.

Ensure that the URL is visible in the document and uses a hypertext-type character format that can be mapped to an element.

- **TargetMarkerNames.** Remove or rename duplicates.

- **Index.** Review index markers and do the following:

Ensure that FrameMaker building blocks markup is correctly applied and spelled. In particular, the <\$npage> building block must start the entry.

Ensure that <\$starange> and <\$endrange> index markers are matched sets or consider removing these building blocks.

Remove character formatting that is not part of your new templates.

Insert a Set of Generated Lists

Insert alphabetical generated lists of markers to easily identify problems you need to correct.

1. From the book file, choose Add > List of > Markers (Alphabetical).
2. In the Setup List of Markers dialog box, move the marker type to the Include Markers of Type and then change the suffix in the Suffix field according to the lists type list. Create one generated file per marker type.

Check the box Create Hypertext Links and click Add.

3. Examine each generated list and fix issues.

Fix Cross-Ref Markers

From the generated list of Cross-Ref markers, hyperlink using Ctrl+Alt to problematic markers and delete each one from the individual files in your book

The following shows the types of Cross-Ref markers that must be deleted or changed in Bold. It is followed by an explanation.

19333: !Head2: Audience 2-1

19893: ChapHead: Title of Chapter 1 1-1

19893: ChapHead: Title of Chapter 1 2-1 - duplicate, will break

19893: ChapHead: Title of Chapter 1 3-1 - duplicate, will break

19893: ChapHead: Title of Chapter 1 4-1 - duplicate, will break

20953: !Head3: Action Icon Menus 5-8

Adapter Metadata Fields - missing a number, will break

...

88001: !Head1: About Content Retention Management 4-2

88304: Paragraph: ??? URM custom metadata fields pane ??? 4-11 - the ?? indicates the wrong type of x-ref applied because it can't get a number from a paragraph format that does not have a number

88700: !Head2: Viewing the Configuration Settings 4-8

...

99797: !Head3: Deleting the Adapter Metadata Fields 3-14

!Head2: Variable Settings 3-9, can never be resolved without a number.

metadata fields:default fields 8-2 - likely mislabeled index markers, change to an index marker or remove; all x-ref markers get stripped

31893: ChapHead: Glossary Glossary-1 - likely won't resolve

Fix Resulting Broken Cross-References

After you have removed all problematic Cross-Ref markers, it is likely that your book will have unresolved cross-references. Generate your book and fix any resulting broken cross-references.

Remove Url-Type Hypertext Markers

From the generated list of Hypertext markers, scan the list for url-type Hypertext markers. These markers have the following form:

Message URL <http://www.company.com>

When you find such a url-type Hypertext marker, make sure the text in the document shows the complete url, then remove the markers. Apply a hyperlink-type character format that will map to an element.

Fix TargetNameMarker Markers

From the generated list of TargetNameMarker markers, hyperlink using Ctrl+Alt to duplicate markers and delete or change each duplicate.

Fix Index Markers

From the generated list of Index markers, search for a left or right angle bracket. When you find one, examine the index marker for two possible issues and modify accordingly. Hyperlink using Ctrl+Alt to edit the problematic markers.

The following shows the types of Index markers that should be modified, followed by an explanation.

metadata:<\$nopage>adapter - improper use of FM building block, <\$nopage> must appear the beginning of the entry, thus <\$nopage>metadata:adapter

<ITALICS>database table<Default Para Font> - change to a character format in your current templates, for example <Italic>databasetable <Default Para Font>

The following shows <\$startrange> and <\$endrange> building blocks that are mismatched. The example shows that there are no <\$startrange> blocks for any of the <\$endrange> building blocks. Consider eliminating the startrange/endorange building blocks if your primary delivery format is HTML and mobile formats.

<\$endrange>hardware activation file:deleting 165

<\$endrange>hardware activation file:displaying contents of;hardware activation file:comparing to current features 152

<\$endrange>hardware activation file:installing 175

<\$endrange>hardware activation screen reference 1

CONVERSION METHODS: HOW ARE DIFFERENT CONVERSION METHODS PREPARED?

This sections discusses FrameMaker and Word conversions.

WORD CONVERSIONS

There are two methods of converting Word documents. Both are dependent on well-defined format styles that are consistently used. Word documents too frequently contain all or mostly ad hoc formatting where a Heading 1 is actually Normal set to 24-point bold type and worse, lists are Normal defined to include a bullet. Such formatting results in structured conversions that have almost no structure.

Word-to-FrameMaker-to-Structured FrameMaker

Use this method if you have numerous Word documents to convert that use the *same or very similar templates*.

Open Word documents directly in FrameMaker. Ensure that the existing Word styles are present in the paragraph catalog. Also, ensure that the named character catalog formats are applied for bold, italic, code, and other inline text formats.

1. Using the existing Word styles, write a FrameMaker conversion table to structured authoring. For information on creating a conversion table, refer to Adobe FrameMaker documentation and to *Moving from Unstructured Documents to Structured XML* by Tom Aldous at <http://www.slideshare.net/abelsp/moving-from-unstructured-documents-to-structured-xml>
2. To convert embedded graphics that are now in your FrameMaker file, follow these steps:

Choose **File > Utilities > HTML Setup**.

In the HTML Setup dialog box, click the **Options** button.

In the HTML Options dialog box, under **File Format** for Images choose the format that is appropriate for your project.

Save the file as HTML.

Graphic files are now written into your current directory, available for you to reimport into your document by reference.

Word-to-HTML-to-XML

Use this method if you have numerous Word documents to convert that use different templates. This method provides a good generic conversion that you can continue to use for content you may receive from developers.

1. Open your Word document and ensure that the Word format styles are used to format the document. Character formats for inline bold, italic, and code convert very well even when named character styles are not used.

Lists must use Word list format styles, not a Normal paragraph format that has a bullet or number included.

If the entire content of the Word document is in a table (usually to provide a side-heads appearance), remove the over-wrapping table.
2. Save your Word (or Excel) document as HTML using **Save As Web Page** utility.

Choose **File > Options**.

From the Options dialog box, choose the **General** tab, then click the **Web Options** button.

In the Web Options dialog box, clear the "Disable features not supported by" and the "Rely on CSS for font formatting" check boxes.

Click **Save As** and choose **HTML**. When you save as HTML, embedded graphics are saved as separate files that are referenced by the HTML file.

3. Use XSL to write a transformation from the HTML DTD to your DTD. Use one of the various open-source tools as the starting point for your transformation. Remember to use the DTD declaration that is specific to your authoring environment when you create your transformation. For more information, see:

Transform XHTML article to Docbook <http://stackoverflow.com/questions/8189155/transform-xhtml-article-to-docbook>

Converting HTML to Docbook SGML/XML Using html2db <http://ppadala.net/projects/tidy/>

4. Open the resulting XML file in your structured FrameMaker environment.

FRAMEMAKER CONVERSIONS

There are several conversion paths from unstructured FrameMaker to structured FrameMaker.

Unstructured to Structured FrameMaker Two-Step Process

This process preserves 90% or more of the formatting of your document. This is the best process for preserving cross-references, index markers, and text entities (variables), and more. This is a two-step process because you first convert from your current unstructured templates to an unstructured form of your XML templates. This allows you to more easily see formats that did not map well before structuring. It also allows you to perform a book-wide conversion.

This process includes the following steps:

1. Create an unstructured form of your current XML templates by stripping the structure and redefining your cross-reference formats to be paragraph-based.
2. Ensure that you have performed all of the tasks described in Preconversion Considerations: How Do You Prepare Documents a Different Authoring Model and Preconversion Plan: What Are the Components of a Preconversion Plan.
3. Remove all generated files from your book (table of contents, lists of examples/figures/tables, and index).
4. Using tools such as TemplateMapper (I like this tool best. It is very easy to use and very fast to setup. Contact Chris Despopoulos directly to purchase this tool, despopoulos_chriss@yahoo.com, about \$200.) or Silicon Prairie software (<http://www.siliconprairiesoftware.com/>). Map your current template formats to formats in the unstructured form of your XML templates. TemplateMapper allows you to map paragraph, character, table, cross-reference, and conditional text formats as well as variables, master pages, and reference pages. Additionally, the tool allows you to apply specific template types to each file, such as mapping chapter formats to chapters or mapping preface formats to prefaces, etc.
5. After mapping, inspect your files for visual and format compliance with the unstructured form of your XML templates. Make any necessary corrections:

Lists must use correct nesting order. Level-two lists cannot be used outside of the context of a level-one list.

Subsequent paragraphs or code belonging to a list item must use formats designed only for that purpose or your lists stop during the structuring process.

Apply proper heading structure. Section headings must be in hierarchical order if your DTD does not use recursive headings.

Remove any type of forced formatting including tabs, multiple non-breaking spaces, multiple blank spaces in anything that is not code, blank paragraphs. or Shift-Return line breaks.

Remove table row spans because these do not structure well. Table column spans may remain.

Correct lists that mix ordered and unordered list items in a single list.

6. Using the structured conversion mapping table that you have created, structure your entire book at once.
7. Import the structured template formats to all of the files in your book. Update the book and import the EDD. Use FrameMaker's validation function until your book is valid.

Unstructured to Structured FrameMaker One-Step Process

This process works if your current templates are simple and very similar in structure to your XML templates. I do not recommend this process because it very unlikely that you will be able to make a good 1-to-1 mapping between your existing formats and your XML elements. When you perform the structuring step, your resulting files are often hours or days away from being valid. This is especially difficult for users who are new to XML or new to the DTD.

This process includes the following steps:

1. Ensure that you have performed all of the tasks described in Preconversion Considerations: How Do You Prepare Documents a Different Authoring Model? and Preconversion Plan: What Are the Components of a Preconversion plan?
2. Using the structured conversion mapping table that you have created, structure your entire book at once.
3. Import the structured template formats to all of the files in your book. Update the book and import the EDD. Use FrameMaker's validation function until your book is valid.

The majority of your validation cleanup will be lists that start or stop prematurely, lists that do not nest correctly, tables that do not format correctly, and cross-references that become broken.

Unstructured FrameMaker to HTML to XML

Use this method to quickly convert simple, short documents of 50 pages or less. This method is useful and generic because it allows you to convert many different template formats quickly. However, certain formatting is lost in this conversion as well as index terms and cross-references (though the cross-reference text remains).

1. From your FrameMaker file, choose **File > Utilities > HTML Setup**.
2. In the HTML Setup dialog box, from the **Map** menu, choose **Paragraph Formats**. Then change the From and To values to map the paragraph formats to HTML elements. For lists, adjust the Nest List at Depth value.

Now choose **Character Formats** from the **Map** menu and again, adjust the **From** and **To** values.

Finally, choose **Cross-Reference Formats** from the Map menu and adjust the **From** and **To** values.

3. In the HTML Setup dialog box, click the **Options** button.

In the HTML Options dialog box, under **File Format for Images** choose the format that is appropriate for your project.

4. Save the file. This saves the HTML mappings to a FrameMaker Reference Page. You can use this file for other conversions that use the same template by importing the Reference Page formats.
5. Now save the file as HTML.
6. Use XSL to write a transformation from the HTML DTD to your DTD. Use one of the various open-source tools as the starting point for your transformation. Remember to use the DTD declaration that is specific to your authoring environment when you create your transformation. For more information, see:

Transform XHTML article to Docbook <http://stackoverflow.com/questions/8189155/transform-xhtml-article-to-docbook>

Converting HTML to Docbook SGML/XML Using html2db <http://ppadala.net/projects/tidy/>

Open the resulting XML file in your structured FrameMaker environment.

ACKNOWLEDGEMENTS

The knowledge was developed in-house with George Eross, Principal Developer; Robert Crews, Senior Documentation Manager; Ted Gilchrist, Principal Developer; Ken Kipnes, Principal Developer. Adobe FrameMaker documentation and generally available information on tools, DTDs, and transformations was also used.

ADDITIONAL RESOURCES

Moving from Unstructured Document to Structured XML: It's Easier Than You Think, Thomas Aldous, Integrated Technologies, Inc. <http://www.slideshare.net/abelsp/moving-from-unstructured-documents-to-structured-xml>

Migrating from Unstructured to Structured FrameMaker, Sarah O'Keefe Scriptorium Publishing, <http://www.scriptorium.com/Unstructured2StructuredFM.pdf>

AUTHOR CONTACT INFORMATION

Edna Elle
Senior Principal Program Manager
Oracle
500 Oracle Parkway
Redwood City, CA 94065
edna.elle@oracle.com

AUTHOR BIOGRAPHY

Edna Elle is a Senior Principal Program Manager for Documentation at Oracle with 17 years experience in authoring tools, documentation conversion, and HTML accessibility features for blind and low-vision readers. Before she became a Senior Principal Program Manager, Edna was the Director of the Oracle DocTools group. Edna has worked on a variety of authoring tools including FrameMaker, Arbortext Epic, Interleaf, Dreamweaver, and proprietary authoring tools. She has designed conversions that include SGML-to-MIF, MIF-to-MIF, and XML/SGML-to-XML. Since 2007, Edna has handled the many large-scale conversions for the Oracle acquisitions of Siebel, Stellent, BEA, Sun Microsystems, and others. Edna holds a patent for accessible definition links in online documentation (US7493560).

PLANNING AND CREATING ENGAGING INFOGRAPHICS

Michael R. Opsteegh

Despite being over used and often abused by mainstream media in the last few years, infographics are a powerful way of communicating large amounts of disparate data to inform or persuade your audience in a manner that is often more engaging than text alone. Advancements in technology, social media, the internet of things, and biomedicine have created an explosion of data that paragraphs of text and spreadsheets cannot adequately explain. We have seen an explosion of infographics appearing in magazines, blogs, and business presentations in the last few years. Technical communicators need to understand how infographics work before creating their own or working with a designer.

Information graphics, or infographics, can be a powerful way to persuade your audience and convey copious amounts of data in a manner that is digestible and entertaining to the reader. Infographics can be simple charts or graphs that help the reader visualize the data, or they can be complex diagrams that tie in disparate data to convince the reader of causal relationships. No matter how large or small, infographics can be an island of eye candy in a sea of text, and readers are more likely to rely only on the infographic than any text that may be provided to support it. To this end, infographics can be persuasive, but they can also be misleading and fraught with ethical missteps. Since infographics can rely on large data sets and, by their very nature, are designed to amplify certain aspects of the data over others, we must be careful in how we visualize our information, which data we chose to include or omit, and how we label our data.

GETTING STARTED

Planning and creating engaging infographics takes time, thoughtfulness, and creativity. You should let practice and patience guide your infographic projects and understand that each project presents its own unique challenges.

Often, your first ideas are not your best. Sometimes, the story you wanted to tell isn't supported by the data or the data are affected by factors you hadn't considered, like inflation. Sometimes, the data doesn't tell a story at all or the story is predictable and uninteresting. Sometimes, your first method of data visualization isn't the best suited for the situation. Practice and patience will help you hone the critical reasoning and creative skills required for creating engaging infographics.

REQUIRED SKILLS

Planning and creating engaging infographics requires a diverse skill set that traverses liberal arts, social sciences, and mathematics. As technical communicators, we are used to taking multidisciplinary approaches to our work, and our skills lend themselves to creating infographics.

They say the MFA is the new MBA (Tepper), which is to suggest that creativity is becoming one of the most valued attributes in business. As technical communicators and infographic designers, the most valuable skill you have is the ability to tell a story. You must possess a curiosity that drives you to find the story in the data, and you must possess the creativity to tell that story.

You must have some knowledge within the sphere or industry in which you are working. This knowledge will direct you.

Additionally, a design acumen is a useful and desirable quality to have, but it isn't required. You can supplement a lack of design skills by hiring a graphic artist to render your infographic for you or use a service like Visual.ly (<http://visual.ly>) or Infogr.am (<http://infogr.am>). By the same token, someone who is a graphic design wizard doesn't necessarily have the skills to analyze the data and suss out a compelling story that underpins an infographic.

In addition to soft skills mentioned above, you must be comfortable with math and statistics. If you are not comfortable, you must have a working knowledge of statistical terms and functions.

HELPFUL TOOLS

Like any multidisciplinary art form, creating infographics requires the technical communicator to be familiar with a variety of tools. I'd suggest that you have an intermediate to advanced knowledge of the following tools:

- Evernote or Microsoft Word

You need a place to record your thoughts, ideas, questions, and findings. I prefer Evernote because it allows you to easily save information from a variety of sources, and it syncs easily with mobile devices.

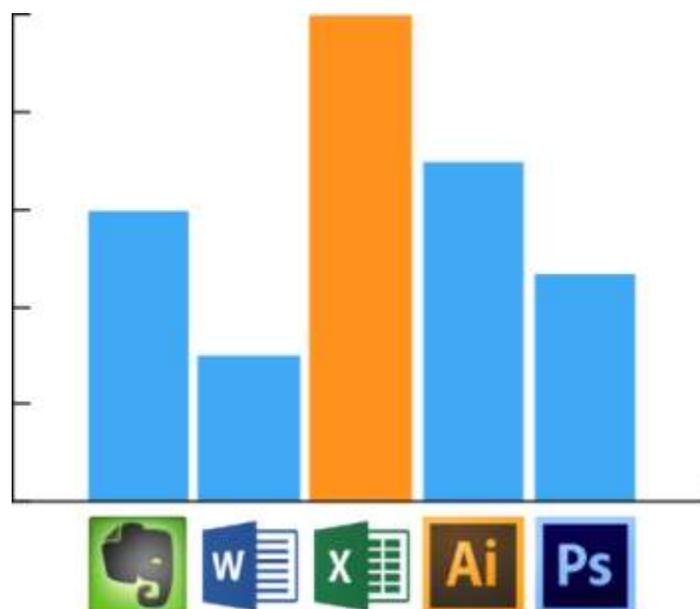
- Microsoft Excel

You need a tool to help you organize data, perform calculations, finesse the data by rounding, and quickly visualize your data. Excel seems to be the Swiss Army knife in this regard, and it's ubiquitous.

- Adobe Illustrator or Adobe Photoshop

Finally, you need a tool in which to create your final product. I prefer Illustrator. Photoshop does amazing things with photographs, but Illustrator is better suited to illustrations, line art, and vector graphics. Also, Illustrator has some, albeit antiquated, graphing and chart tools.

You may find other tools that suit your needs better. Feel free to replace these tools or complement them with your own.



The tools you use may vary, but the most important tool is the spreadsheet software that you use to organize and visualize your data.

With the recent popularity of infographics, new tools are continually being developed. Since an infographic can cover virtually any topic, most infographic-specific tools have a few very specific templates that you can use for free or for a little money, but these tools are designed to entice you to pay for a custom infographic. I have already mentioned two services like this: Visual.ly and Infogr.am.

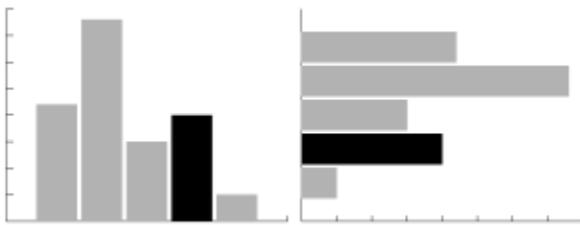
For the purposes of this presentation, we're going to focus on creating our own infographics.

UNDERSTANDING TYPES OF DATA VISUALIZATIONS

There are a wide variety of ways to visualize data. The one you choose (or ones you choose for complex infographics) should depend on what kind of data you are displaying and how the data relate to each other (Wong, Tufte). Here are some of the most common data visualizations:

- Bar charts

Bar charts show basic size comparisons between multiple subjects.



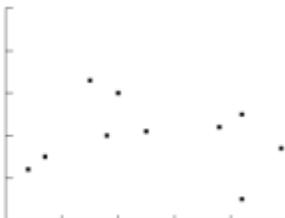
- Pie charts and doughnut charts

Pie charts show how a group of data are broken down into parts when the sum of the parts equals a whole (or 100%). Doughnut charts are the same as pie charts, but with the center blown out.



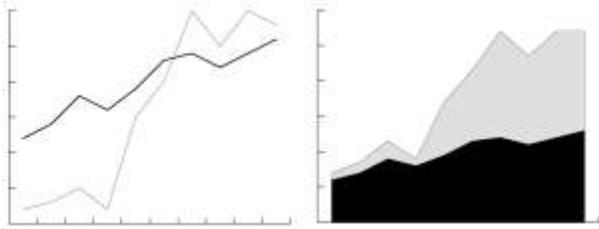
- Scatter plots

Scatter plots display individual points of data and can often reveal a trend. You probably created many scatter plots in your high school math classes.



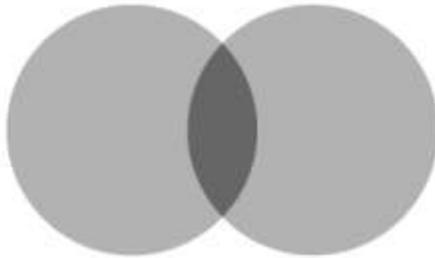
- Line graphs and area charts

Line graphs are like scatter plots, but the points of data are spread along regular intervals and are connected by a line. An area chart is a line graph that is filled in.



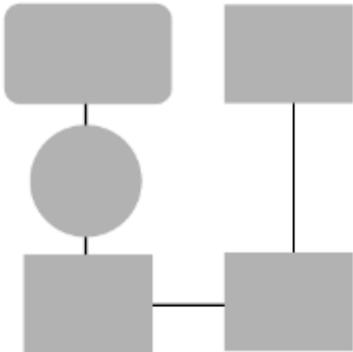
- Venn diagrams

Venn diagrams show the possible relationships between discrete items.



- Flow charts

Flow charts are good at displaying relationships and decision trees. Elements in the flow chart are connected by lines indicating a relationship.



- Pictograms

Pictograms are created using glyphs or pictures that represent some physical aspect of the data they represent, where each glyph represents a set number.



The first attempt at visualizing the data may not be the best suited. Try multiple types of visualization to see if a new story emerges or to verify that your data are correct, as you may notice inconsistencies when viewing the data differently.

CREATING YOUR INFOGRAPHIC

The exact steps you take to create your infographic may vary from project to project, but your overall workflow will look something like this (Wong):

1. Research
2. Edit
3. Plot
4. Review

RESEARCHING

Research is that critical stage where you gather information. If your information is not accurate or reliable, there is no point in continuing. Ensure that you gather information from authoritative and independent sources. Be sure to obtain permission to use the data.

EDITING

In this stage, you will find your story. Identify outliers in the data that might make the story more interesting, but be aware that not all of them do. Identify gaps in the data and locate the missing data if you can. Filter and simplify your data; large data sets may contain information that is superfluous to your story. Be careful not to omit data that has a bearing on the outcome of your story. Finesse the data by dropping unneeded decimal places and rounding the numbers. Determine if percentages or absolute numbers are appropriate for your infographic.

PLOTTING

To many, the plotting may be the most exciting part of creating an infographic. This is the stage where you chose the right visualization for your data, choose the appropriate scale and increments, and affix the appropriate labels. These considerations will be discussed in depth during the presentation.

Additionally, this is the stage where you would develop a wireframe if you were creating a complex infographic. Here, you would finalize the dimensions of the illustration based on the media and obtain a signoff from the stakeholders.

REVIEWING

Proofread your infographic. Check the data visualization against your research to ensure that it makes sense. Check your data visualization against other data that might be available to insure your visualization is accurate. Ask a colleague to review your infographic, preferably someone unfamiliar with your research.

Once your review process is complete, you are ready to publish your infographic.

ETHICAL CONSIDERATIONS

Infographics are powerfully persuasive tools. With great power, however, comes great responsibility. Omitting data that would have a bearing on the reader's decision is obviously an egregious offense. It isn't always intentional, so you should be on the lookout for some of the more common missteps.

You may be inclined to focus on a positive part of the story. For example, you may create a line chart showing test scores were up by 5% last quarter, but if you omit the the prior three quarters where test scores had plummeted 10%, you aren't giving the reader a clear picture.

You may be inclined to compare unlike things or not choose the right measurement by which to compare them. For example, if you showed two stocks had risen by \$10 each, the angle of the slopes would be the same. If, however, one stock had risen from \$50 to 60, while the other from \$10 to 20, the performance of the latter is far better in terms of percentage gains.

These examples barely scratch the surface of the ethical considerations that must be made while planning and creating infographics. Be vigilant in your projects.

CONCLUSION

We live in a data-driven world where we are bombarded by big data and measure everything in economic terms. We are drowning in information. It is overwhelming. We can throw our users a life preserver by helping them visualize data, complex processes, and relationships between disparate objects—the information required to make daily decisions. Infographics can help us influence our readers, managers, clients, and colleagues.

RESOURCES

Drell, Lauren. "This Man Makes Data Look Beautiful." *Mashable* (11 December 2012).
<http://mashable.com/2012/12/11/data-visualization-jer-thorp>.

Friedman, Vitaly. "Data Visualization and Infographics." *Smashing Magazine* (14 January 2008).
<http://www.smashingmagazine.com/2008/01/14/monday-inspiration-data-visualization-and-infographics>.

Meyer, Eric K. *Designing Infographics* (Indianapolis, Ind: Hayden Books), 1997.

Tufte, Edward R. *Invisioning Information* (Cheshire Conn: Graphics Press), 1990.

---. *The Visual Display of Quantitative Information, 2e* (Cheshire Conn: Graphics Press), 2001.

---. *Visual and Statistical Thinking: Displays of Evidence for Making Decisions* (Cheshire Conn: Graphics Press), 1997.

Wong, Donna M. *The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures* (New York: W.W. Norton & Co.), 2010.

REFERENCES

Tepper, Steven. "Is an MFA the New MBA?" *Fast Company* (28 March 2013).
<http://www.fastcompany.com/3007541/mfa-new-mba>.

Tufte, Edward R. *The Visual Display of Quantitative Information, 2e* (Cheshire Conn: Graphics Press), 2001.

Wong, Donna M. *The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures* (New York: W.W. Norton & Co.), 2010.

AUTHOR CONTACT INFORMATION

Michael R. Opsteegh
Sr. Technical Writer
Eyefinity
38 Discovery, Suite 250
Irvine, California 92618-3129 USA
+1 949.390.8320

AUTHOR BIOGRAPHY

Michael Opsteegh has been a technical writer in the software and financial services industries for nine years and is currently a senior technical writer for Eyefinity, which supports eyecare professionals with industry-leading software and services. Michael specializes in professional and technical communications that include user guides, websites, policies and procedures, and illustration and he adds value through research, insight, and enthusiasm. He holds a masters degree in English, rhetoric, and composition and a certificate in technical and professional writing from California State University, Long Beach. He has written articles on comics and typography in technical communication that have appeared in STC's *Intercom*. Michael occasionally blogs on the topics of typography and page design at bestfontforward.com.

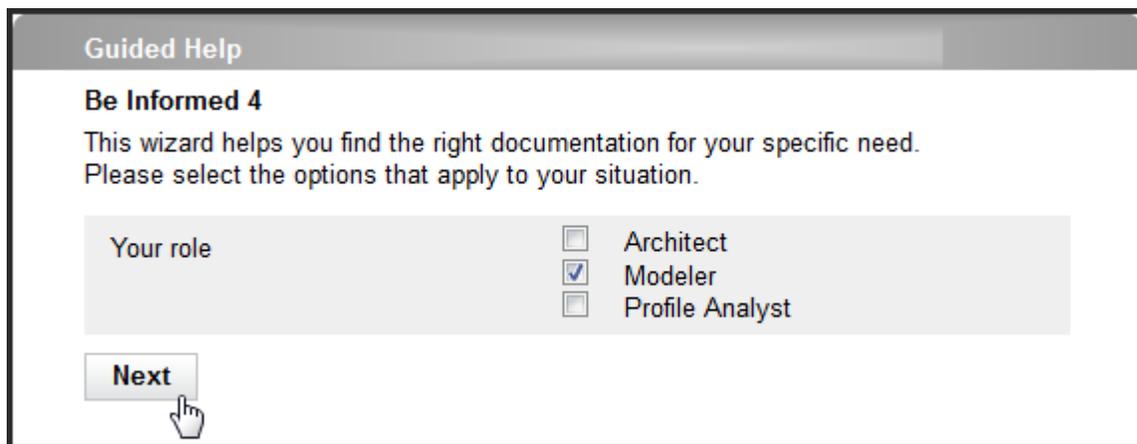
USING SEMANTIC TECHNOLOGY TO CREATE PROCESS-DRIVEN DOCUMENTATION

Kees van Mansom

For most product manufacturers, delivering adequate documentation is a constant challenge. The increasing complexity of products and the stricter regulations for product documentation often lead to extensive documentation sets with 500+ pages manuals on-line or in print. The question is: how effective is all this documentation?

SMART, USER-CENTRIC DOCUMENTATION

Users can be overwhelmed by long, jargon-loaded user manuals, and frustrated by a lack of clear answers. When customers don't find a solution, blame inevitably falls on the product and the company. Bad documentation - incomplete, unclear, out of date - creates a negative engagement with products and brands.



The screenshot shows a 'Guided Help' window with a title bar. Below the title bar, the text reads 'Be Informed 4' followed by 'This wizard helps you find the right documentation for your specific need. Please select the options that apply to your situation.' Below this text is a form with the label 'Your role' and three radio button options: 'Architect', 'Modeler' (which is selected with a blue checkmark), and 'Profile Analyst'. At the bottom left of the form is a 'Next' button with a mouse cursor pointing to it.

Figure 1: Smarter documentation asks users what they need to know

Smart documentation leads users to success and satisfaction. These are the users who share their positive experience with family and friends, and convince them to become customers in turn. Using semantic technology, we can create the smart documentation that guides each customer through the actions, decisions, and choices they can take to maximize their user experience. Semantic Technology encodes meaning and separates it from content and application codes.

BUSINESS CASE

Semantic technology can also transform the documentation process. The user manual is no longer created after the fact, but built in tandem with the product. Changes in content or structure are tested and published on the fly, because the documentation is instantly executable in a web browser.

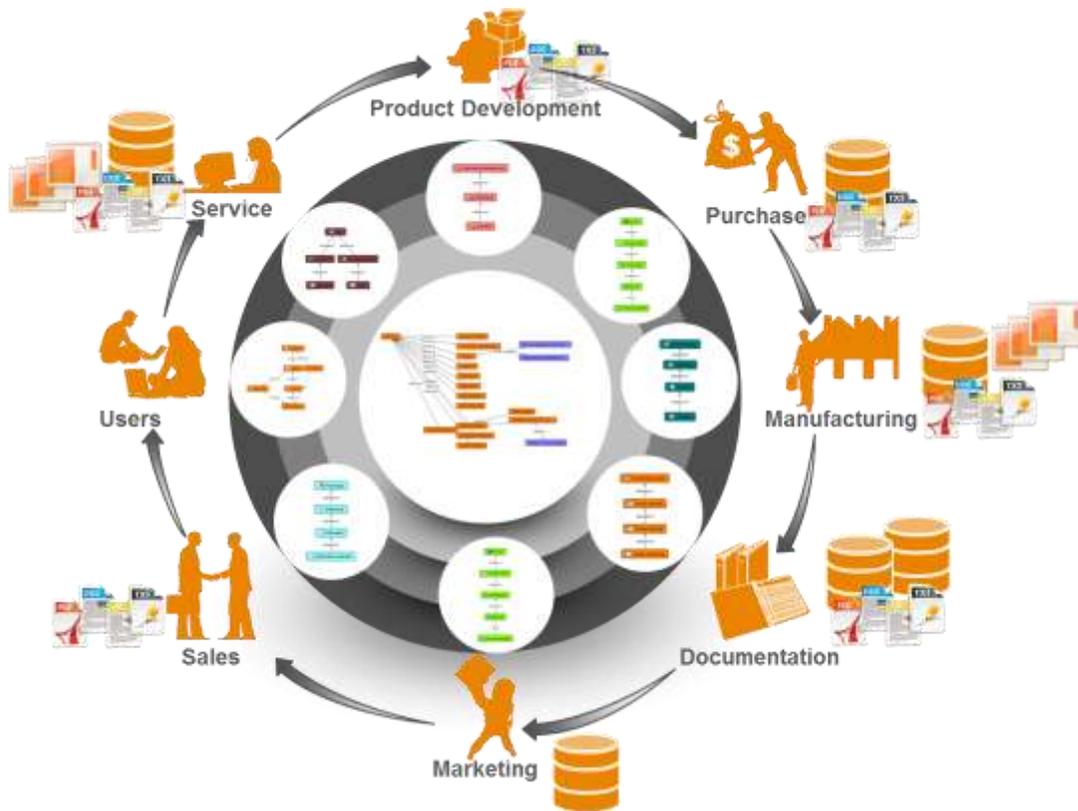


Figure 2: Documentation in the product life cycle

Since the gap between development and documentation disappears, documentation is not frozen describing last year's model. Documentation teams and support desks can deliver real value to customers, instead of chasing support tickets. The total costs of documentation will decrease and become more predictable as a result of an optimized process for various purposes and audiences throughout the whole product life cycle.

USING SEMANTIC TECHNOLOGY

With semantic technology we can describe the functionality, processes and context of a system or device in a model, add meaning to concepts and relations and represent knowledge separately from documents, data and programming code. It enables us to add a knowledge layer on top of it to support the context specific needs of actors in the product lifecycle. By doing so, all output forms can be combined in one smart user assistant.

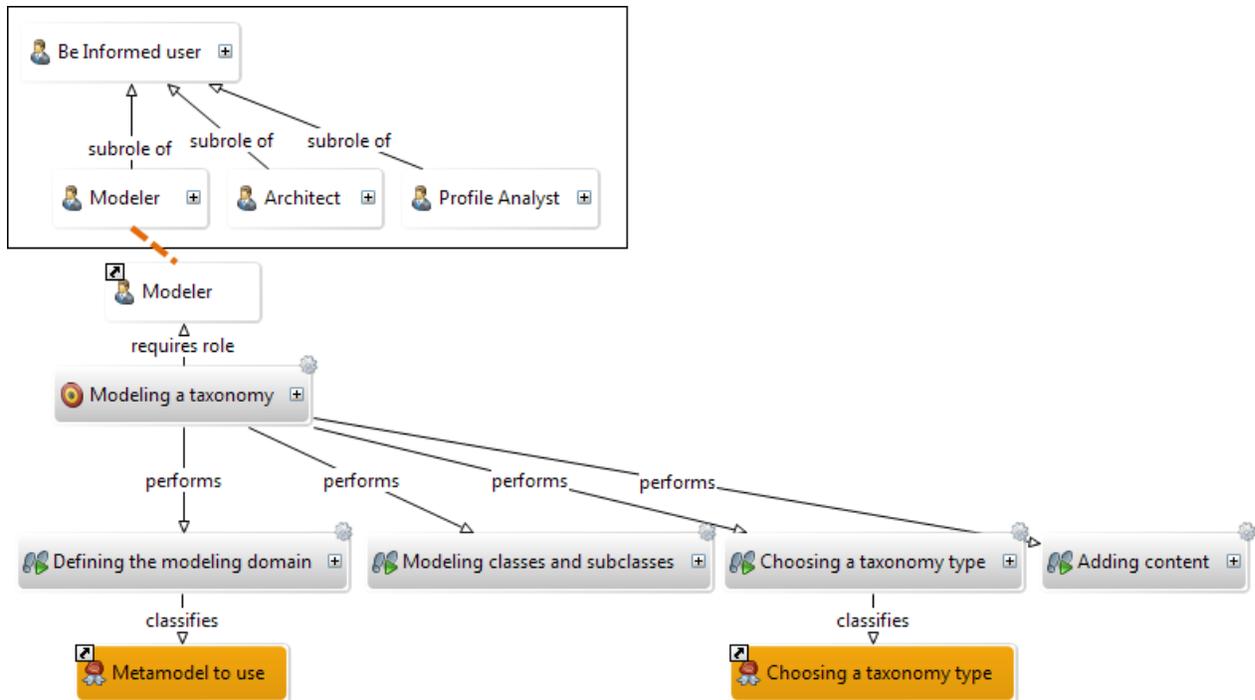


Figure 3: Use semantic technology to create models of the documentation

Implementing semantic technology is a logical next step in the evolution of documentation. Semantic technology allows us to make customer support part of the product in a seamless and effective manner. This makes our products better and improves customer engagement with products and brands.

GET READY FOR THE NEXT WAVE IN DOCUMENTATION

At the STC Summit 2013 in Atlanta, I will demonstrate how we can use semantic technology to create process-driven documentation. I will do this by live modeling with a semantic toolset. As the models in the semantic environment are instantly executable, we can work together on some documentation concepts, change the models and review the results in a web browser. The session will be highly interactive with plenty of opportunities to influence the modeling process and simulate specific situations.

When: **May 6th, 2:00 pm** (EDT)

Level: **Advanced**

DOWNLOAD THE FULL WHITE PAPER

My white paper “Improve Product and Brand engagement through smart documentation” is available for download on: <http://info.beinformed.com/stc13>.

REFERENCES

Davis, M. “Semantic Wave: Industry Roadmap to Web 3.0.”, 2007-2013

Horton, W. “Designing and Writing Online Documentation”. 2nd ed. John Wiley and Sons: New York, 1994.

Hughes, M. "Moving from Information Transfer to Knowledge Creation: A New Value Proposition for Technical Communicators." *Technical Communication* 49, no. 3, 2002: pp. 275–285.

Johnson-Laird, P.N & Mackiewicz, R. "Reasoning from connectives and relations between entities". *Mem. Cogn.* (2012) 40 : 266-279 (available on-line on Springerlink.com).

Mansom, C. van "The Flexible Intranet". *Intercom*, March 2008, pp. 6 – 11 (cover article).

Mansom, C. van "How information becomes knowledge". *Communicator Autumn 2007*, pp. 12 – 14.

Mansom, C. van "How knowledge becomes usable". *Communicator Winter 2007*, pp. 34 – 36.

Norman, D.A. "The Design of Everyday Things". *Basic Books*, 2002 (previously published as "The Psychology of Everyday Things", 1989).

AUTHOR CONTACT INFORMATION

Kees van Mansom
Senior Information Architect
Be Informed
Wapenrustlaan 11-31, 7321 DL Apeldoorn
The Netherlands
k.vanmansom@beinformed.com
+31 6 250 19 874



AUTHOR BIOGRAPHY

Kees van Mansom is a senior information architect at Be Informed, specialized in transforming information to knowledge and making this knowledge executable for different audiences. In this approach, documentation is a key factor, not as a static end product but as a dynamic knowledge solution that enables secure decision making and presents information based on the context and information needs of the end user.

At Be Informed, Kees is responsible for the design and development of a knowledge platform for Be Informed employees, partners and clients. He is also the designer and writer of a series of entry-level tutorials about the Be Informed product line.

CONTENT STRATEGY AND DESIGN

[Plain Talk: Get the Marbles Out of Your Message](#) page 93

PLAIN TALK: GET THE MARBLES OUT OF YOUR MESSAGE

Eric Koup

If you've ever spoken to someone who mumbles—or if you happen to be a recovering mumbler—maybe you've heard the expression, "It sounds like you've got marbles in your mouth." (Yes, Mom, you were right.)

It's hard to understand someone who's speaking with a mouth full of marbles. Your written content can be full of metaphorical marbles, which also makes it hard for your reader to understand your message. So, how do you get the marbles out of your message? Plain Talk®.

WHY PLAIN TALK IS IMPORTANT

Vanguard, one of the world's largest investment providers, caters to many types of investors, including direct investors, employer-sponsored plan participants, and financial advisors. But no matter the audience, message, or medium, Vanguard's fundamental communication philosophy is the same: Plain Talk.

But, Plain Talk isn't just a slogan, it's a necessity. Vanguard believes that investing doesn't have to be complicated; investors just need clear information to help them make smart choices. Plain Talk helps makes communications as clear as possible, which, in turn, can help people become more confident investors.

In fact, Vanguard believes so much in the Plain Talk philosophy, that it's part of Vanguard's "Pledge to Clients." This promise to create clear, candid communications is given the same consideration and weight as other promises, like pledging to act ethically and to protect your money and personal information.

HOW TO BE "PLAIN TALK-Y"

A few commonsense—but not necessarily common practice—tips go a long way in helping content follow a Plain Talk style. Here are some:

- **Write to the audience's level.** How would you explain your topic or theory to a "newbie?" (I always imagine that my Mom is reading what I'm writing.)
- **Focus on the benefits, not the features.** What's the WIIFM for the reader, not you or your company.
- **Cut your content ruthlessly.** The fewer words, the easier a document is to read and understand.
- **Use a conversational tone.** Engage your reader with an informal writing style. (Consider using "five-cent" words; writing in the second person and active voice; embracing contractions; and avoiding jargon.) An appropriate sense of humor can work as well.
- **Be factual, sincere, candid, fair, and balanced.** Readers can see through "fluff" content and unproven claims. Both can lead to mistrust—and so can "sins of omission."

MAKE YOUR MESSAGE COME ALIVE

- Look for ways to illustrate your message to make it stick. In March 2012, Vanguard rented a food truck, decorated it in a brand-consistent manner, and sold gourmet coffee in various East Coast cities for 28¢ a cup.

- What's the connection between "underpriced" coffee and mutual funds? Vanguard manages mutual funds at-cost—you pay only what it costs Vanguard to run the funds. The coffee sold in the food truck cost 28¢ to make, so that's what customers paid for a quality cup of coffee. You can bet anyone who made a purchase from the "At-Cost Café" got a better understanding of what it means to invest at-cost.

HOW TO SELL THE IDEA AT YOUR JOB

If you think your company's materials could benefit from a Plain Talk philosophy, it's time to pitch the idea. Here are a few steps to get you started:

- **Make Plain Talk part of the culture.** Add a Plain Talk philosophy to your company's style guide. (Negative incentives can work too; perhaps you could make writers add a quarter to the "donut fund" every time they choose the word "utilize" when "use" will work.)
- **Build creative briefs for creative teams.** These documents not only help the writers craft on-point messages but they also cement the message before the creative part of the project begins. So, if a project changes direction, the team needs to edit the creative brief before work can continue.
- **Research writing and usability experts.** Get some ammunition for why a Plain Talk philosophy is important. I like Anne Wylie and Jakob Nielsen for writing and usability tips, respectively.

RESOURCES

<http://www.atcostcafe.com>

Nielsen, Jakob. "Alertbox" e-newsletter. <http://www.nngroup.com/articles/subscribe/>

Wylie, Anne. "Wylie's Writing Tips" e-newsletter. <http://www.wyliecomm.com/resources/wylies-writing-tips/>

REFERENCES

Sabin, William A. *The Gregg Reference Manual, Twelfth Edition*. (New York, NY: McGraw-Hill), 2010.

AUTHOR CONTACT INFORMATION

Eric Koup
 Writer, Vanguard
 P.O. Box 2600
 Valley Forge, PA 19482
 610.669.9638
eric_koup@vanguard.com

AUTHOR BIOGRAPHY

Eric Koup is a 16-year veteran at Vanguard, where he has worked in multiple communications groups. Eric holds a B.A. in Communications from Loyola University in Maryland and an M.B.A. in Marketing from Villanova University.

Plain Talk is a registered trademark of Vanguard.

EDUCATION AND TRAINING

[Creating Learning That Lasts with Interactive Documents](#) page 97

[The Traditional Executive Summary Versus the Video Game Proposal One-sheet](#) page 101

[User Assistance, Technical Communication, and Learning](#) page 105

CREATING LEARNING THAT LASTS WITH INTERACTIVE DOCUMENTS

Jamie Gillenwater

To add value to technical communication, consider increasing your audience's learning. By applying popular adult learning styles, such as Kolb's Lewinian Experiential Learning Model and Fleming's VARK theory, you can engage your users and decrease how frequently they use reference materials. Many tools are available to create interactive elements within your final output.

There's room for debate over whether technical communicators are responsible for helping their audience learn the technical content we share. However, if we can ultimately save our companies and clients time through effective communication, we are adding value to our careers and profession.

LEARNING THEORIES

To move successfully from producing content to engaging readers for learning, it is important to understand current learning theories. As with all theories, these ideas have not been proven, but trainers apply the methods to ensure consistent results for their students.

Rohrer and Pashler point out that only a small percentage of studies that support learning styles use research that was well designed. However, the researchers also note that trainers' observations throughout the years support the idea that students with poor visual-spatial ability learn more effectively from verbal instruction rather than a diagram.

Additionally, I recommend using a variety of delivery methods to appeal to your students' various strengths. A balanced approach will ensure that your users get the information they need, even if each module does not cater to each individual's learning style.

KOLB'S LEWINIAN EXPERIENTIAL LEARNING MODEL

Kolb (1984) suggests that adult learners must begin the learning process with a "here-and-now experience." His proposed cycle allows users to enter the process with their experience and knowledge, and then test that knowledge against the new information (Figure 1).

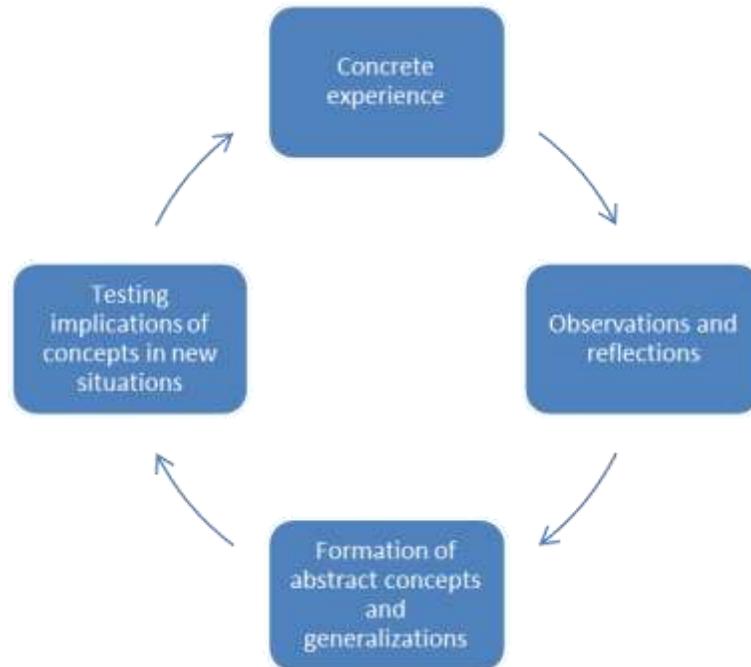


Figure 1 Kolb's Lewinian Experiential Learning Model

Other learning theories support Kolb's model, including Dewey's Model of Learning and Piaget's Model of Learning and Cognitive Development. Each theory supports the idea that learners build their new ideas based on their past experiences, pre-conceived ideas, and testing of new information.

With a combination of verbal, visual, and interactive content, users can complete the experiential learning model in a low-risk environment. If employees are being trained on a piece of fragile equipment, they can experience the processes, see how the pieces fit together, and test potential situations themselves without worrying about damaging the physical equipment.

FLEMING'S VARK THEORY

Neil Fleming's VARK theory is one of the most commonly known adult learning theories. This theory suggests that trainers modify their training to engage students through various teaching methods. These methods include Visual, Auditory, Reading, and Kinesthetic. Fleming shows that each learner favors a particular teaching method. Visual learners learn the most by seeing a demonstration or illustration. Kinesthetic learners learn by doing. Creating interactive content that demonstrates how to accomplish a task, followed by the opportunity for the user to test the system, will meet the needs of multiple learners.

Fleming has developed his research into a business, which focuses on gathering additional data about learning styles, creating learning style profiles for clients, and leading workshops about how to personalize training programs based on these profiles.

INTERACTIVE TUTORIAL SUCCESS

To determine the effects of interaction, Bolliger and Supanakorn worked with a group of learners who were given interactive tutorials to supplement regular course materials. Overall, the students agreed "the tutorials were useful in learning the software programs" required in their course.

Based on a VARK questionnaire developed by Fleming, the majority of the students were classified as auditory learners, with reading and kinesthetic learners coming in second. When asked why the participants appreciated the tutorials the reasons varied, but included:

- the ability to review information, which helped them understand the course materials,
- the self-explanatory nature of the tutorials,
- the actual step-by-step guide on how to complete tasks,
- the usefulness and helpfulness of the information in the tutorials, and
- online accessibility from any computer.

As this research demonstrates, interactive and engaging instructions encourage your audience to learn your material rather than merely refer to your information.

APPLYING INTERACTION

Opportunities to learn through interaction can help readers more readily understand complex content. Many popular technical communication tools offer the ability to include interaction. Here is a partial list of software products that include such interaction:

Training software

- Adobe Captivate
- Articulate Storyline
- Lectora Inspire

Documentation software

- Adobe FrameMaker
- Adobe InDesign
- Quark App Studio

3D engineering software

- AutoDesk Inventor Publisher
- 3DVIA SolidWorks Composer
- QuadriSpace Document3D

As with every major design decision, it is important to consider your users' environment, where they might access the information, what will help them understand your content, and how interactive content fits into your current documentation processes.

RESOURCES

Fleming, Neil. *Teaching and learning styles: VARK strategies* (2nd ed.) (Christchurch, N.Z. Fleming), 2005.

REFERENCES

Bolliger, Doris U. and Supawan Supanakorn. "Learning styles and student perceptions of the use of interactive online tutorials." *British Journal of Educational Technology* 42.3 (May 2011): 470-481. <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8535.2009.01037.x/full>.

Rohrer, Doug and Harold Pashler. "Learning styles: Where's the evidence?" *Medical Education* 46.7 (2012): 634-635. <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2923.2012.04273.x/full>.

Kolb, D.A. *Experiential Learning: Experience as the Source of Learning and Development* (Englewood Cliffs, NJ: Prentice Hall), 1984.

Phillips, Joseph. *All-In-One Certified Technical Trainer Exam Guide* (New York, NY: The McGraw-Hill Companies), 2012.

AUTHOR CONTACT INFORMATION

Jamie Gillenwater
Global Technical Documentation Specialist
Capsugel
535 N. Emerald Rd.
Greenwood, SC 29646
864.980.2521

AUTHOR BIOGRAPHY

As a technical communicator, Jamie Gillenwater develops information to push readers beyond reading to learning. She works as the Global Technical Documentation Specialist for Capsugel. She has managed information security projects, cost-saving manufacturing projects, and training sessions.

Her professional interests include training, effective communication, and unique delivery solutions. Jamie serves as the president for Connect Young Professionals, secretary for STC Carolina, and an active member for the Greenwood Chamber of Commerce Women's Leadership Council.

THE TRADITIONAL EXECUTIVE SUMMARY VERSUS THE VIDEO GAME PROPOSAL ONE-SHEET

Ann Jennings, PhD

The video game proposal one-sheet is similar to the traditional executive summary but with some important differences. The one-sheet is a marketing document and a condensed explanation of a highly creative endeavor executable only by a team of disparate specialists.

As exciting as your favorite video game may be to play, that game was created over a period of years through the tedious work of game designers (who may also be the writers of the game story), artists, programmers, production personnel, and other specialists. The game story and concept may have been sold in advance to a publishing company—or not, if the game is a simple one meant to be played on a mobile device. In either case, at some point in the game's development, a one-sheet (a specialized executive summary) may have been written.

In the broadest sense, a one-sheet serves a purpose similar to that of a traditional executive summary: to present a condensed overview of a project to decision makers or other interested people. But as I learned while preparing to teach a course in video game story writing, the one-sheet is a unique type of document particular to the demands of the video game industry, and its use is encouraged by industry practitioners.

Note: Although the one-sheet can also be referred to as an executive summary, for the purposes of this report, I will refer to the video game-related document as a one-sheet.

To determine the similarities and differences between the video game proposal one-sheet and the traditional executive summary, I reviewed several important game design books and several leading technical communication and business writing text books. The results are presented below, beginning with the better known traditional executive summary and proceeding to a discussion of the lesser known one-sheet. By reading this material, technical communication practitioners may discover an intriguing world of documents that work in the service of a special endeavor, the creation of a video game.

TRAITS OF THE TRADITIONAL EXECUTIVE SUMMARY

The traditional executive summary, as it is taught nationwide in university business and technical report writing courses, displays several basic traits.

- Independence: The document is meant to stand alone (Alred, Brusaw, and Oliu, 2012, p. 193; Johnson-Sheehan, 2012, p. 302).
- Efficiency: The document is a condensed version of a fully developed report (Alred, Brusaw, and Oliu, 2012, p. 193; Gerson and Gerson, 2014, p. 450; Johnson-Sheehan, 2012, p. 301; Markel, 2012, p. 528).
- Length: The length can vary from one page to as much as 5-10% of a fully developed report, depending upon the length of that report (Alred, Brusaw, and Oliu, 2012, p. 193; Gerson and Gerson, 2014, p. 451; Markel, 2012, p. 528).
- Contents/Specificity: The document paraphrases important points from a fully developed report (Johnson-Sheehan, 2012, p. 301).

- Categories of information covered: The document establishes the existence of a problem and mentions methods of research, findings, conclusions, and recommendations for solving the problem (Alred, Brusaw, and Oliu, 2012, p. 193; Gerson and Gerson, 2014, p. 451; Markel, 2012, p. 528).
- Additional information: The document may establish the qualifications of the writers to solve the problem and may offer a “total project budget” and “a project management plan and timetable” (Gerson and Gerson, 2014, p. 451).

TRAITS OF THE VIDEO GAME PROPOSAL ONE-SHEET

Video game designers/writers characterize the one-sheet in several ways.

- Independence: The one-sheet is used as a stand-alone piece to distribute to those involved in planning, producing, and marketing a game (Rogers, p. 431; Bates, p. 18). It also serves as the first portion of a longer document called the “high-level design document” or “concept document” (Dille and Platten, p. 101) or “the ten-page design document” (Rogers, p. 435).
- Efficiency: As the first section of a fully developed design document, the one-sheet does not intentionally repeat information located later in that document.
- Length: The length can vary from one to five pages (Dille and Platten, p. 101).
- Contents/Specificity: The one-sheet covers points particular to the creation, production, and publishing of video games (Dille and Platten, pp. 101-104; Rogers, p. 431).
- Categories of information covered: The one-sheet identifies key facts about the video game that is proposed for development. These facts are the game title, game genre, synopsis, game category, platforms on which the game will be played, any existing property/license on which the game is based—or the game’s status as “original intellectual property,” play mechanics, technology to be used to build the game, target audience, unique selling points, marketing summary (Dille and Platten, pp. 102-104); and existing games that the new games resembles (Rogers, p. 432).
- Additional information: The one-sheet may also provide a proposed time frame and costs and a statement of potential risks and ways to reduce them (Bates, pp. 19-20).

COMPARISON: THE EXECUTIVE SUMMARY AND THE ONE-SHEET

The two previous bulleted lists establish some clear similarities and differences between the traditional executive summary and the video game proposal one-sheet. I have noticed additional differences while reading scores of traditional executive summaries and a lesser number of one-sheets written by professional video game writers and by students.

The purpose: The traditional executive summary chronicles the orderly progression from the identification of a business or technical problem, to the selection of methods by which to obtain information about the problem and potential solutions, to the results of the research, to conclusions and recommendations. The one-sheet is a recommendation: finance the development of my game because I am showing you an exciting story and gameplay that a special audience will want to buy.

The voice: While both types of documents rely primarily on businesslike wording, the one-sheet also accommodates the imaginative use of words as writers describe their vision of a game that does not yet exist. The writers try to make their readers imagine characters, a story, motion, atmosphere, and a unique landscape/seascape/space scape.

The subject matter: Whereas the traditional executive summary may focus on a business proposal or a process that is either proposed or has already occurred, the one-sheet concentrates on the future. The one-sheet emphasizes the potential for a new game that will contain exciting gameplay, attract players, and bring financial success. The one-sheet describes enough of the story and characters that artists, audio experts, programmers, and a sales team can visualize the game and so that decision-makers can authorize (or deny) funding for the development of the game.

The importance of action: Unlike many traditional executive summaries, the video game proposal one-sheet focuses on action—gameplay—and the tools, technology, and devices that enable gameplay.

In summary, the video game proposal one-sheet is similar enough to the traditional executive summary to seem related, but different enough to stimulate the imagination of technical communicators and offer the potential of learning a new way to express technical information.

REFERENCES

Alred, Gerald J., Charles T. Brusaw, and Walter E. Oliu. The Business Writer's Handbook (10th ed.) (Boston, MA: Bedford/St. Martin's), 2012.

Bates, Bob. Game Design: The Art and Business of Creating Games (Roseville, CA: Prima Publishing), 2001.

Dille, Flint, and John Zurr Platten. The Ultimate Guide to Video Game Writing and Design (New York, NY: Lone Eagle Publishing Co.), 2007.

Gerson, Sharon J., and Steven M. Gerson. Technical Communication: Process and Product (8th ed.) (Boston, MA: Pearson), 2014.

Johnson-Sheehan, Richard. Technical Communication Today (4th ed.) (Boston, MA: Pearson), 2012.

Markel, Mike. Technical Communication (10th ed.) (Boston, MA: Bedford/St. Martin's), 2012.

Rogers, Scott. Level Up: The Guide to Great Video Game Design (Chichester, West Sussex, UK: John Wiley & Sons, Ltd.), 2010.

AUTHOR CONTACT INFORMATION

Ann Jennings
Department of English
University of Houston Downtown
One Main St., Suite S-1045
Houston TX 77002-1001
713.221.8453

AUTHOR BIOGRAPHY

Ann Jennings, PhD, is a professor of English who teaches technical and professional writing courses to undergraduate and graduate students at the University of Houston Downtown. Her course in video game story writing covers the basic creative and business documents common to the development of video games. In 2009, she was awarded STC's Jay R. Gould Award for Excellence in Teaching Technical Communication.

USER ASSISTANCE, TECHNICAL COMMUNICATION, AND LEARNING (INSTRUCTIONAL DESIGN)

Nicky Bleiel, Saul Carliner, Kevin Siegel, and Matt Sullivan

User assistance is one form of informal learning. In this panel, we'll invite the audience to join us in a conversation about how users find what they need, what technical communicators need to provide, and how we can produce more effective user assistance.

A deeper understanding of instructional design principles can help technical communicators create more effective user assistance more efficiently. The following sections explain how technical communication and training are similar, their differences, and how they work together to guide and teach.

DEFINITION OF INFORMAL LEARNING

Informal learning, in the context of the workplace, refers to a situation in which workers control some or all of the following aspects of the process of acquiring knowledge and skill (Wihak, et al, 2008):

- Process (who controls and assesses the learning process?)
- Location (was it intended for learning or not?)
- Purpose (is learning is a primary or secondary goal)
- Content (is the content technical or practical and related to an everyday skill—considered more informal—or abstract—considered more formal?—or technical, or related to a practical, everyday skill)
- Consciousness (were workers aware that learning occurred)

CHANGE YOUR TERMINOLOGY: TRANSLATING TECHNICAL COMMUNICATION-ESE AND INSTRUCTIONAL DESIGN-ESE

(From Carliner, S. (2012b.) Used with permission.

Generic Term	• Audience
Term Used in Tech Comm	• User
Term Used in Training	• Learner

Generic Term	<ul style="list-style-type: none"> • Statement of Purpose
Term Used in Tech Comm	<ul style="list-style-type: none"> • Task
Term Used in Training	<ul style="list-style-type: none"> • Objective
Generic Term	<ul style="list-style-type: none"> • Prewriting Research
Term Used in Tech Comm	<ul style="list-style-type: none"> • Audience and Task Analysis
Term Used in Training	<ul style="list-style-type: none"> • Needs Assessment • Needs Analysis
Generic Term	<ul style="list-style-type: none"> • Design Activities
Term Used in Tech Comm	<ul style="list-style-type: none"> • Content Plan • Outline
Term Used in Training	<ul style="list-style-type: none"> • Instructional Design • Instructional Strategy
Generic Term	<ul style="list-style-type: none"> • Writing
Term Used in Tech Comm	<ul style="list-style-type: none"> • Writing
Term Used in Training	<ul style="list-style-type: none"> • Message Design

Generic Term	<ul style="list-style-type: none"> • Reviews
Term Used in Tech Comm	<ul style="list-style-type: none"> • Editing • Peer Review • Technical Review
Term Used in Training	<ul style="list-style-type: none"> • General: Formative Evaluation • Instructional Review • Expert Review
Generic Term	<ul style="list-style-type: none"> • Trial Run
Term Used in Tech Comm	<ul style="list-style-type: none"> • Usability Test • Field Test • Heuristic Analysis
Term Used in Training	<ul style="list-style-type: none"> • Pilot

Name of the Discipline Used by Professionals	
Technical Communication (preferred) Technical Writing Instructional Design Language (English) for Specific Purposes Information Design Content Development Content Strategy	(none preferred) Education Learning and Development Learning and Performance Performance Improvement Training Training and Development

Name of the Discipline Used by the Academic Community	
Technical Communication (preferred) Also: Applied Linguistics Human-Centered Engineering Information Design Language (English) for Specific Purposes Technical communication Technical Writing	Human Resource Development (preferred) Also: Adult Education Educational (Instructional) Technology Training (a division of Human Resources research within the field of Management, or a division of Industrial and Organizational Psychology within the field of Psychology) Workplace Learning

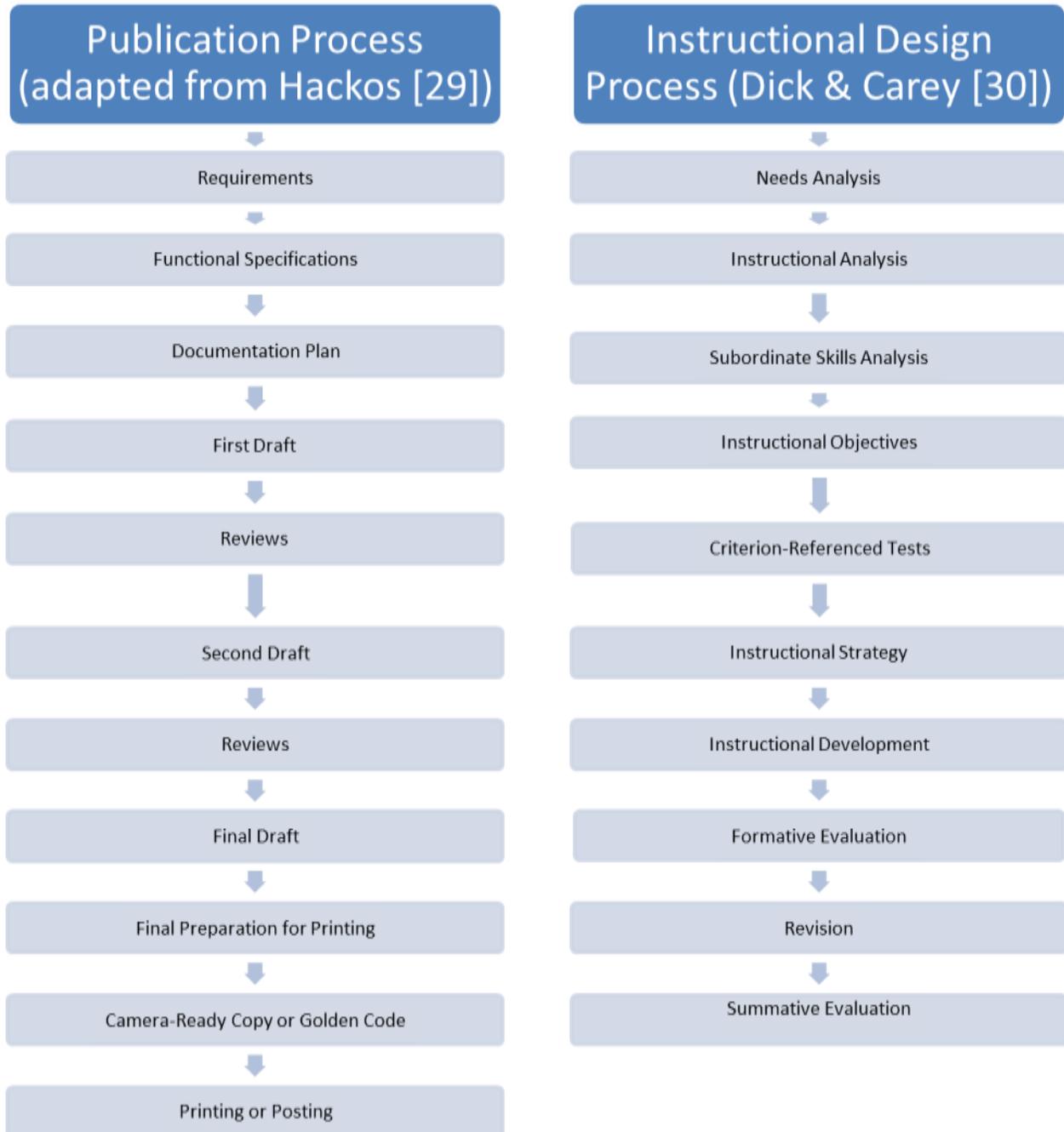
CHANGE YOUR PERSPECTIVE: DIFFERING INTELLECTUAL FOUNDATIONS OF TECHNICAL COMMUNICATION AND INSTRUCTIONAL DESIGN

(From Carliner, S. (2012b.) Used with permission.

	Technical Communication	Both	Training
Roots	Rhetoric and Composition	Cognitive Psychology	Education Andragogy
Research Methods	Primary: Critical		Primary: Empirical

ADJUST YOUR PROCESSES — A COMPARISON OF THE PUBLICATIONS, AND ISD PROCESSES

(From Carliner, S. (2012b.) Used with permission.



WHILE SIMILAR, THE JOBS FUNDAMENTALLY DIFFER: THE DISTINCTIONS AND OVERLAPS AMONG THE WORK AND WORK PROCESSES OF TECHNICAL COMMUNICATORS AND TRAINERS

(From Carliner, S. (2012b.) Used with permission.

WORK

Goals: Technical Communication

Information that users are not necessarily expected to retain.

Goals: Training and Instructional Design

Develop skills that workers must master to successfully perform their jobs.

Job Responsibilities: Technical Communication

Majority write as their primary job responsibility

Other responsibilities:

- Editing
- Managing Technical Communicators

Job Responsibilities: Training and Instructional Design

Majority work as instructors.

Other responsibilities:

- Instructional design
- Administrative staff
- Senior planners
- Managing Trainers

Competencies: Technical Communication

- Project planning
- Project analysis
- Solution design
- Organizational design
- Written communication
- Visual communication
- Content development

- Content management
- Final production

Competencies: Training and Instructional Design

- Analysis of the performance underlying a request for training
- Design of an appropriate *intervention*,
- Classroom facilitation,
- Support for the transfer of training to the job
- Evaluation of training

Certifications Available: Technical Communication

Certified Technical Communicator

Certifications Available: Training and Instructional Design

For the complete range of competencies in Training and Development:

- Certified Professional in Learning and Performance (USA)
- Certified Performance Technologist (USA)
- Certified Training and Development Professional (Canada)
- Learning and Development Professional (UK)

Instructor-only:

- Certified Training Practitioner (Canada)
- Certified Technical Training (USA)

WORK PROCESSES

Analysis: Technical Communication

Requirements

Analysis: Training and Instructional Design

- Needs Analysis
- Instructional Analysis
- Subordinate Skills Analysis
- Instructional Objectives
- Criterion-Referenced Tests

Design: Technical Communication

- Functional Specifications
- Documentation Plan

Design: Training and Instructional Design
Instructional Strategy**Development: Technical Communication**

- First Draft
- Review
- Second Draft
- Review

Development: Training and Instructional Design

- Instructional Development
- Formative Evaluation
- Revision

Production: Technical Communication

- Final Draft
- Final Preparation for Printing
- Printing or Posting
- Camera-Ready Copy or Golden Code

Production: Training and Instructional Design

Not mentioned

Implementation: Technical Communication

Not mentioned

Implementation: Training and Instructional Design

- Administering courses (such as enrollment and related record keeping, classroom setup)
- Teaching individual class sessions

Evaluation: Technical Communication

Not mentioned

Evaluation: Training and Instructional Design

Summative evaluation (Kirkpatrick methodology)

RESOURCES AND REFERENCES

_____. (2010.) *Edutainment*. Wikipedia. Viewed at <http://en.wikipedia.org/wiki/Edutainment>. Visited January 3, 2011.

Aguinis, H. & Kraiger, K. (2009) *Benefits of training and development for individuals and teams, organizations, and society*. *Annual Review of Psychology*, 60, 451–474.

Aral, S. (2010.) *Content is king: Combining network and text analysis*. Presented at the 2010 Meeting of the Academy of Management, Montreal, QC, August 9, 2010.

Bennett, S., Maton, K., & Kervin, L. (2008.) *The 'digital natives' debate: A critical review of the evidence*. *British Journal of Educational Technology*, 39(5), 775-786.

Bloom, J. & Powell, E. A. (1984.) *Museums for a New Century: A Report of the Commission on Museums for a New Century*. Washington, DC. American Association of Museums.

Broad, M. L. & Newstrom, J. W. (2001.) *Transfer of Training: Action-Packed Strategies to Ensure High Payoff from Training Investments*. New York, NY: Perseus Books.

Burk, L. A. & Hutchins, H. M. (2007.) *Training transfer: An integrative literature review*. *Human Resource Development Review*, 6(3), 263-296.

Cappelli, P. (2008.) *Talent on Demand: Managing Talent in an Age of Uncertainty*. Cambridge, MA: Harvard Business Press.

Carliner, S. (2012a.) *Informal Learning Basics*. Alexandria, VA: ASTD Press.

Carliner, S. (2012b.) *Different approaches to similar challenges. An analysis of the occupational cultures of the disciplines of Technical Communication and Training*. *IEEE Transactions on Professional Communication*, 55(2), 160-174

Carliner, S. (2002.) *Choices and challenges*. *Technical Communication*, 49(4).

Carliner, S. (2000.) *Eight Things That Training and Performance Improvement Professionals Must Know about Knowledge Management*. Published online by Minneapolis, MN: VNU Business Media.

Carliner, S. (1995.) *Every Object Tells a Story: A Grounded Model of Design for Object-Based Learning in Museums*. Unpublished doctoral dissertation. Atlanta, GA: Georgia State University.

Carter, S. D. (2005.) *The growth of supply and demand of occupational-based training and certification in the United States, 1990–2003*. *Human Resource Development Quarterly*, 16(1), 33-54.

Carliner, S. & Bernard, C. (2011a.) *An integrative review of literature on the perceptions of HRD*. *Proceedings of the 2011 Academy of Human Resource Development Research Conference in the Americas*. St Paul, MN: Academy of Human Resource Development.

Carliner, S. & Bernard, C. (2011b.) *A qualitative study of the perceptions of workplace learning professionals*. 2011 Joint Canadian Society for the Study of Adult Education and the Adult Education Research Conferences. Toronto, ON: June 9, 2011.

Carliner, Legassie, Belding, MacDonald, Ribeiro, Johnston, MacDonald, & Hein. (2009). *How research moves into practice: A preliminary study of what training professionals read, hear, and perceive*. *Canadian Journal of Learning and Technology*, 35(1). Viewed at <http://www.cjlt.ca/index.php/cjlt/article/view/510/240>. Visited December 26, 2009.

Clark, R. (2010.) *Closing keynote*. 2010 Association for Educational Communications and Technology International Convention. Anaheim, CA: October 29, 2010.

Colley, H., Hodkinson, P., & Malcolm, J. (2003). *Informality and formality in learning: a report for the Learning and Skills Research Centre*. London: Learning and Skills Research Centre.

Conference Board. 2009.

Cross, J. (2007.) *Informal Learning: Rediscovering the Natural Pathways that Inspire Innovation and Performance*. San Francisco, CA: Pfeiffer.

Cseh, M., Watkins, K. E., and Marsick, V. J. (1999.) *Re-conceptualizing Marsick and Watkins' model of informal and incidental learning in the workplace*. In Kuchinke, K. P.(ed.), *Proceedings, Academy of Human Resource Development Conference, Volume I*. Baton Rouge, LA: Academy of Human Resource Development, 349–356.

Dale, M. & Bell, J. (1999.) *Research Brief No. 134: Informal Learning in the Workplace*. UK Department for Education and Employment.

Dick, W, Carey, L. & Carey, J. (2008.) *The Systematic Design of Instruction*, (7th ed.). Burlington, MA: Addison-Wesley.

Driscoll, M., & Carliner, S. (2005.) *Advanced Web-Based Training: Adapting Real World Strategies in Your Online Learning* San Francisco: Pfeiffer.

Downing, J. (2006.) *Using customer contact center technicians to measure the effectiveness of online help systems*. *Technical Communication*, 54(2), 201-209.

Ellinger, A. D. (2005), *Contextual factors influencing informal learning in a workplace setting: The case of "reinventing itself company"*. *Human Resource Development Quarterly*, 16(3), 389–415.

Eraut, M. (2000.) *Non-formal learning and tacit knowledge in professional work*. *British Journal of Educational Psychology*, 70(1), 113-136.

Falk, J. H. & Dierking, L. (2000) *Learning from Museums: Visitor Experiences and the Making of Meaning*. Walnut Creek, CA: AltaMira Press.

Gery, G. (1991.) *Electronic Performance Support Systems*. Tolland, MA: Gery Performance Press, Inc.

Gibbons, M. (2002.) *The Self-Directed Learning Handbook*. San Francisco, CA: Jossey-Bass.

Gibbons, M. (2008.) *What is self-directed learning*. Viewed at www.selfdirectedlearning.com. Visited December 11, 2010.

Gilbert, T. (1978.) *Human Competence: Engineering Worthy Performance*. New York: McGraw-Hill.

Hager, P. (2004). *Lifelong learning in the workplace? challenges and issues*. *Journal of Workplace Learning*, 16(1-2), 22-32.

- Kay, R. (2007.) *A formative analysis of resources used to learn software*. *Canadian Journal of Learning and Technology*, 33(1). Viewed at <http://www.cjlt.ca/index.php/cjlt/article/view/20/18>. Visited March 19, 2011.
- Knowles, M., Swanson, R. A. & Holton, E. F. (2011.) *The Adult Learner (7th ed.)* Burlington, MA: Butterworth-Heinemann.
- Kristof, N. D., (2009.) *Learning how to think*. *New York Times*, March 26, 2009. Viewed at <http://www.nytimes.com/2009/03/26/opinion/26Kristof.html?ref=opinion>. Visited March 26, 2009.
- Laiken, M., Edge, K., Friedman, S., & West, K. (2008.) *Formalizing the Informal: From Informal to Organizational Learning in the Post-Industrial Workplace*. New York, NY: Springer.
- Lalonde, D. (2010.) *Sondage AQIII 2010 : La formation continue negligee*. *Direction Informatique*, October 12, 2010. Viewed at <http://www.linkedin.com/news?viewArticle=&articleID=223052599&qid=3254662&type=member&item=32053081&articleURL=http://www.directioninformatique.com/DI/client/fr/DirectionInformatique/Nouvelles.asp%3Fid%3D59624%26cid%3D79&urlhash=DQ62>. Visited November 26, 2010.
- Leigh, D. & Watkins, R. (2005) *E-Learner success: Validating a self-assessment of learner readiness for online training*. In Carliner, S. & Sugrue, B. (eds.) *Proceedings of the 2005 ASTD Research-to-Practice Conference-within-a-Conference*. Alexandria, VA: ASTD Press.
- Lim, D. H. & Morris, M. L. (2006.) *Influence of trainee characteristics, instructional satisfaction, and organizational climate on perceived learning and training transfer*. *Human Resource Development Quarterly*, 17(1), 85-115.
- Livingstone, D. (2010.) *The relationship between workers' practical knowledge and their job requirements: Findings of the 1998, 2004 and 2010 National Surveys of Work and Lifelong Learning*. Presented at *Canadian Society for Training and Development Annual Conference and Trade Show*, Toronto, ON: November 17, 2010.
- Mager, R. (1997.) *Preparing Instructional Objectives*. Atlanta, GA: Center for Effective Performance
- Marsick, V. J. (2009.) *Toward a unifying framework to support informal learning theory, research and practice*. *Journal of Workplace Learning*, 21(4), 2009, 265-275.
- Marsick, V. J. & Watkins, K.E. (2001.) Chapter 3. *Informal and incidental learning*. In Merriam, S.B. (ed.) (2001.) *The New Update on Adult Learning Theory*. San Francisco, CA: Jossey-Bass.
- Marsick, V. J. & Watkins, K. (2011.) *Pursuing Research in Organizations that is Useful to Practice*. *Academy of Human Resource Development Research Conference in the Americas*. Schaumburg, IL, February 24, 2011.
- Marsick, V. J. & Watkins, K. (1990.) *Informal and Incidental Learning in the Workplace*. London, UK: Routledge.
- Merriam, S.B. (2001.) *Editor's notes*. In Merriam, S.B. (ed.) (2001.) *The New iUpdate on Adult Learning Theory*. San Francisco, CA: Jossey-Bass, 1-2.
- Millar, R. *Plenary: Informal learning in the workplace*. *Third annual symposium of the Work and Learning Knowledge Centre*. *Work and Learning Knowledge Centre of the Canadian Council on Learning*. Ottawa, ONT. May 15, 2008.
- Ogata, H. & Yanno, Y. (n.d.) *Context-aware support for computer-supported ubiquitous learning*. Viewed at <http://www-yano.is.tokushima-u.ac.jp/ogata/clue/WMTE-03-1-50.pdf>. Visited January 10, 2010.

Oregon Sea Grant (2004.) *Developing the art and science of free-choice learning*. Viewed at http://seagrant.oregonstate.edu/makingadifference/stories/Free_ChoiceSoA.pdf. Visited January 3, 2011.

Schulz, M., & Rosznagel, C. S. (2010). *Informal workplace learning: An exploration of age differences in learning competence*. *Learning and Instruction*, 20(5), 383-399.

Selwyn, N. (2010.) *The educational significance of social media*. ED-MEDIA 2010. Toronto, ON: Association for the Advancement of Computers in Education, July 2, 2010.

Skule, S. (2004). *Learning conditions at work: a framework to understand and assess informal learning in the workplace*. *International Journal of Training and Development*, 8(1), 8–20.

Wedman, J. & Tessmer, M. (1993.) *Instructional designer's decisions and priorities: a survey of design practice*. *Performance improvement quarterly*. 6(2), 43-57.

Werquin, P. (2007), *Moving Mountains: will qualifications systems promote lifelong learning?*. *European Journal of Education*, 42, 459–484.

Westbrook, T.S., & Veale, J.R. (2001). *Work-related learning as a core value: An Iowa perspective*. *Human Resource Development Quarterly*. 12(3). 301-317.

Wihak, C., Hall, G. Bratton, J., Warkentin, L., Wihak, L. & MacPherson, S. (2008.) *Unpublished report. Work-Related Informal Learning: Research and Practice in the Canadian Context*. Ottawa, ON: Work and Learning Knowledge Centre of the Canadian Centre for Learning.

AUTHOR CONTACT INFORMATION

Nicky Bleiel
Lead Information Developer
ComponentOne Doc-To-Help
201 South Highland Avenue, Third Floor
Pittsburgh, PA 15206
412.681.4343
nickyb@componentone.com
www.nickybleiel.com

Saul Carliner
eLearning Fellow and Associate Professor
Concordia University
Department of Education
1455 boulevard de Maisonneuve Ouest
Montreal, Quebec H3G 1M8
Canada
514.848.2424 x2038
saulcarliner@hotmail.com
www.saulcarliner.com

Kevin Siegel
President
IconLogic, Inc.
www.iconlogic.com
410.956.4949
Blog: <http://iconlogic.blogs.com/>

<http://www.facebook.com/IconLogicInc>
www.twitter.com/kevin_siegel

Matt R. Sullivan
Adobe Certified Instructor/Expert
28371 Camino Dimora
San Juan Capistrano, CA 92675
714 798-7596
matt@mattersullivan.com
www.mattersullivan.com

AUTHOR BIOGRAPHIES

Nicky Bleiel (@nickybleiel) is the Lead Information Developer for Doc-To-Help. She has 18 years of experience in technical communication; writing and designing information for software products in the documentation, media, industrial automation, simulation, and pharmaceutical industries. She is the Vice President of the Society for Technical Communication. See nickybleiel.com for a complete list of her articles and presentations.

Saul Carliner is the e-Learning Fellow and an associate professor at Concordia University in Montreal. His research and teaching focus on the design of content for the workplace, the management of groups that produce the content, and the transfer of research to practice. His publications include *Physical, Cognitive, and Affective: A Three-Part Model of Information Design* (winner of a Best of Show award in the Frank R. Smith competition), the classic text *Techniques for Technical Communicators* (with Carol Barnum), and the best-selling *Training Design Basics*. He is a fellow and past international president of STC. For more information, visit saularliner.com.

Kevin Siegel, founder and president of IconLogic, Inc., has been teaching adult learners since 1992. As a certified technical trainer, technical writer, help author, instructional designer and eLearning developer, Kevin continues to sharpen his skills as a distiller and presenter of “need-to-know” information. Evolving from a 5-year military tour in photojournalism, Kevin’s “skills and drills” approach accelerates the learning curve for thousands of customers and fans alike. Kevin has written hundreds of workbooks teaching readers how to use software from Adobe, Quark, Techsmith, and Microsoft. His recent top-selling editions include how-to workbooks on Adobe Captivate, Adobe RoboHelp and TechSmith’s Camtasia Studio 8. A frequent speaker at conferences and trade shows, Kevin was recognized by Adobe Systems, Inc. as one of their Top Five Most Highly Rated Adobe Certified Instructors, Worldwide. Kevin has recently delivered presentations at the International Legal Technology Association (ILTA) Annual Conference, WritersUA, the eLearning Developers Conference (DevCon), the Intelligent Content Conference and the Adobe Learning Summit.

Matt Sullivan (@mattersullivan) is an Adobe Certified Trainer (ACI/ACE) in 10 different Adobe applications, including the entire Technical Communication Suite. He recently co-authored *Publishing Fundamentals: Unstructured FrameMaker 11* with Sarah O’Keefe. He has presented at STC Summit for the past few years, as well as WritersU/A, LavaCon, Intelligent Content, The HOW Design Conference, and DocTrain West. He owned and operated GRAFIX Training and Consulting, an Adobe Certified Training Provider, for 13 years before managing training operations for roundpeg, inc. Since then Matt has specialized in online training and services. He creates blended online solutions, and helps clients create dynamic, responsive electronic output from traditional print documentation. See mattersullivan.com for blog entries and other helpful content.

PEOPLE, PROJECTS, AND BUSINESS MANAGEMENT

[A Manager's Guide to Acquiring Technology](#) page 121

[Bending Without Breaking: Info Dev Flexibility in Agile](#) page 123

[Our Processes Are a Mess!](#) page 131

[Supporting Customers: Onsite and Online](#) page 135

A MANAGER'S GUIDE TO ACQUIRING TECHNOLOGY

Richard Hamilton

Acquiring technology is one of those jobs that happens just infrequently enough that the hard-earned lessons from the last technology acquisition have been forgotten when the next one comes around. It shouldn't be that way. It's much better to keep an eye on technology needs on a regular basis, matching current customer needs with the available technology.

In 2011, the STC Summit management progression contained a session titled, *Acquiring Technology: First Steps*. This session examined three broad steps that all organizations need to traverse before they acquire technology: defining the problem, defining the requirements, and writing the specification. The session this year focuses on the ongoing question of defining your needs and keeping them, and the technology that supports those needs, up to date.

ASSESSING YOUR NEEDS

Normally, the best way to evaluate tools needs is to first look at customer needs separately from the technology you are currently using or are planning to use. This is usually excellent advice. We often let new technology obscure our needs or even lead us to misunderstand them.

This happens because of the “shiny toy” syndrome. That is, we see the shiny toy and decide that we need it, without considering how we will use it. This syndrome sells a lot of Porsches and Corvettes, but nearly always the purchaser of a Porsche or Corvette doesn't need a sports car. He (or more rarely she) just “wants” the “shiny toy.”

Therefore, separating the need from the means to satisfy that need usually yields a more useful outcome. If you take your aging mother for a drive every weekend, analyzing your needs will probably lead you to a four-door sedan she can get in and out of without a crane. Understanding your needs may not lead to the most exciting result, but it is more likely to lead to the most sensible one.

However, in the technical communication world, technology is changing so quickly and radically that we cannot ignore it, even in the early stages. Our customers expect our content to be available through the latest venues and on their brand-new mobile devices. They expect you to interact with them using social media, and every year they expect you to deal with the latest trend.

TECHNOLOGY ACQUISITION AS A CONTINUOUS ACTIVITY

Smart managers treat planning as a continuous process, not a write it and forget it activity. The plan is always alive and open for change as circumstances change. Technology acquisition is no different. The technical communication landscape moves so rapidly that if you don't keep constant track of what is happening, you will fall behind.

In the last five years, the hot topic at each STC Summit has changed. This year it's content strategy, next year it will be something else. While you may not need (or want) to jump on every bandwagon, each trend has at least a core of useful ideas and often much more. Even if social media – to randomly pick a hot topic from a year or so ago – has been replaced by content strategy as this year's idea, the core concepts hold true. If you

missed the “year of social media,” you missed important information about how to interact with your customers.

Therefore, technology assessment needs to become part of your regular job. Fortunately that is pretty easy to do. After all, we do like our shiny toys, so looking at the latest shiny toys and seeing how they apply to our current environment is a valid exercise.

The trick is that you must look past specific tools and consider core capabilities. To do that, you need to understand the way applications work and what unique capabilities they offer. To carry the car analogy a bit further, if you hear about the Tesla electric car, focus on the game-changing aspect of the technology, not the very cool wrapper. If you take that approach, you may find that a Chevy Volt, Nissan Leaf, or something entirely different ends up fitting your needs better. Or the Tesla could still be it, or your current Honda.

A good process doesn't demand change. Instead, it provides what pilots call “situational awareness.” That is, the constant knowledge of where you are and where you are headed. If you know what the latest technology can do, you can then decide if you need that technology. And as a side benefit, you'll be ready when your manager asks you about some new shiny toy he or she saw at a trade show.

RESOURCES

Gentle, Anne. Conversation and Community: The Social Web for Documentation, 2nd Edition. (Laguna Hills, CA: XML Press), 2012.

Hamilton, Richard L. Managing Writers: A Real World Guide to Managing Technical Documentation. (Laguna Hills, CA: XML Press), 2009.

AUTHOR CONTACT INFORMATION

Richard L. Hamilton
Publisher
XML Press
<http://xmlpress.net>
24310 Moulton Pkwy, Suite O-175
Laguna Hills, CA 92637
970.231.3624

AUTHOR BIOGRAPHY

Richard L. Hamilton is founder of XML Press, which is dedicated to producing high quality, practical publications for technical communicators, managers, marketers, content strategists, and the engineers who support their work. He trained as a musician and computer programmer, but after a few years plying those diverse trades, he became a documentation manager, working at several companies, including Unix System Labs and Hewlett-Packard. He founded XML Press in 2008.

Richard is the author of *Managing Writers, A Real World Guide to Managing Technical Documentation* (Hamilton, 2009).

BENDING WITHOUT BREAKING: INFO DEV FLEXIBILITY IN AGILE

Alyssa Fox

Agile development is a popular methodology used by software companies. When working in an agile environment, technical writers face several challenges in planning, workload balancing, and execution.

Flexibility is key in an agile environment, especially if you're working on multiple scrum teams. That flexibility improves communication, streamlines planning, and contributes to an overall "whole team approach" to quality.

INTRODUCTION

Businesses today are cutting back and doing more with less. People have more to do and less time and budget with which to do it. In software companies, one way this can manifest itself is in resources moving among multiple projects. When NetIQ began using an agile development process, we searched far and wide for information about how to work using that process when working on multiple teams. What we found was that most of the writers out there working with agile development were dedicated to one product team. As technical writers, we have unique challenges in agile development.

At NetIQ, the members of our Information Development team move from product team to product team. We have lead writers on a few product teams that generally don't move around, but for the most part, our team members work on at least two projects simultaneously. While the flexibility of our team members can be beneficial, it can also be challenging to plan for and execute work on multiple teams. This paper describes how technical writers can work effectively on multiple scrum teams to drive and contribute to a whole team approach to quality.

MAKING THE TRANSITION

Making the transition to agile development takes time. Regardless of which development process you're currently using, it will likely take several sprints, and possibly even a release or two, for your project team to feel comfortable with the new processes and way of life.

DESIGN DOCUMENTS

While agile development de-emphasizes design documentation in favor of working software, by no means does that mean all documentation is done away with. Comprehensive specification documents are replaced with shorter, more fluid documents such as user story documents and paper prototypes. Agile development involves working closely with customers to ensure you are developing what they ask for, and allows for refinement and rework along the way. A user story document describes what the user story is to do, acceptance criteria for the user story, and any additional information useful to Quality Engineering (QE) and Information Development (ID).

It's essential that the team understand the problem they are trying to solve with each user story, as well as define clear acceptance criteria. This understanding helps the team work more closely together to solve pain points for their users in a faster and higher-quality way.

THE WHOLE TEAM APPROACH

Though NetIQ has been working in scrum for a few years, we had adopted some practices that weren't healthy or beneficial. In addition, our failures to ship products when we said we would pushed us to look at what we were doing that wasn't working and make some changes.

THE OLD WAY

We were functioning at the most basic level in scrum teams, yet many of our teams had divisions among functional areas. These silos led to the whole team not seeing the same vision of what we were building. Often Development would meet with Product Management ahead of time and get a thorough description of the requirements, or define the user stories first, then bring in QE and ID later. Or Development looked at their prototypes and QE looked at their test cases, but they didn't always match up because there wasn't well-defined acceptance criteria for each user story.

The silos the teams found themselves in led to QE seeing themselves as the sole guardians of quality. They were the gate the product had to go through to be labeled "ready to ship." Documentation was viewed as the sole responsibility of the ID team.

This separation often led to QE and ID tasks lagging sprints behind Development, rather than Development, QE, and ID working together as a team to code, test, and document user stories all within the same sprint. Rather than calling a user story "done," we tended to call it "done" when Development was done coding, and "done done" after the story had been tested and documented. While the "done done" designation sounds silly, it's descriptive of how the tasks for user stories were previously viewed – that is, two different stages of "done" were used, which goes against the ideals of an agile team.

THE NEW WAY

After being retrained, and after many conversations with Product Management, we have started looking at our approach in a new way. Now when we plan a release and populate that backlog, the whole team understands exactly what we are (and are not) building. The entire team accepts responsibility for completing **all** aspects of each user story, including testing and documentation tasks.

Also, the entire team agrees that the user story is not done until it is coded, it is tested, all stop-ship bugs are fixed, and it is documented. This thought process helps spur the team to plan for testing and documentation tasks to occur within the same sprint as development tasks, rather than lagging behind. As a result, we now have one definition of "done," and that means the user story has been coded, tested, and documented. If any of those items aren't done, the story is considered "not done."

HOW DO YOU GET THERE?

Making QE and ID truly part of the project team means giving them the support they need to adapt to the pace of agile development. Ways to help improve this include the following:

- Ensure your developers do unit testing. The more complete a piece of code is before going through acceptance and regression testing, the less likely it will be returned with bugs.

- Set up automated acceptance and regression testing. Manually testing everything in your software is hugely time-consuming and reduces your flexibility in shipping quickly.
- Include QE and ID in estimates. Basing a release date off of Development estimates alone is a recipe for failure.
- Fix bugs within sprints. Every time a bug escapes a sprint, you're building technical debt into your product.
- Help in other areas when your tasks are done for a sprint (for example, automating test cases, doing usability testing, setting up environments for others, writing first drafts of documentation, grooming the backlog). While this is ideal, note that resource constraints and obligations to other projects might mean you need to work on another project's sprint tasks once you are done with this project's sprint tasks.
- When sprint planning, ensure the team develops vertical slices (back-end, middle, user interface), rather than working on one component per sprint.

Feature testing, regression testing, and documentation must occur **throughout the sprint**, not just at the end of the sprint. If your team has a testing or documentation crunch at the end of the sprint, you are doing something wrong – find out what it is and fix it. Some possible reasons for a crunch like this include the following:

- You have not automated your testing.
- There was not enough information given to the QE and ID team members early.
- Coding was scheduled until the very last day of the sprint.
- There were last-minute scope cuts that had to be handled in the documentation.

RELEASE PLANNING

Release planning involves choosing a set of user stories that have been estimated, that match the Product Manager/Product Owner's requested release theme, which we expect we can fit into a timely release. These decisions are based on the scrum team's expected velocity.

You should have at least enough sprint-able high-priority user stories broken down to fill the first sprint. Ideally, you will have a rough idea of what stories you will address in the first two to three sprints. Resolve major dependencies and uncertainties in early sprints so that you can estimate a date range for shipping.

ADAPTING YOUR REVIEW CYCLE

When we used the waterfall process, we used a review cycle that included three drafts of each book: a first draft, an approval draft, and a quality edit draft. The first draft and quality edit draft were internal to Information Development. All of these drafts happened toward the end of a software release cycle, and were not reviewed by anyone outside the Information Development team until the project was feature complete and all features were documented in one fell swoop. Unfortunately, the time when we sent out our approval draft for review by other functional areas such as QE and Development often coincided with their busiest times of the release cycle – testing the product and fixing bugs.

With agile development, we now write documentation for a feature in the sprint in which it is developed and tested. On more mature products, we no longer send out the entire book as a first draft for ID review. Each

time a feature is documented, that piece of documentation must be reviewed by a peer editor and by QE for technical accuracy before the team can close the user story. We consider that the “first draft” of that piece of documentation. We still send out approval drafts later in the release cycle. For most reviewers, at that point it is simply a chance for them to see the book in its entirety, since they have reviewed all the new and updated parts of the book previously.

The benefits to this approach include the following:

- QE and Development no longer have to review an entire book (or more) at a time.
- ID receives more thorough reviews since QE and Dev have more time during each sprint to review smaller pieces of the documentation.
- The documentation is more technically accurate, and therefore of a higher quality, due to more thorough reviews.
- The writer’s capacity needed for an approval draft is reduced because most of the work has already been done in previous sprints. Therefore, if you have multiple books on multiple products, there is not as big of a hit all at once on your time.

CREATING USER STORIES

User stories define a software system requirement, or what you are building, **from the user’s point of view**. When creating a user story, include enough information for all functional team members to perform their tasks. User stories are often part of an epic, or theme. For the whole team approach to be successful, it is imperative that the Product Manager prioritize the user stories in the backlog, and that the team groom the backlog frequently.

MAKING USER STORIES ABOUT THE USER

User stories should address a pain point for the user. NetIQ teams have been working closely together to ensure that user stories describe the problem the user is having, acceptance criteria, and appropriate acceptance tests. The entire team provides input into this process. Though defining these aspects of the user story this thoroughly when populating the backlog takes more time upfront, it saves a tremendous amount of time later in the sprint because everyone understood what they were building from the beginning.

PROBLEM STATEMENTS

The problem statement for the user story should describe what behavior is functionally broken and how. In other words, what is the user’s pain? The developer working on the issue writes the initial statement, then the product architect, QE, and ID provide input and ask questions. Everyone involved approves the problem statement before it’s included in the user story.

ACCEPTANCE CRITERIA

Acceptance criteria describe how users will know the problem has been fixed. Before QE accepts a user story, it must meet the acceptance criteria. Again, the developer working on the issue writes the initial criteria, and the product architect, QE, and ID provide input and ask questions. Everyone involved approves the acceptance criteria before it’s included in the user story.

ACCEPTANCE TESTS

Acceptance tests are the tests that QE will run to determine whether the user story meets the acceptance criteria. Development and QE determine these, and these rarely require ID input.

WHY SPEND TIME WRITING THESE?

Writing problem statements, acceptance criteria, and acceptance tests for user stories is beneficial in several ways, such as the following:

- For each user story, everyone knows what problem they are fixing (problem statement), why they're fixing it (problem statement), what the expected behavior is once they fix the problem (acceptance criteria), and how they'll determine whether they fixed the problem (acceptance tests).
- The user stories clearly define the scope of the release.
- Pre-defined acceptance tests eliminate unnecessary ad-hoc testing.
- Since everyone agrees on the problem statement and acceptance criteria, they become the main source for writing the documentation. Therefore, writing and getting the documentation approved should require less time.
- They can expose possibilities for usability testing.
- They can bring to light issues that aren't really issues or solutions that don't provide the fix the user really needs. If you can't describe how the issue impacts the user, it's probably not worth spending the time to fix. If you can't articulate the acceptance criteria in a way that shows the user the problem is fixed, you probably haven't identified the root cause of the issue. If you know what the fix is, you should be able to explain why it works from the user's perspective.
- **They force everyone to think about the user impact.**

Doing this work on user story definition benefits each functional area as well. They ensure Development is providing a complete end-to-end solution of what is expected (and nothing more). They ensure QE is testing what's promised from a user perspective. They ensure ID understands the feature in a way that they can write documentation targeted to the user's needs in this area. Finally, they ensure the product owner is comfortable with the team's understanding of the requirements.

BACKLOG GROOMING

NetIQ uses planning poker to estimate time needed for user stories in the backlog. During planning poker, the entire project team discusses the requirements and gets clarification as needed from the product owner. The team then uses story points to estimate the effort required to complete all functional areas' work (Development, QE, and ID) for each user story.

User stories that you will work on in the near future (the next 2-3 sprints) need to be small enough that they can be completed in a single sprint. User stories or other items that are likely to be more distant than a few sprints can be left as epics or themes, and broken down into user stories in future backlog grooming sessions.

The product owner should frequently prioritize the backlog based on story return on investment. While the product owner is responsible for backlog prioritization, the entire team is responsible for backlog grooming.

SPRINT PLANNING

When planning a sprint, determine the length of the sprint and the workload capacity of each team member. Pull items from the product backlog to the sprint backlog during sprint planning.

Estimate your tasks for that sprint in hours. You must ensure that your hours fit in to your capacity **for that project that sprint**.

DETERMINING CAPACITY

When determining your capacity, consider the following guidelines:

- Consider previous sprint estimates, if available.
- Include vacation and non-sprint responsibilities, such as meetings, customer support, and so on.
- Ensure you allow time for non-task sprint responsibilities as well, such as planning meetings, backlog grooming meetings, demos, and retrospectives.
- Estimate approximately 20-25% of your time for non-task activities mentioned above.
- Do not include the first or last day of the sprint (depending on planning schedule).
- Ensure Development finishes early so QE and ID have time to complete their user story tasks before the sprint ends.

When you serve on multiple teams, you could spend 4-5 hours per week in scrum meetings for each project on which you're working. Work with your lead or manager to spread out the workload in such a way that you can delegate scrum meeting attendance to others. Another option is to attend only the scrum meetings that pertain to your highest priority project at the moment. You can also ask the scrum master to send scrum status reports through email to help those who are unable to attend each scrum meeting for each project. While it's preferred you attend each scrum meeting for each project on which you work, sometimes it just isn't possible.

SIZING DOCUMENTATION WORK

You can create more accurate estimates by applying a standard number of hours per page of documentation written based on the level of source material. For example, if a task requires one new page of documentation with little source material, the task will take 6 hours. If you need to rework a page of documentation with some source information, it will take 4 hours.

Using a formula to estimate documentation work can make sprint planning much easier. You can plug in your tasks based on the information in the user story to quickly estimate the amount of documentation work required to complete your tasks. You then compare the number of hours required to complete all documentation tasks in the sprint to the total hours of documentation team member capacity.

Ensure you include tasks for usability testing, user interface review, documentation improvements, bug fixing, video work, etc.

“POTENTIALLY SHIPPABLE” PRODUCT

There is often confusion around what “potentially shippable” means in agile. For our purposes, “potentially shippable” means for just the part of the product developed in that sprint. The documentation for those user stories should be complete and shippable. The user interface review for those user stories is complete and

changes have been implemented. The concepts, procedures, and references necessary for that user story are done.

The entire book and help system probably are not complete and shippable until some ID-only user stories are complete in the last sprint or even after sprints. Other documentation tasks, or ID-only user stories, probably are not done.

ID-DRIVEN VS. ID-ONLY USER STORIES AND TASKS

ID-driven user stories and tasks happen **during** sprints. These might include targeted documentation improvement tasks, usability testing stories and tasks, or user interface tasks.

ID-only user stories and tasks happen in later sprints or after sprints. These might include sending out the approval draft for review, incorporation of approval draft comments, production process tasks, and finalizing release notes.

REFERENCES

Agile Testing: A Practical Guide for Testers and Agile Teams. Lisa Crispin and Janet Gregory.

Agile Testing with Lisa Crispin Blog. <http://lisacrispin.com/wordpress/2011/04/26/the-whole-team-approach-in-practice/>

Product Owner Training for NetIQ. Kenny Rubin, Innolution. www.innolution.com

Rocket Surgery Made Easy. Steve Krug.

Mobile and Agile: The Floating Writer's Survival Kit. 2008 STC Summit presentation, Alyssa Fox and Meredith Kramer.

The Whole Team Approach. Internal NetIQ presentation, Ed Spire.

Delivering "On Time, As Promised, With Quality". Internal NetIQ presentation, Angela Stagg.

AUTHOR CONTACT INFORMATION

Alyssa Fox
Director, Information Development
NetIQ Corporation
1233 West Loop South, Suite 810
Houston, TX 77027 USA
713-418-5334
alyssa.fox@netiq.com

AUTHOR BIOGRAPHY

Alyssa Fox is Director of Information Development at NetIQ Corporation in Houston, Texas. She has been involved in technical communication for over 15 years. Her interests and skills include technical communication, user experience and usability, process improvement, project management, and agile development practices. Alyssa is a senior member of Society for Technical Communication (STC) and is currently serving as the STC Secretary. She has also served STC on the Intercom advisory panel, as Program Chair of the 2012 STC Summit, and in various administrative roles in STC Houston. Alyssa has presented at several STC Summits, PMI Houston conferences, and STC Houston and India meetings, and has won awards in the local STC competition.

OUR PROCESSES ARE A MESS!

Larry Kunz

When projects fail, managers often blame inadequate tools and training. Poor processes often are the root cause, however, and fixing those processes is the key to success. When the system isn't working, you need to find the broken parts and fix them by recognizing common process problems and their solutions, by aligning with the organization's content strategy, and by building a business case to justify your recommendations. Whether you're a consultant or a "captive" employee, you can influence the organization to work more effectively.

The woman on the phone worked for a financial services company. Having recently taken a new job as Technical Publications Manager, she'd quickly discovered that no one knew where all of the company's technical information was stored. People hunted down what content they could find and did whatever they could to patch together technical documents. A Wild West mentality had set in.

That's when the light bulb went on, and she blurted out: "Our processes are a mess!"

Of course she was right. Her company had:

- No clear lines of communication
- No system for storing and sharing information
- Confusion about roles, and who was responsible for doing what

As a result, her customers saw content that was inconsistent, incomplete, and inharmonious with the company's hoped-for brand image.

A PROCESS FOR FIXING YOUR PROCESSES

Fixing technical communication processes looks like a big job. Where do you start? Here's a summary of the steps for fixing your processes:

1. Watch. Observe. Talk with people who do the work every day. See what their pains and frustrations are. This will expose the parts of the process workflow that don't work well.
2. Start fixing things that are under your control. Depending on your position in the organization, that might be a little or a lot. Whatever you can change to make things better, change it.
3. Enlist the support of influential people. You'll need their backing as you move outward, beyond your immediate sphere of control.
4. Prepare a business case. You'll need to show that investing in process improvements will return a tangible gain for the company's bottom line.
5. Get approval from the key stakeholders who have the power and the authority to make the improvements you've recommended.

Then—and only then—can you think about other issues like recommending new tools, increasing (or reallocating) the budget, and bringing in new people. Too many managers look to these things first. But if your

organization hasn't addressed its process problems, these things will amount to nothing more than rearranging deck chairs.

RECOGNIZING COMMON PROCESS PROBLEMS

When you start looking for process problems, here are some examples of what you might find.

Poor (or no) communication between groups. If Tech Pubs doesn't communicate with Marketing, for example, customer documentation doesn't align properly with product positioning or brand image. If Tech Pubs doesn't communicate with Technical Support, documentation doesn't anticipate the needs of real people who use the product. Customers feel like they're not being well served. You need communication channels that everyone understands and that are easy to use. But don't stop there. See to it that the channels actually are used. This might require some internal selling, or even a change in the corporate culture. Engineering and Development groups, for example, might need to be enlightened as to the importance of things like product positioning and branding.

Muddled roles. People do better work when they know what's expected of them and when their responsibilities don't overlap. When you define everyone's roles clearly, people feel better about what they're doing, they work more efficiently, and they almost always produce better results. It's also much easier to create a project plan and then track progress against the plan.

Content is scattered, in different formats. Here, *content* means both source files and the customer-facing information that's built from those files. If you can decide on a set of standard file formats and set up a single, shared repository, your organization can realize the efficiencies of reusing content. Your customers see better consistency in presentation format, terminology, and writing style.

Inconsistent presentation. Brand confusion, manifested by inconsistent presentation, often points back to the other problems. But it can also be a problem in itself. When the brand message is garbled or inconsistent, customers become confused or frustrated. To harmonize your content, build harmony among the people who create content through reuse, through the use of style guides, and through training.

MAKING THE MOST OF YOUR CONTENT STRATEGY

In a game of chess, which player is going to have more success?

- The player who devises a series of tactics for handling various situations
- The player who devises a strategy, and then selects tactics to fit the strategy

Processes represent *tactics*, and as such they're much more effective when they're formulated within the context of a *strategy*.

Rahel Bailie defines content strategy as "...aligning content to business goals, analysis, and modeling." ¹ The key here is that content has to support the goals of the business. Content emerges from the obscure hinterlands of the Tech Pubs or Marketing department and finds itself in the center of the board room, aligned with key business processes like branding and product positioning.

So find out whether your organization has a content strategy. It won't be hard. In fact, if you can't find the answer pretty quickly, then it's a good bet that your organization doesn't have a content strategy.

If your organization has a content strategy, this is great news. You can propose process improvements that align with the strategy, and you'll probably get buy-in fairly easily.

If your organization doesn't have a content strategy—which is the far more common situation—you should try to create one. You? Really? Yes, you. It's a matter of making your voice heard.

MAKING YOUR VOICE HEARD

What you have to say deserves to be heard—even if you don't occupy a corner office. But it'll be heard only if you speak the language of the executives, and that means making a business case.

The business case, very simply, shows that investing in something (in this case, a process change) will result in financial gain that surpasses the cost of the investment. Your business case will fall into one of two categories:

- Cost avoidance: saving money by eliminating costs
- Revenue enhancement: earning more money, usually through increased sales of the product

In other words, it results in a positive *return on investment* (ROI).

It might not be obvious how to build a business case or how to translate a change in the Tech Pubs process into positive ROI. For help, you can find some good examples in O'Keefe and Pringle's *Content Strategy 101*². They illustrate ways that an organization might estimate the costs and benefits of process improvements, and they prove that technical communicators really can affect the bottom line.

RESOURCES

Bailie, Rahel Anne, and Noz Urbina. Content Strategy: Connecting the Dots between Business, Brand, and Benefits (Laguna Hills, CA: XML Press), 2013.

Halvorson, Kristina. Content Strategy for the Web (Berkeley, CA, New Riders (an imprint of Peachpit), 2010. Page 83

REFERENCES

¹*Bailie, Rahel Anne. Rahel Bailie Provides A Content Strategy Primer (September 2009).*
<http://thecontentwrangler.com/2009/09/13/rahel-bailie-provides-a-content-strategy-primer/>

²*O'Keefe, Sarah S., and Alan S. Pringle. Content Strategy 101: Transform Technical Content into a Business Asset (Durham, NC: Scriptorium Publishing Services), 2012.*

AUTHOR CONTACT INFORMATION

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

AUTHOR BIOGRAPHY

In more than 30 years as a writer, manager, and planner, Larry Kunz has experienced the transition from book-based documentation to today's integrated delivery of information as a writer and as a person managing the transition.

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 course, Managing the Information Development Process, in the Technical Communication certificate program at Duke University. He is an STC Fellow and received the STC President's Award for leading the Society's strategic planning effort.

SUPPORTING CUSTOMERS: ONSITE AND ONLINE

McKenzie Zeiss and Kate Fletcher (Presenters) and Steven Yenzen (Co-author)

We expanded the range of what technical communicators do by providing direct customer support in two ways:

- On-site support when a customer first goes live with a new software system.
- Online support when customers are upgrading or otherwise in periods of heavy documentation use.

Providing direct support increases customer satisfaction with our documentation and our software. It also helps customers learn to use documentation effectively on their own. Helping technical writers interact directly with customers also benefits us by providing direct feedback and the chance to be an immediate, visible part of the customer support team. This talk summarizes Epic's programs to provide onsite and online customer documentation support, outlines the benefits for Epic and for our customers, and discusses lessons learned and ways to scale customer support programs to suit a variety of organizations.

WHO WE ARE

Epic makes software for mid-size and large medical groups, hospitals and integrated healthcare organizations – working with customers that include community hospitals, academic facilities, children's organizations, safety net providers and multi-hospital systems. Our integrated software spans clinical, access and revenue functions and extends into the patient's home. Located in Verona, WI, we employ about 6400 people, including 118 technical writers. Our software is used by 265,000 physicians to treat 157 million patients every year. That's 49% of the U.S. population and 2% of the world's population.

With software so widely used in such critical situations, our documentation needs to meet a wide variety of needs for a wide variety of users. Our audience includes:

- Customer project team members
- Our own implementation and support staff
- Executives
- Physicians and nurses
- Other clinical and administrative end users (front desk staff, technicians, etc.)

We use relatively specialized writing roles to help us create documentation that focuses on customer needs in each phase of installation. We have teams of implementation writers (for customers still setting up their software), training writers (for end users and customer trainers), and support writers (for project teams working on maintenance, optimization, or upgrade). Even with that range of writing roles, we found two consistent periods in customers' software cycles when they needed extra support: go-live and upgrade.

GO-LIVE

A go-live is the moment when customers turn on their new EHR for all of their end users. Epic provides extensive floor support for at-the-elbow assistance and troubleshooting, and together with the customer

project team we maintain a “command center” to take help desk calls and resolve issues for the first two to three weeks. Because our customers are large health care systems with varying, complex, and highly sensitive workflows, customers need strong support from us to ensure that they can provide uninterrupted top-quality care to their patients. While our go-lives are known in the industry for being successful, it’s important to us to keep finding ways to make them easier for customers. In particular, three areas can compound each other and lead to a “cycle of stress” during go-live:

- Users are getting accustomed to a new system
- Project team is extremely busy, so it’s harder to respond to non-urgent needs
- Executives are intensely interested in monitoring go-live success

To meet these needs, we created the Go-Live Writer program to make sure customers have a technical communicator in the command center. Go-live writers create three types of materials:

- Customized end-user training materials
- Project team management materials
- Executive presentations

POST-LIVE AND UPGRADE

After the go-live dust has settled, customer project teams are faced with a different set of concerns:

- Finding content for troubleshooting and optimization
- Processing release notes during an upgrade
- Learning about new content we create to meet needs they’ve expressed

To meet these needs, we created a role for writers at Epic: Documentation Coordinators. Documentation coordinators serve as the librarians of our documentation library and provide ongoing education and support.

Documentation coordinators host online webinars and Q&A sessions to help customers learn about how best to find and use our documentation through our searchable online database, the UserWeb. They are also available to answer customer emails on an ad hoc basis. However, the most important area of support we provide is with release note review during upgrade.

BENEFITS FOR US

For Epic technical communicators, the benefits of both go-live and ongoing support are very similar. Above all, interacting directly with customers gives us the chance to see how people are using our documentation and what unanticipated needs they have. We get intensive immersion learning and direct feedback.

We also benefit from increased visibility and integration. As a writer, it’s easy to end up on the sidelines of the software development and customer support processes. Interacting directly with customers keeps us visible to both them and our Epic colleagues.

Finally, providing direct customer support gives us an emotional satisfaction that is too often missing from technical writing. We don’t just write documents and send them out—we watch people use them. We help people use them, and in doing so we help people improve the quality of health care they provide. Getting the

opportunity to get our feet wet and participate directly in customer support gives writers a feeling of engagement and achievement that goes beyond what we experience from writing and editing alone.

BENEFITS FOR CUSTOMERS

For customers, the benefits differ according to the type of support. Go-live writers boost project teams' credibility with physicians and executives, two audiences particularly demanding of high-quality documentation provided at the moment it's needed. Documentation coordinators help make customer upgrades easier and faster, freeing up the project team to focus more on optimizing the system for their users.

LESSONS LEARNED

As we've developed the Go-live Writer and Documentation Coordinator programs, we've learned a bit about what's most important for making writer support programs successful.

First, make sure the benefits are clear to other roles. Our writer support programs provide a great deal of value to customers, but that wasn't always self-evident to other roles who work directly with customers on a day-to-day basis. Implementation and technical support staff can sometimes feel protective of their customers, making it hard for writers to get access to them. Starting by emphasizing the help we can offer was essential to smoothing out relationships so that we could start working with customers effectively.

Second, make sure all the preparation is in place before you start. For documentation coordinators, that primarily means making sure that you know your customer contacts and work with them early in an upgrade. This prevents them from reaching a crisis point where they aren't sure how to get through the documentation they need to read in time to complete their upgrade on schedule. For go-live writers, it primarily means making sure that you've laid the groundwork you need to walk on site and start writing: Where will you plug in your laptop? Who are your customer contacts? Does the whole customer project team know that you are there and that they should bring user education issues to you? Who will print tip sheets and how will you distribute them?

Finally, we've found it essential to manage priorities and expectations. We are all full-time technical writers in addition to providing customer support, and when we don't set clear limits we risk either disappointing customers or letting our core work suffer. We've found that customers and other Epic staff are very willing to accept those limits when they are made clear from the outset.

SCALING THE PROJECT

Our customer support programs are robust and involve a relatively large time commitment and travel expenses. Not all technical writers work in circumstances that allow them the time or the budget to make the same commitments. However, our programs are large-scale examples of strategies that you can implement in a smaller version.

- Start by deciding what your key goals are. Your goal might be to make go-lives easier for customers or to gather direct feedback on your documentation. It's also important to be honest at the outset about what's feasible for your organization, in terms of time, money, and the personalities of the writers involved. At Epic, both our go-live writer and documentation coordinator programs are on a volunteer basis because not all writers are interested in doing that kind of work. Trying to force writers

who aren't interested in direct support to interact with customers won't be successful. Consider trying a pilot or an internal version first. For example, you might decide that you can't send writers on site for a go-live, but you can have them available online at your organization to create customized training documentation and send it to customers. Lower-maintenance versions might entail having writers do periodic online demos or Q&A sessions without having individual writers assigned to customers to provide ongoing individual support. Along the same lines, consider adding a writer to a group that is already traveling to a customer site for a chance at some immersion learning. Above all, don't be billable. We are an excellent value for customers when we are expenses-only, but most customers are very budget-conscious. The benefits we reap from both our customer support programs are significant enough that it's well worth it for us to provide support without billing for our time.

AUTHOR CONTACT INFORMATION

McKenzie Zeiss
Technical Communicator
Epic
1979 Milky Way Dr.
Verona, WI 53719
608.777.9000

Kate Fletcher
Technical Communicator
Epic
1979 Milky Way Dr.
Verona, WI 53719
608.777.9000

Steven Yenzer
Technical Communicator
Epic
1979 Milky Way Dr.
Verona, WI 53719
608.777.9000

AUTHOR BIOGRAPHIES

McKenzie Zeiss has been a technical communicator at Epic, a healthcare information technology company, for three years. She has a Ph.D. in English from the University of California—Irvine.

Kate Fletcher has over 15 years of communications experience in the IT, construction, and financial services sectors. She works as a technical communicator at Epic, a healthcare IT company. Her projects include methods to help customers prioritize release notes, internal and external publicity of documentation, and customer education/support. Kate has a Bachelor's degree in English and history and an MBA in risk management from the University of Wisconsin—Madison.

Steven Yenzer has spent nearly three years at Epic as a technical communicator. He graduated from the University of Maryland with a degree in English Language and Literature.

PROFESSIONAL DEVELOPMENT

[Building and Maintaining Your Personal Brand](#) page 141

[Interviewing SMEs: Covering the Bases and Hitting Homers](#) page 145

[Shockproofing Your Use of Social Media](#) page 155

[Staking Your Claim on the Social Media Frontier](#) page 159

BUILDING AND MAINTAINING YOUR PERSONAL BRAND

Larry Kunz

Your personal brand is your professional identity. It's up to you to create and cultivate it. Today's social media offer new opportunities and a new urgency for personal branding. Learn how your colleagues have built their personal brands, and discover how you can build an effective brand for yourself.

We used to call it *reputation*. It's the impression that other people in our professional lives form about you, the personality that you want to project as a professional.

Seth Godin, the media-savvy author and public speaker, defines a *brand* as “the set of expectations, memories, stories and relationships that, taken together, account for a consumer’s decision to choose one product or service over another.”¹

We often think of branding in terms of companies like Coca-Cola, Apple, or IBM. Each of those brand names evokes an image in your mind that goes beyond the company’s logo and product line. In the same way, today’s professionals understand that each of us has a personal brand. One of the first was Tom Peters, who in 1997 coined the term *personal branding* and who offered the following advice²:

- Define yourself as an individual, not in terms of the company you work for.
- Figure out what differentiates you from other professionals. What do you want to be famous for?
- Everything—phone conversations, email messages, conduct in a meeting—is part of the brand message you’re sending. Had he written a few years later, Peters surely would’ve added to the list blog articles, Facebook posts, and tweets.

In today’s interconnected and social world, your personal brand spreads faster and farther than ever. It’s more malleable, and it’s more likely to blend your professional life with your personal life. Your personal brand is your professional identity. It’s up to you to create and cultivate it.

GETTING STARTED

I surveyed several technical communicators, all of whom have built strong personal brands, and asked what they find essential to building and maintaining a personal brand. Here’s what they said:

- Experience and exposure
- Focus and courtesy
- Enthusiasm and a well-defined niche or target market
- Being myself; sharing some of my personal interests along with the professional stuff
- Discovering what’s unique about myself

The last three items echo Peters’ advice to figure out what differentiates you from others. It’s also the first step in consultant Robin Fisher Roffer’s process for building a personal brand³. Other steps include attracting your audience, identifying obstacles, finding mentors and coaches, and one that comes up again and again—both in blog articles and in my survey—being true to yourself.

Asked how they use social media for personal branding, survey respondents mentioned two tools in particular: blogging and Twitter. Start a blog, get onto Twitter, and then remain active. A good level of activity is at least one blog post per week and two or three tweets per day. As one respondent said, “continue to put out good, broad, [and] engaging content on a regular basis.”

Blogging and tweeting work well because they’re the easiest way to gain visibility for your brand. You can use them to:

- Build a persona: You’re making a way for people to see you as an individual, as a person, rather than just a name.
- Engage in conversations about the profession: Contribute your opinions and insights, and respond to those of others.

While a blog and a Twitter account will probably form the underpinning for your personal brand, you need to develop a whole package. Author and blogger Dan Schawbel lists a mixture of things, some social media and some traditional, that go into creating a brand⁴:

- Business cards (have your own cards, even if you’re employed)
- Résumé and cover letter
- Video résumé (this one might be ahead of its time)
- Portfolio
- Blog or website (consider buying the domain *yourname.com*)
- LinkedIn profile
- Facebook profile
- Twitter profile
- Wardrobe
- Email address (firstname.lastname@gmail.com is fine; CoolDude@aol.com is wrong on many levels)

SUSTAINING YOUR BRAND

Like a website, a personal brand doesn’t just get created. It has to be maintained. The ingredients that make up your personal brand have to be managed and cultivated. Set a Google alert to monitor what people are saying about you on the web: who’s favoriting your tweets, who’s linking to your blog, and who’s just talking about you.

Talk with a few trusted colleagues to understand how they perceive your strengths and weaknesses. Use this information as you craft your personal brand.⁵

As your brand gains visibility, you’ll need to make decisions about:

- How much (if any) personal information to mix with the professional. You can reveal things about your family (with caution, for privacy reasons), your activities, and your hobbies. It’s probably best to stay away from religion or politics: for better or worse, people will judge you.
- The conferences and events you’ll attend (and speak at).
- Your involvement and visibility in professional organizations like STC.

- Dress and physical appearance. Do you refuse to work anywhere you can't wear blue jeans? Or are you convinced that everybody loves a sharp-dressed man?
- Diction and language. When I speak, I don't mind if you hear a slight Southern drawl. With the right audience, I might even sprinkle in some regional idioms. But you won't hear me curse or swear—not because I'm an altar boy, but because it isn't the image I want to project.

There's no one right or wrong answer in these areas. Make the choices that work for you, but understand that every choice affects the way people will perceive you. Always make choices that reinforce the brand image you're trying to project.

MORE GOOD ADVICE

There's much more to say about personal branding. All of the people listed in "References" will give you good pointers. Here are a few.

Pursue topics that you're passionate about and that you enjoy. The process of maintaining your personal brand becomes a lot more interesting and satisfying.

While blogs and Twitter are great tools, words alone won't cut it. People need to see you in action. In the words of leadership consultant Greg Hartle: Do more, talk less. Show, don't tell.⁶

Write content and select activities that help build *trust* in your personal brand. Companies and individuals want people to feel comfortable interacting with them.⁷

Prepare to be surprised. One survey respondent has developed a reputation for being an expert at giving webinars—even though that wasn't what she set out to do. She's embraced it, though: in addition to her regular webinars, she's begun giving webinars on how to give webinars.

It's not a competition. If you define yourself in terms of your employer, or in terms of others, you'll find that your personal brand is at the mercy of those organizations and people. While it's fine to let people know where you work, be sure that they see you not as a cog in the corporate machine but as an individual who happens to work there. In the end, it comes down—again—to being true to yourself.

Details matter, but keep the big picture in mind and don't get sidetracked by minutiae. What you do as a professional, day in and day out, is more important than tweaking your Facebook photo.

Be open to forming new ties with other professionals. As an example, one survey respondent described how her interest in needlework led her to meet-ups at conferences, which strengthened her professional alliances and introduced her to new ones.

ACKNOWLEDGEMENT

Special thanks to the technical communicators who participated in my survey: Sharon Burton, Sarah Maddox, Sarah O'Keefe, Alan Porter, and Kai Weber.

REFERENCES

[1] Cited by Margie Clayman. "Manage Your Marketing: Building Your Brand Top to Bottom." (January 14, 2013) <http://claymanmarketingcommunications.wordpress.com/2013/01/14/manage-your-marketing-building-your-brand-top-to-bottom/>

[2] Peters, Tom. "The Brand Called You." *Fast Company* (August/September 1997) <http://www.fastcompany.com/28905/brand-called-you>

[3] Roffer, Robin F. "8-Steps to Igniting Your Personal Brand." (July 2012) <http://bigfishmarketing.com/8-steps-webinar-thankyou/> (free registration required)

[4] Schawbel, Dan. "Personal Branding 101: How to Discover and Create Your Brand." Mashable (February 5, 2009) <http://mashable.com/2009/02/05/personal-branding-101/>

[5] Quast, Lisa. "Build A Personal Brand, Not Just A Career." Forbes (November 19, 2012) <http://www.forbes.com/sites/lisaquast/2012/11/19/build-a-personal-brand-not-just-a-career/>

[6] Hartle, Greg. "Personal Brand: It's Not All About You." Business Unplugged (2012) <http://www.carolroth.com/blog/personal-brand-its-not-all-about-you/>

[7] Kunz, Larry. "Building Trust in a Corporate Blog." I'd Rather Be Writing (guest post—October 21, 2011) <http://idratherbewriting.com/2011/10/21/building-trust-in-a-corporate-blog/>

AUTHOR CONTACT INFORMATION

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

AUTHOR BIOGRAPHY

In more than 30 years as a writer, manager, and planner, Larry Kunz 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 course, Managing the Information Development Process, in the Technical Communication certificate program at Duke University. He is an STC Fellow and received the STC President's Award for leading the Society's strategic planning effort.

INTERVIEWING SMES: COVERING THE BASES AND HITTING HOMERS

Jennifer Anthony, Carrie Chambers, Frances Gambino, and Barbara Giammona

Truth or urban legend: Will brownies guarantee a Subject Matter Expert's time? When you do secure a meeting, how can you make the best use of that time? Here are some SME-handling tips from a seasoned, multi-industry, panel of experts to help you cover the bases and hit home runs in your interactions with SMEs.

It's a constant in our profession – the sometimes arduous task of collaborating with a Subject Matter Expert (SME) who is busy, elusive, or outright uncooperative. Regardless of the industry in which you work or the type of content that you create, at some point you will face the challenge of drawing out timely and complete information from expert partners in order to complete your project. While you can certainly do a lot of the legwork on your own by digging through reference materials, design documents, marketing plans, and corporate websites, there is no substitute for the expertise or perspective that a SME can provide. In these cases, you *will* need to coordinate the input of others to deliver accurate, quality results.

We can make light of the challenge by swapping stories of how to win over SMEs with brownies, but the truth is that there is much more to making a successful connection with a SME. Consider the following advice from four seasoned experts in the Technical Communications field on how to deal with the most common problems you may encounter when working with a SME.

WHAT IS YOUR FAVORITE SNEAK TECHNIQUE TO GET SMES TO PLAY ON YOUR TEAM? HAVE YOU EVER USED BROWNIES?

A smile (even in virtual relationships) and a pleasant, responsive, team-oriented, professional attitude go a long way toward warming the hearts of difficult or reluctant people. I admit that I have used brownies, cookies, chocolate, and even flowers *after the fact* to thank people for their support of my efforts. But overall, the best sneak techniques in my arsenal has always been to be direct about what I want, asking for it in a timely manner, being persistent (remaining polite even when I am applying “tough love”), and expressing appreciation during and after the project. Technical Communicators are often sarcastic people who use their language gifts to subtly attack their colleagues. Try to watch out for that in yourself! In the absolute worst case, you can always call in the “big guns,” going to your manager or over the head of an uncooperative person to get what you need. But I try to resist that approach, leaving it only as a very *last* resort.

My favorite list of suggestions in this area comes from *Technical Writing 101* by Alan S. Pringle and Sarah S. O'Keefe. On pages 84 – 86 of their handy general guide to our profession, they cite “(Almost) 30 ways to get information from developers.” Many of the suggestions in this article expand on ideas found on that list. (Barbara Giammona)

ARE THERE ANY TOOLS YOU HAVE USED TO COLLABORATE WITH SMES THAT HAVE BEEN PARTICULARLY HELPFUL?

Although interfacing with SMEs face-to-face is often ideal, it is not always practical. You may be busy, the SME may be busy, you and the SME may not be present in the same geographical location, or there may be

other communication barriers. The following collaboration tools are especially helpful when working with SMEs to gather information, review content, and test documentation:

- **E-mail.** This is an obvious tool that every writer uses on a daily basis already. Keeping e-mail records of conversations with SMEs can be particularly helpful when you need to refer back to previous communications. Also, e-mail allows both you and the SME to respond after all information has been gathered and at appropriate times (which is especially helpful if you are in different time zones).
- **Telephone.** If you need a quick response – and you need it now – the old fashioned telephone is still a good tool to use.
- **Video Conference.** When you need to speak face-to-face, but work in different geographic areas, this tool comes in very handy. It is always nice to put faces with names, and a video conference can help build your professional relationship with the SME.
- **Instant Messaging.** This tool has expanded in recent years to allow you to keep records of your conversations, initiate video chat, and also initiate screen sharing. You can often get immediate responses from your SMEs via instant messaging because people can respond while they are multitasking on conference calls, etc.
- **Wikis.** Most companies have at least one wiki now, which can provide a great sharing repository between you and SMEs. Use wikis to track project work, deliverables, and meeting minutes in one place where everyone can access them.
- **Bug Tracking Systems.** Although it may not seem like it, bug tracking systems can be a good SME collaboration tool. You and the SME can work together on items and keep track of your work and resolutions in the system. Your work is visible to others (project managers, product managers, other developers, QA specialists) and can promote total team involvement. (Carrie Chambers)

HOW DO YOU DEAL WITH SMES FOR WHOM ENGLISH IS NOT THEIR FIRST LANGUAGE?

Working in a global economy means that you are often collaborating with people who do not share your language, culture, or norms. Although sometimes frustrating when communicating, building businesses with workers from around the world is extremely beneficial to output, customer relationship building, and innovation. As a technical communicator, you often need to extract information from SMEs who speak English as a Second Language (ESL). To successfully communicate with these SMEs, it is often best to first employ some of the collaboration tools mentioned above that rely on written, not spoken, communication (like instant messaging, wikis, and e-mail). It is often helpful to repeat back to the SME what you believe just heard to ensure that you are communicating effectively and clearly. You should not be afraid to let the SME know when you are having a hard time understanding – just make sure that you are courteous and polite. (Carrie Chambers)

YOUR SMES ARE TOO BUSY TO RESPOND—NOW WHAT?

When project schedules are compressed, and everyone is doing the job of two people, pleas to the SME for needed information can often fall on deaf ears. Getting the information you need from non-responsive SMEs requires a two-pronged approach:

Make it as easy as possible for the SMEs to help you!

- Don't be vague. Provide direct, detailed questions.
- If you have a question that references existing text, send along a PDF page extract, with the text in question highlighted. Don't make SMEs search for the paragraph you have a question about.
- If you have questions about specifications or bug reports, provide links to the exact document or bug report number you are referring to. Even better, pull out the text you are referring to, and include it in your email or question list.
- Provide as much background information as you can. Reduce the SME's research burden.
- Your SMEs are more likely to promptly respond if they think it will only take 10 minutes of their time to provide the answer.
- Ask your SMEs how they would like to work with you. Do they prefer you send all your questions to them at once? Or would they rather you send them one at a time? Do they prefer email or in-person meetings?

Never threaten, and don't make them mad! Be polite.

- Whenever possible, provide plenty of time for the SME to respond to you. A deadline of two days is more likely to be met than a deadline of two hours.
- Maintain a positive tone in all your emails. Send friendly reminder emails when a deadline is missed. Even if you are angry or frustrated, don't let that come out in your email, as it will get you nowhere.
- Make an in-person visit when possible. Drop by their cubicles with a smile on your face to let them know you are still waiting for their answers.
- Only escalate an issue to your project manager or the SME's direct manager as a last resort.
- Remember to say please and thank you. It sounds simple, but the basics of etiquette can go a long way when working with SMEs. And, if a SME goes above and beyond, don't be afraid to let his or her manager know how awesome they are. Flattery will get you everywhere.(Jennifer Anthony)

HOW DO YOU MEET YOUR DOCUMENTATION GOALS WHEN THE CULTURE OF THE COMPANY DOESN'T ASSIGN A LOT OF VALUE TO THE DOCUMENTATION EFFORT?

While most companies know that they *need* user documentation, the value assigned to creating that documentation isn't always high. A company culture where user documentation is thought of as a "necessary evil" can make it difficult for you to do your job. How do you convince SMEs to help you if they don't see that as an important—or required—part of their job?

In these circumstances, personal relationships make all the difference. Even if the overriding attitude is "no one reads the manual, so we don't need to put much effort into it," once you have personal relationships with the people on your project team, you can usually get them to help you.

- Friends don't let friends write bad user manuals! Become friends with your fellow SMEs: laugh, joke, share frustrations, commiserate about project obstacles, and go to lunch once in awhile. Offer to help them do their job by volunteering to edit specs or review a GUI.

When you have cultivated a friendship with your SME colleagues, they *want* to help you. They become more open to answering your questions and providing information. It's not so much that it is part of their job, but rather that they are supporting you as a friend.

- In particular, be friendly with your project manager. He or she has the power to get you the help you need:

Never be late on your deadlines – so your project manager knows you are reliable and here to contribute to his or her success.

Offer your services as an editor, to make his or her job a little easier (some of those project plans are pretty horribly written!).

Make sure your project manager knows you are on his or her side; then he or she will more likely be on your side when you need help getting the information or assistance you need. (Jennifer Anthony)

HOW DO YOU WORK WITH SMES ON VIRTUAL TEAMS?

Whether your SME is located 100 miles away or half-way across the world, working with a SME when you are on a virtual team can pose some extra challenges.

It's entirely possible that you have never met your SME in person, or even spoken with them on the phone before the start of your project. In this case, it's important to enlist the help of other team members that you already have a relationship with.

- Ask your project manager, or anyone with whom you have a mutual relationship, to "introduce" you via email or a conference call. This helps break the ice, and is less awkward than calling someone out of the blue with a question you need answered right away. This also helps set the stage for your role on the project team, and what the SME can expect from you and vice-versa.
- It's even better if the person doing the introducing is on-site with the SME. For example, if I know offshore writers on another project who are on-site with the SME I need to work with and who already have a relationship with that SME, I'll ask them to introduce me to the SME. A friendly face can open many doors!

Once you've been introduced to your virtual SME, many of the techniques you use for dealing with a busy SME are even more important when you are working with a SME via email or telephone.

- Set up conference calls when possible, on a standing weekly or bi-weekly basis when necessary.
If there is a large time difference (for example, 12+ hours), try to schedule calls for a time that is more convenient for the SME. Remember, you are asking for their help, and you want to make it as easy as possible for them to do so!
- If there is a language barrier (heavy accents or poor phone lines make it difficult to have a productive conversation), stick with email. Don't spend an entire conversation asking the SME to repeat statements or talking around the static of a poor connection. It is as frustrating to the SME, as it is to you. And it may make the SME less willing to help you.
- When sending questions or review requests via email, provide as much information as possible. If you send a question to a SME in another time zone, don't make the SME wait for a clarification from you before s/he can provide the answers you need! This wastes valuable time. Send as much information as you have, or think is necessary in order for the SME to answer your question. (Jennifer Anthony)

HOW DO YOU DEAL WITH GETTING REQUIRED INFORMATION FROM PEOPLE OVER WHOM YOU HAVE LITTLE OR NO POWER OR INFLUENCE?

Most of my technical communication roles have involved working with SMEs who are members of different organizational units – outside my project team, corporate hierarchy, or company. Consider a similar situation in which you are a technical communicator who has little or no organization-based influence over the SME from whom you need to gather information.

Suppose that the SME has been identified by a professional network third party as a viable resource for critical information that you need to complete a task. Or perhaps the SME has been identified as a source of information by his organization, but he has prioritized the task lower than your project deadline can support – or not prioritized the task at all! Even worse, consider that the SME is a corporate resource who operates in a “world” of his own, is a classic “difficult person,” or is simply not willing to share his time or expertise. In these cases, there is no appealing to the SME’s manager. You need to combine your people skills and your technical communication skills to get the information that you need.

In situations where organizational dynamics do not facilitate or guarantee the cooperation of the SME, or in other instances in which you need to obtain information from an SME over whom you have little or no influence, give the SME good reasons to willingly provide the information that you are seeking. When I need to obtain required information from an SME over whom I have little or no power or influence, I focus on appealing to the SME’s professionalism, or ego, using these **Three Ps**:

1. **Partner** - Help the SME see himself as a valuable partner in the delivery of an important, quality solution:

Create an environment of partnership in which to collaborate, basing your end of the partnership on respect for the SME’s time and expertise.

Clearly define what you need from the SME in the context of how s/he can effectively partner with you to deliver the best result.

For example: *“I’m writing a training manual on how to adopt the agile project management model in a financial software development environment. Your name has come up again and again throughout the company as the person who has done this the best so far here at ABC. I’m looking for your answers to these four questions below. I expect it will take you no longer to answer than it does to conduct your morning scrum meetings. Please consider sending me your responses by next Friday so I can reflect your expertise in my draft document.”*

2. **Position** - Position the SME’s contributions so s/he will, willingly, offer support:

Frame the SME as a true, industry subject matter expert for the specific information that you are seeking. Relay how he was precisely identified or highly recommended as the Subject Matter Expert to supply the information that you are seeking.

Describe how the SME’s input critically impacts the quality of your result. Communicate how or why a successful solution relies on his unique expertise or perspective. For example: *“Your agile management implementation represents the best practice in your company. I’d like to use it as a case study or as part of the Best Practices chapter in my document.”*

Structure and describe the information gathering process to show how easy it will be for the SME to provide the information that you need. Use the techniques previously offered to minimize the SME’s input time.

3. **Promote** - Convince the SME to collaborate on a project by relaying how his participation can promote his professional or SME status:

Describe how his input will help represent the result well within his own professional circle or in other professional circles. Use the suggestions previously noted to thank the SME directly. Additionally, publicly show appreciation for the SME's participation or otherwise publicly promote his contributions as valuable and noteworthy.

For example: Reflect the SME's contributions in your document's *Resource List* or *For More Information* section.

These **Three Ps** will not only help you succeed in gathering information from an SME over whom you have little or no influence, but they can also help you establish a professional connection that you can tap into again or create a network channel that flows both ways between your expertise and the SME's! (Frances Gambino)

HOW DO YOU HANDLE GETTING INFORMATION IN SHORT-CYCLE, AGILE PROJECTS?

Short-cycle technical communications projects typically require the same stages as longer-term projects. The main challenge in managing short-cycle, agile projects is quickly creating an effective connection with the SME to support the information gathering process.

For example, the practice of meeting once a week will not effectively support the delivery of a three-week project. In a short-cycle project, contingencies will likely require immediate attention in order to keep the project moving to completion in the shorter span of time. Consider these tips and techniques for gathering information in short-cycle, agile project environments.

As a true project management discipline, agile project management can be difficult to anchor or implement. However, the most high-level principles of agile project management can quickly be applied to short-cycle project scenarios. These core agile project management practices will facilitate your information gathering process in short-cycle projects:

- **Build a solid project plan, flexible enough to support the achievement of a short-cycle goal.** For example: The project plan milestones to update a 550-page training manual will vary depending on whether you have six weeks or six months to deliver your result. For the six-week project cycle, you may choose to schedule the project's test, write, update, edit, and index milestones chapter-by-chapter rather than separating the test, write, edit, index phases across two or more passes of the entire document.
- **Establish a pattern of frequent, productive checkpoints with the SME.** True agile project management practices call for a daily "scrum meeting" in which all project participants report on three basic points: *What did I accomplish yesterday? What will I do today (next)? What are my roadblocks?* For short-cycle projects, extend these concepts to your interaction with the SME. For the six week project cycle noted above, you may choose to hold a 30-minute checkpoint with the SME every other day and schedule chapter review drafts to pass between you and the SME at each checkpoint. You may want to hold these checkpoints in a neutral setting, or conduct them over coffee in a meeting room, to avoid distractions and facilitate an efficient dialogue.
- **Publish status continuously.** To maintain a productive sense of urgency and a collaborative sense of progress, as well as provide an easy reference for breaking down roadblocks, publish an end-of-day status for the SME and copy all project participants who need to support your deliverables.

Consider using simple headings and bullets. Be sure to pose requests for information or validation so the SME can easily provide an annotated response (i.e., YES or ACCEPT).

- **Develop contingency plans before you need them.** To avoid project delays caused by unexpected problems, set aside 15 minutes a day to anticipate what could possibly go wrong and work through possible solutions on your own. This preventative maintenance training can help you react quickly to problems when or if they occur. By anticipating problems beforehand, and regularly practicing contingency planning, you will also be able to react more confidently when problems do occur. (Frances Gambino)

WHAT ARE PARTICULAR ISSUES YOU MIGHT DEAL WITH REGARDING SMES AS A MANAGER OF TECHNICAL COMMUNICATORS?

The biggest issue that managers of Technical Communicators deal with in relationship to SMEs is actually the personalities of Technical Communicators. We are a profession of “creative/technical” people, who can focus with laser-sharp attention one minute on the minutia of the writing and editing process, and the next to broad, sweeping concepts, levels above what we are writing about. Either extreme swing of that pendulum can potentially make us someone that SMEs have trouble working with. Most SMEs want to drop a rough draft of written material in our lap or have one quick meeting with us to feed us facts, and then be done with us until the final document review. But we often either want or need to engage with these busy people more deeply than they want us to. Knowing just when to approach someone, how to use the time when you do, and being politically and personally savvy (face-to-face and in e-mail) are probably the biggest challenges in initiating and maintaining good SME relationships. Managers – you need to have frank conversations with your employees about their skills in these areas. And employees – you should be open to understanding where you may need to improve and work with your boss on a plan to build your skills in these areas. (Barbara Giammona)

RESOURCES

Blanchard, K. and Johnson, S. The One Minute Manager (New York, NY: William Morrow & Company, Inc.), 1982.

Carnegie, Dale. Secrets of Success, Dale Carnegie's Golden Book (March 23, 2013).
http://www.dalecarnegie.com/assets/1/7/Secrets_of_Success.pdf.

Carnegie, Dale & Associates. How to Win Friends and Influence People in the Digital Age (New York, NY: Simon & Schuster, Inc.), 2011.

Schwaber, K. Agile Project Management with Scrum (Redmond, WA: Microsoft Press, Microsoft Corporation), 2004.

REFERENCES

Pringle, Alan S., and O'Keefe, Sarah S. Technical Writing 101 (Research Triangle Park, NC: Scriptorium Press), 2009

AUTHOR CONTACT INFORMATION

Jennifer Anthony
Principal Technical Writer
Invensys Operations Management
26561 Rancho Parkway South
Lake Forest CA 92620
949.638.8002

Carrie Chambers
Director of Documentation
Eyefinity
38 Discovery, Ste. 250
Irvine, CA 92618
949.390.8306

Frances Gambino
Executive Director, Technical Content Management
Information Builders
2 Penn Plaza
New York, NY 10121
917.339.6730

Barbara Giammona
Senior Manager, Communications and Technical Publications
Invensys Operations Management
26561 Rancho Parkway South
Lake Forest CA 92620
949.639.8354

AUTHOR BIOGRAPHIES

Jennifer Anthony has been a technical communicator in the Orange County high-tech industry for nearly 20 years. She is currently a Principal Technical Writer at Invensys Operations Management, specializing in developing software documentation and user assistance, and leading the style guide committee. She is Senior Member of STC, and holds a BA in Communications - Journalism from California State University, Fullerton. She also holds a Certificate in Technical Editing from STC.

Carrie Chambers is a Certified Professional Technical Communicator (CPTC) with a Master's degree in Technical and Scientific Communication from Miami University in Ohio. She has been a technical writer for over 10 years and is currently the Director of Documentation at Eyefinity in Irvine, and a lecturer at California State University, Long Beach, for a Technical Communication class. Carrie is a senior member in STC and has been a member of the Orange County STC (OCSTC) chapter for 10 years. In the past, Carrie has held the position of *TechniScribe* Newsletter Managing Editor and Nominating Committee Member and Chairperson. Carrie is currently the OCSTC Scholarship Chairperson, OCSTC 2nd VP of Membership, and the Public Relations Manager for the Southern California Technical Documentation Competition. Carrie has won two STC Awards of Excellence for her work on the *TechniScribe*, multiple awards in the Southern California Technical Documentation Competition, and the 2011 OCSTC Distinguished Chapter Service Award.

Frances Gambino is the Executive Director of Technical Content Management at Information Builders, primarily located in New York City. She has worked as a Technical Communications Manager, and as a Technical Communications adjunct professor, for over 25 years. Frances holds a Bachelors Degree in Technical Communications from Polytechnic Institute of New York and a Masters Degree in Library & Information Science from Drexel University.

Barbara Giammona, panel moderator, is a technical communicator who has just passed the 30-year mark in her career. With nearly 25 years experience as a manager of technical writers in several industries, Barbara has been a frequent presenter at STC events, especially on management topics. She is currently the Senior Manager of Technical Publications for Systems R&D at Invensys Operations Management. She is an STC Fellow, and holds a BA in English from UC Irvine and an MS in the Management of Technology from the Polytechnic Institute at New York University.

SHOCKPROOFING YOUR USE OF SOCIAL MEDIA

Ben Woelk

Shockproofing Your Use of Social Media is part of the Professional Development progression at Summit 2013. It provides an overview of security practices and tools that help you better protect yourself and your company/client online.

TEN TIPS

Most professionals agree that they should be using social media as part of their online presence. However, every week there are reports of compromised accounts and inappropriate use. Here's a "top ten" list of practices to use social media safely.

- **Tip #1: Use strong passwords**

Passwords provide the first line of defense. Are you using the same password for multiple accounts? Length trumps complexity. For example, ItwasaDark4&stormyNight is a much stronger password than 7&*(jyld?Vd34 (and much easier to remember). Consider using a password safe to generate and store passwords for use online.

- **Tip #2: Keep your computer patched and updated**

Attackers take advantage of vulnerabilities in operating systems and applications to attack computers. Ensure that you keep your software up-to-date.

- **Tip #3: Use appropriate security software**

A key concept in information security is the idea of layers of defense so that if one layer is breached, the other layers will stop the attack. One method of accomplishing this is through the use of antivirus and anti-spyware and by using a firewall.

- **Tip #4: Learn to recognize phishing and other scams**

Phishing, the act of tricking a user into revealing his or her login credentials, is a key method organized crime uses in identity theft. Recognizing phishing attempts and not supplying your login credentials is critical.

- **Tip #5: Use social networks safely**

Organized crime attacks users of social networks regularly. One reason is because users on social networks tend to be less careful about clicking on links that appear to be sent from friends. If you receive a link to a video or picture from a friend, contact the friend to ensure that he or she actually posted or sent you the link.

- **Tip #6: Remember who else is using social networking sites**

Organized crime and online predators leverage social networks to spot and attack victims. Employers also research job applicants on social media to see what image they portray. Think about the image you portray. Remove inappropriate pictures. Untag yourself in inappropriate photos taken by others.

- **Tip #7: Be wary of others**

Did you receive a message with a link to a video or webpage? Don't assume that unexpected messages with links on Facebook or Twitter are really from your friends. Their accounts may be compromised.

- **Tip #8: Search for your name**

Concerned about how your name may be used on online? Use "Me on the Web" or Google Dashboard to set up an alert for your name. Google will send you an email at the frequency you specify summarizing when your name has been used.

- **Tip #9: Guard your personal information**

Don't post personal information that can be used to hack your accounts or steal your identity. Do you use your dog's name as a password? Is your dog mentioned on your Facebook page? Make a deliberate decision about what information you choose to share. Don't just provide detailed information when setting up a profile.

- **Tip #10: Use privacy settings**

Social media such as Facebook, Twitter, et al. provide privacy settings. Make sure you understand how these social networks interact. For example, if you're using FourSquare, don't connect it to Twitter. If you do, you risk letting unknown people that you're at a specific location (or that you're not at home).

INFOSEC COMMUNICATOR BLOG

For much fuller discussions of these issues and tips on staying safe online, visit benwoelk.com.

RESOURCES

Woelk, Ben. *Infosec Communicator Blog*, <http://benwoelk.com>.

RIT Information Security, <http://www.rit.edu/security>

Woelk, Ben and Morgan, Hannah, *Bulletproofing Your Career Online*, (Proceedings, STC Summit Conference, 2012).

REFERENCES

Woelk, Ben. *Infosec Communicator Blog*, <http://benwoelk.com>.

Woelk, Ben. "Digital Self Defense for Technical Communicators." *Intercom* (November 2010): 7-9. <http://intercom.stc.org/2010/11/digital-self-defense-for-technical-communicators/>.

AUTHOR CONTACT INFORMATION

Ben Woelk
Policy and Awareness Analyst
Rochester Institute of Technology
151 Lomb Memorial DR
ROS 10-A204
Rochester, NY 14623
585.354.6247

AUTHOR BIOGRAPHY

Ben Woelk is Policy and Awareness Analyst for the RIT Information Security Office. He is responsible for security awareness communications, training programs, and policy development. Ben has authored a number of articles on information security for non-technical users and is adjunct faculty at RIT in Professional and Technical Communication and in Networking, Security and Systems Administration. Current activities include leveraging social media for security awareness, participating in the Awareness and Training Working Group in the EDUCAUSE Higher Education Information Security Council, and Outreach Team Lead for the STC Community Affairs Committee. Ben is a past president of STC Rochester. Twitter: @benwoelk. Blog: benwoelk.com

STAKING YOUR CLAIM ON THE SOCIAL MEDIA FRONTIER

Laura A. Palmer, Ph.D.

Technical communicators are poised to take on significant roles on the social media frontier. Where once social media was exclusively the domain of marketing and public relations, its growth as a method of professional communication means new doors are opening. Technical communicators interested in increasing their professional value can do so by bringing their expertise to the social media functions that operate behind the scenes. Areas such as policy development, audit and compliance, strategic planning charters, and risk management rely on what technical communicators do best. Now is the time to stake a claim in social media's frontier and create a value-added role that you, as a technical communication professional, can bring to the conversation.

For technical communicators who want to move towards working in social media, options exist that many of you may not have considered. In this short paper, you'll learn about key roles and functions that are vital for today's social business model. Not only will these roles resituate the value of a technical communicator in an organization, they'll build bridges with the other communication disciplines in your organization.

Social media is about community, conversations, and engagement as they occur organically across a variety of social network sites. As social media exploded in popularity over the last several years, companies everywhere jumped on the bandwagon to meet customers and join in the online conversations. The impetus to become a *social business* was so pressing that many companies struck out for the social media frontier with little thought about the journey's logistics. This lack of planning meant the social media bandwagon was more of a covered wagon rattling down a rough-hewn trail.

It wasn't long before organizations realized they were poorly prepared to engage with their customers through social media. Missing were policies, strategies, and goals for guiding the social media efforts. Absent was a basic knowledge of legal issues surrounding the social media space, and unconsidered were the ideas of risk, audit, and compliance as they related to social media.

As they become more aware of what's missing in their social business model, companies of all sizes will be looking to fill critical gaps. According to Nora Ganim Barnes and Ava Lescault (2012), an astonishing 65% of fast-growing companies will look to current employees to fill social media roles. Look no further, I say—your technical communicator is ready and willing.

NEW ROLES IN THE PROFESSION

What are the new roles for technical communicators in social media? These new roles start with the changes to rigid occupational boundaries. For most technical communicators, social media typically belongs to the other side of the communication house. Marketing or public relations, as the units responsible for promoting the brand and its external communications, handle the social media functions. With its focus on two-way conversations, corporate messages, and direct engagement with customers, social media hasn't seemed like a natural fit for technical communication professionals and what they feel are their best talents.

However, that communication house model is undergoing a renovation. In larger companies, the once-closed rooms where each communication profession resided are dissolving in favor of an open floor plan. Now, the communications professionals are intermingling in ways they never have. In smaller companies, social media's rise has meant that communication responsibilities have shifted. As Nichole Kelly (2013) points out,

social media involvement has increased the marketing department's workload exponentially. While large budgets and extra personnel can solve workload problems in big companies, many smaller firms, according to Ganim Barnes and Lescault (2012), can't afford a dedicated internal team. Thus, the time has come for other communicators—technical communicators, in particular—to step onto the social media frontier.

THE OTHER SIDE OF SOCIAL MEDIA

Ask most businesses about social media and they'll describe how they use Facebook or Pinterest to reach their customers. Few will talk about the measures in place to ensure good governance and effective oversight. That's because many companies haven't solidified what happens behind the scenes in their social media efforts or determined who'd be the best person for that job. What they need to know is that you, the technical communicator, are uniquely positioned to be part of the social media team.

Consider what you do and how your skills and knowledge would fit in these areas.

- Social media policy development
- Social media audit and compliance
- Social media strategic plan development
- Social media legal/regulatory advisement
- Social media risk assessment management
- Social media style guide creation

DEVELOPING SOCIAL MEDIA POLICIES, GUIDELINES, AND REGULATIONS

Can you see yourself taking what you know about policies and procedures and using that expertise to create a set of social media guidelines? Anxious to engage with their clients and customers, many companies went into social media with no policies in place. Even today, businesses operating as *social* have no such charter at all. Unfortunately, a company opens itself up to several kinds of risk, including reputation damage and legal action, when it fails to codify its social media practices.

Without adequate employee guidelines for basic codes of conduct, online ethics, and even online manners, a social business may see the wrath of angry customers play out in a cringe-worthy way on the company's social media channels. Worse yet, without guidelines for online disclosures of intellectual property, confidentiality, and even regulatory information, the next level of wrath could very well play out in the courtroom. Yet, Ganim Barnes and Lescault (2012), found that only 24% of the fastest-growing private U.S. companies have a written social media policy.

Policies can also extend beyond governing online exchanges. According to Allison Cain (2012) in "The Social Media Scene," policies can extend to what social media products to use, account management and supervision, and even monitoring processes. Because social media practices are a moving target, they're in need of a process that can ensure regular review and updates.

What you can do: Does your company have formal social media policies in place? If not, policy development is the perfect starting place for any technical communicator interested in social media.

AUDIT AND COMPLIANCE

What would you think of compiling all the social media documents—policies, committee meeting minutes, strategy guides—and performing a social media audit? Could you analyze the materials, synthesize key findings, and write a report about your company's social media compliance? While this sounds like a job for a specialized internal auditor, they need not apply. Technical communicators are, once again, ready to step into a key position many companies have yet to consider.

Social media audits and compliance assessments function as a way to determine if the social media activities are aligning with communication policies, corporate messages, business goals, and strategic plans (Jacka & Scott, 2012). These audits also help companies identify weaknesses in their approach and point out areas where risk is a concern.

Technical communicators work with both internal and external documents in their daily jobs; they can be well-versed with the plans that drive organizational change and operations across the entire company. Couple this knowledge with a technical communicator's research and analytical skills, and you have the perfect person to conduct a social media audit.

What you can do: Has your company done a social media audit to ensure its efforts maximize efficiency and reduce risk? If you're a technical communicator looking for a larger research/analysis project followed up with a recommendation report, this is the area for you.

STRATEGIC PLANS

If you walked into the office and asked to see the social media strategic planning documents, what would you get? Would it be a comprehensive plan for the organization's social media efforts or would someone jump from a cubicle and yell, "We're going to start using Twitter next week"?

Unfortunately, most companies substitute tactics—like using a new media channel with little to no thought—for a strategic plan. According to Jacka and Scott (2011), this lack of a plan opens a company up to even more risk. While other areas of the organization may have a strategic charter in place, very often social media is ignored. Social media, as an effort requiring a strategic plan, may receive little attention because it crept quietly into the corporate ecosystem. It may also be under the strategic radar because of how the company perceives social media. If upper management views social media as the place where people say too much and share cat photos, then the corporate culture won't be willing to invest in it as a strategic tool.

Strategic communication and the technical communicator go hand-in-hand. Writing a strategic social media plan to align with the company's overall goals would be an ideal task for a professional technical communicator. Strategies are key to moving a company forward—they're a plan for success. And social media strategies don't operate in isolation. A savvy technical communicator will know to align the social media plan with other organizational strategic documents and build a body of knowledge.

What you can do: Is there a strategic plan at your workplace for social media? If there isn't, locate the strategic planning documents and begin a discussion on how social media can become part of the plan across the entire organization.

A LOOK AT THE LAW

Does every question in your organization need a response from legal counsel? More than likely, the answer is no. Yet, who in your company is the most informed about general legal and regulatory issues based on their day-to-day work? Once again, it's you, the technical communicator, who could be the informal social media para-legal point person. In fact, this is such an interesting area, you'll be tempted to learn even more.

Social media seems like the Wild West of communication—anything goes. While that's partly true in our personal uses, it's far from the case for businesses. There are significant legal and regulatory repercussions in the social media world. Once you put information out into the social media sphere, it's there forever and it can spread like wildfire. Thus, it's a good idea to make sure social media communications are correct, legal, and follow regulatory mandates.

As Robert McHale (2012) notes, navigating the legal risks is more challenging every day. For McHale, accidental disclosures of data, deceptive product endorsements, and labor law violations are just some of the issues amplified by social media. Depending on the industry, countless regulatory and compliance issues may also be at stake.

However, technical communicators have been writing procedures and instructions while maintaining a close eye on the legal and ethical issues. Who better to act as the point person?

What you can do: Is there a guide for your company on legal/regulatory issues and social media communication? Taking this often complex material and translating it to actionable items would be right up a technical communicator's alley.

MANAGING RISK ONLINE

What's the biggest risk your company has? Can you name it? Is it what you'd think? More than likely, your company's biggest area of risk is, according to Jacka and Scott (2011), its online reputation. Good policies, a sound social media strategy, and an understanding of the legal and regulatory issues can help any company minimize risk, but social media in itself is a risk. Social media channels can amplify the smallest corporate issue in ways we never imagined and shake a company's reputation.

While PR takes on crisis communications as their frontline job, staffing reductions may mean those positions are already over-worked. However, as the communication professions work more closely together, the technical communicator, as a person with excellent technical and research skills, can do regular reputation management reviews and help PR/Marketing focus its efforts.

What you can do: Is online reputation being monitored for your company? If it is monitored, examine the reports to see where policies need updating or creating. If it's not monitored, step up and offer to be the person who conducts social listening and reports on online risk.

WHAT'S OUR STYLE?

Remember that style guide you created a couple of years ago—the one that governs spelling and acronyms for your organization? Have you updated it for your company's social media efforts? Better yet, have you extended this document so it explains the brand voice and approach your organization will have in the social media sphere? If you're a technical communicator, and the answer to the above is NO!, then you've got a job to do.

Part of making social media effective across the organization requires, according to Jim Belosic (2012), solidifying your company's online identity and character. It's important that the company's representatives speak with one voice. The nature of that voice depends entirely on your organization. Some companies can channel a lighter persona in their communications and use current trendy phrases and colloquial turns. But that may not be right on every channel, and it may not be right at all for your company.

Assessing the voice of the company and determining if it's suitable across all social media channels is perfect for technical communicators. Audience analysis—the foundation of technical communication practices—means you're prepared to learn about what's happening on social media.

What you can do: Visit your company's social media channels and learn about the audience. See how they communicate with each other; work with marketing to help refine your style guide so everyone understands how your company speaks in the social sphere.

BRINGING IT ALL TOGETHER

For most technical communicators, the social media frontier seems a distant horizon. Yet, by becoming the internal-facing social media expert for your organization, you can resituate the value of a technical communicator in an organization, and forge a new and very important path for the discipline. Technical communicators are the perfect choice to connect various departments in an organization and be at the forefront of an integrated social media model.

RESOURCES

Flynn, Nancy. The Social Media Handbook. San Francisco: Pfeiffer, 2012. Print

Peterson, Chris. "The First Step Is to Understand Your Audience." 2012. Web. 20 Mar. 2012.

Scott, Peter R., and J. Mike Jacka. Auditing Social Media: A Governance and Risk Guide. Hoboken: John Wiley & Sons, Inc., 2011. Print.

REFERENCES

Belosic, Jim. "8 Tips for Training Social Media Marketers." Social Media Examiner. 2012. Web. 29 Mar. 2013.

Cain, Allison. "The Social Media Scene." Internal Auditor Aug. 2012 : 45–49. Print.

Ganim Barnes, Nora, and Ava M. Lescault. "The 2012 Inc . 500 Social Media Update : Blogging Declines As Newer Tools Rule." Center for Marketing Research. 2012. Web. 22 Mar. 2013.

Jacka, J. Michael, and Peter R. Scott. "The Whole World's Talking." Internal Auditor 2011: 52–56. Print.

McHale, Robert. Navigating Social Media Legal Risks. Indianapolis: Que, 2012. Print.

Kelly, Nichole. How to Measure Social Media. Indianapolis: Que, 2013. Print.

AUTHOR CONTACT INFORMATION

Dr. Laura A. Palmer
Assistant Professor—Information Design & Communication
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060
lpalmer2@spsu.edu

AUTHOR BIOGRAPHY

Dr. Laura Palmer is an assistant professor of Information Design & Communication at Southern Polytechnic State University. In her academic pursuits, Dr. Palmer's research interests center on people and their uses of technology. She is currently researching social media as it relates to professional communication and public discourse. Her other recent work has explored how social media functions as a site of engagement in higher education; she has presented her social media work at several recent conferences. As a faculty member, Dr. Palmer teaches courses at both the graduate and undergraduate levels. Those courses include *Information Architecture*, *Content Strategy*, *Social Media Integration*, and *Writing Across Media*. She is also the Senior Research Associate at the Usability Center at SPSU. Recently, Dr. Palmer became part of SPSU's new Center for Digital Engagement and its Social Media Outreach division. Dr. Palmer holds a Ph.D. in Technical Communication & Rhetoric with a minor in Visual Theory.

USER EXPERIENCE AND USABILITY

[Addicted to Meaning: Mental Models for Technical Communicators](#) page 167

[Formalizing the Technical Communication and User Experience Relationship](#) page 173

[Gamification is Here: Build a Winning Plan!](#) page 179

[Useful, Saleable, and Buildable: The Role of UX](#) page 189

ADDICTED TO MEANING: MENTAL MODELS FOR TECHNICAL COMMUNICATORS

Kai Weber

This paper explores how “meaning” works in technical communication, why it fails, and how you can create meaningful documentation. Drawing on the cognitive psychology of mental models and user experience design, Kai Weber shows with examples why minimalism works but FAQs don’t, and how to write for users without irritating them.

Being meaningful is an essential, but elusive feature of good technical communication. It is just as essential as being correct and clear, concise or consistent: Understanding how and why communication is meaningful can help you to make your documentation more effective and to make your product more useful.

WHAT IS MEANING?

Information theory posits a hierarchy of information which proceeds from data at the bottom via information and knowledge to wisdom at the top (Weinberger 1-5). For example, data is the sheer fact that the Microsoft Office 2007 software has an "Office button" icon in the upper left corner. The information that this icon gives you access to functions such as opening, saving, and printing a file helps you with generic functions. The knowledge that this functionality has replaced the File menu adds meaning and supports your active experience.

So meaning occurs at the knowledge stage in this hierarchy when you make sense of data and information, when you "connect the dots" into something that you can apply purposefully. Meaning gives answers to questions such as "So what?", "What does this mean for me, in my situation?" and "Why should I care?" (Cortada 4).

WHY SHOULD TECHNICAL COMMUNICATORS CARE?

Technical communicators should care about meaning, because we are in the business of creating meaning and transmitting it to users. We can provide all the data and information we want, if it is not meaningful to customers, it is wasted and dead. Any time documentation fails, it is either because meaning has not been created or not been transmitted successfully. Documentation that merely informs the user "To print a file, click the Print button" does not support the user in any active experience. It does not create any meaning if it omits the context, such as the task the user may be engaged in, the prerequisites and the expected results of the user's action.

HOW DOES MEANING GET TRANSMITTED IN COMMUNICATION?

The Mathematical Theory of Communications by Shannon and Weaver of 1949 explains communication as the transmission of messages that are broken up into small signals (see Fiske, ch. 1). A sender (a technical communicator) uses media (HTML files) via channels (the internet) to transmit messages to receivers (users). This model aims to show what keeps communication efficient and accurate.

This technical model is clear, linear, and easy to understand. However, communication doesn't really work this way beyond the very technical levels of a telephone cable, a radio wave, or a TCP/IP network. It's only in the brain where "information is born – it's not in the beeps" (Heinz von Foerster, quoted in Gleick 417). In fact, as Shannon admitted, the question of meaning is "irrelevant to the engineering problem" of communication that his theory explains (Gleick 416).

Semiotics is a second theory of communication (see Fiske, ch. 4). It explains communication as the production of meaning between people who exchange messages in cultural contexts. The aim of this theory is to ensure that communication is meaningful for the reader. Note that the reader is not a passive recipient of the message, as in Shannon and Weaver's model, but a participant who is equally active as the author of the message.

When creating messages, we use different types of signs, such as words and images, which have different ways to represent and refer to objects in the world around us. Words in semiotic terms are symbols. They refer to things by sheer convention not by direct similarity: The word "chair" neither looks nor sounds like a chair. Images, such as photographs and maps, in semiotic terms are icons. They represent things by similarity. Photos are usually pretty close two-dimensional representations, while maps are more abstract and often use abstractions and words to refer to things. To combine such signs into messages, we apply codes, such as dialect and jargon. Codes are systems of rules which most participants share, more or less. So codes form a social practice of communication conventions that apply in specific cultural context.

In this semiotic model, readers (not writers!) construct the meaning of a message. That is not to say that the meaning is whatever the reader wants. Rather, the meaning is how the reader interprets the signs according to the code convention. For example, a user will assume that a user manual contains photos or schematic images that represent the layout of the product and try to create meaning accordingly.

The advantage of semiotics over the earlier model for my purpose is that semiotics focuses on exchanging meaning between people. Because communicating meaning is more ambiguous than ensuring the technical accuracy of communication, semiotics is more complex than Shannon and Weaver's theory, due to all the moving parts: Signs, such as words or even images, can lend themselves to creating different meanings at different times or for different groups of readers. (Think of ideologically, religiously, or politically charged words or images.) A code such as your regional dialect can make you feel a member of the group or it can exclude you as an academic or professional jargon might do. Whether you feel included or excluded influences your understanding of the message and the meaning you take from it.

However, technical communication suffers less from these shifts and ambiguities, because a lot of our signs and codes are restricted and well-defined. Or are they?

WHY MEANING IN TECHNICAL COMMUNICATION IS EASY AND STILL FAILS

Technical communication can avoid many of the semiotic pitfalls because it applies a "narrowcast code" (Fiske 76-77). A narrowcast code addresses a limited, more or less homogenous audience that shares cultural background and experiences and has a general willingness to learn the applied code of descriptions and instructions such as a user manual contains it. (Compare this with the opposite, the broadcast code, which addresses a wide and diverse audience, sometimes called the "masses", with messages of the lowest common denominator that seek to affirm knowledge and prejudices rather than instruct and teach something new.)

All the restrictions of a narrowcast code benefit technical communication: The audience of a software manual shares a common background of computer savvy. The manual can presuppose a certain vocabulary or define

it in its glossary. The context is confined to the intended tasks of the software, and the expected outcomes of the tasks are clear.

With all these major restrictions on communication in place, the user's path is confined to a tunnel, and all that is left for the manual to do is to put in some lights and a railing to safely guide the readers through.

If it was that easy, we wouldn't find meaning in technical communication so elusive – and we wouldn't despair to get users to read the manual.

So even semiotics fall short by underestimating the most dynamic part of the equation, the reader.

My next witness is Heinz von Foerster, who above preferred the brain to the beeps as the seat of information. His theory of radical constructivism holds that there is no meaning but the one created by the reader; meaning derives from the reader's ability to deduce something useful from a text (von Foerster 21, 69). This extreme position becomes more plausible when we consider context as a powerful influence on meaning: While we build on past experiences and what we have learned from them, it is ultimately our specific, current situation that drives our understanding and shapes the meaning we create. In this sense, each situation is a new beginning (von Foerster 27).

Different experiences with technical communication can illustrate this point: An advanced topic in a manual about customizing a complex setup will be incomprehensible and meaningless to a novice user – but it still can be useful to the same user after initial training and some experience. The primacy of the individual situation also explains why FAQs (Frequently Asked Questions) frequently do not work: Because they are not frequently asked by actual users, unless they concern actual, real situations.

To paraphrase Heinz von Foerster and Fred Dretske, both philosophers who think about the nature of knowledge: "Beauty is in the eye of the beholder, and meaning is in the head of the receiver." (Gleick 417).

HOW DO READERS CREATE MEANING?

So how do our readers create meaning in their heads? In short: By bouncing off their "mental models" against the world. A mental model is a mental representation of the world, of our place in it, and of how stuff works. To make it efficient and maintainable in our minds, mental models are semi-consciously selected and incomplete. (Else the information overload would paralyze us.) Our mental models are limited to what we think we understand and what we think we can or want to do.

Most of us have a mental model of a restaurant: We understand it's a place where we get fed in exchange for money. We can go there, be seated, be served, eat, pay, and leave. But our mental model is defeated at a self-service buffet restaurant where we are neither seated nor served. To prevent this defeat, such restaurants put up signs not required at more conventional places: "Take tray here" and "Pay here". These signs help us to adjust our mental model, along with our range of options to act and behave. For further adjustment of our mental model, the restaurant informs us on a leaflet that they can also cater our wedding. Alas, because each situation is new and adaptations of mental models are often fleeting, we forget this information soon, because it is not meaningful for us – and someone else caters our wedding.

Mental models have several strengths: They are flexible and teachable (as many functions of the brain are). As a result, we can add self-service buffets to our model of restaurants after we've visited one – or even heard about one from a friend. They are a springboard to meaningful knowledge by helping us learn. In mental models, we "connect the dots" and remember successful experiences. What worked yesterday, in a self-service buffet or with a new product you operate, shapes how you approach questions and tasks tomorrow. Mental models help us decide what we try to solve a problem – and also where and how we look for help.

Mental models are also a major foundation in the fields of human-computer interaction and user experience design. Hence we technical communicators can learn from usability expert Jakob Nielsen's explanation: "A mental model is *what the user believes* about the system at hand" which "impacts how they use it" (Nielsen, his emphasis).

Nielsen gives several examples of mixed-up mental models concerning software which illustrate how our beliefs or mistaken knowledge lead us to deduce that a software product or website is not working: Some users confuse operating system windows, such as Microsoft's File Explorer, with browser windows in Microsoft's Internet Explorer and expect both to copy, move, or delete files identically. Other users confuse icons and applications – hence the warning that deleting an application icon from the desktop does not actually remove the application itself. (Of course, many of these confusions can and should be solved in better user experience design, but that is not the focus of this paper.)

From Nielsen's work, technical communicators can learn two relevant limits of mental models: They are inert, despite all flexibility. "Stuff that people know well tends to stick, even when it's not helpful" (Nielsen). Hence, the Save function still uses the icon of a diskette which fewer and fewer people have actually seen or used. And hence, many technical communicators are still forced to create deliverables in PDF or CHM, even though superior, more efficient formats and tools have been available for a long time. The inertia points to a more fundamental limit and to the real reason why technical communication so often fails: Mental models are ultimately uncontrollable!

With technical communication, we can at best hope to guide users towards appropriate behavior. This may or may not extend their mental models, it may or may not anchor a meaningful experience to which they resort later when they face a similar problem.

This has been a rather bleak turn of events. Given the inertia of uncontrollable mental models, how do we manage to communicate anything?

ADDICTED TO MEANING

The answer lies in a second aspect of meaning itself: Yes, meaning depends on shifting signs, on contextual codes, and on our erstwhile experiences. But we are also addicted to meaning in the sense that we desire and hope for a generally meaningful life. This is the motivational force behind our questions for meaning: "What does it mean for me?" and "Why should I care?"

This aspect is easy to recognize in "big" questions for the meaning of life, but it also permeates small everyday tasks – the kind we address in technical communication. We *want* to connect the dots, we *want* to make sense, even of the software we use, and it frustrates us to the point of personal insult when we cannot, because the stupid thing simply won't work!

Even in less consequential matters than the correct use of appliances and software, we go to great lengths to discover meaning in the world – or to create it as what we believe we know. Misheard lyrics of pop songs are a harmless, fun example. Jimi Hendrix didn't seem to make a lot of sense singing in "Purple Haze": "Excuse me while I kiss the sky." Instead, many people heard: "Excuse me while I kiss this guy." (Look up "Mondegreen" on Wikipedia for more fun examples.)

Or consider the first well-known version of the Apple logo, the apple with a bite in it in horizontal colored stripes (see Konnikova). What does it mean? It's Adam and Eve in paradise and their quest for knowledge manifest! Or no: It's a reference to Isaac Newton sitting beneath an apple tree and discovering the law of gravity! Or no: It's a tribute to Alan Turing, computer science pioneer, who committed suicide by biting into a cyanide-laced apple! (Turing was gay, hence the rainbow-colored stripes).

The point is that one explanation after another is as meaningful as any urban legend can hope to be – but they are all not the meaning that was originally intended. The truth is a lot more mundane: The apple is the company's namesake. The rainbow denotes color in general, because the Apple II was the first home computer to display color images. And the bite was added to indicate scale, so people didn't confuse the fruit with a cherry. Or so says Rob Janoff, the logo's designer (see Raszl).

So we are addicted to meaning. We seek to connect the dots in any plausible way to come up with a meaning and a solution. But when things get weird, technical communication often fails to deliver a satisfying answer. Our formulation of the problem doesn't match any of answers which the manual provides.

Then we turn away from the manual – but our quest for meaning is not finished yet. We tap other resources: We ask a friend who may not have the answer, either. But compared to the documentation, a friend is much more likely to provide a meaningful experience. One friend may remember having had a similar problem and who knew a solution. Another may have discovered a workaround. Someone may ask in return what we're working on, anyway – and inform us that this task has just become obsolete.

The point is that there are many more meaningful scenarios to resolve a task or conflict than most common documentation offers. And while we cannot address every possible outcome, we can do better and meet the users and their mental models halfway by creating more meaningful documentation.

HOW CAN WE CREATE MORE MEANINGFUL DOCUMENTATION?

To apply a designer's tenet by Jakob Nielsen to technical communication, it is a key objective for technical communicators to make the documentation "communicate the system's basic nature well enough that users form reasonably accurate (and thus useful) mental models".

This goal leaves us basically two strategies. We can try adjusting the users' mental models which seems the natural choice: Users may know similar products, but to really learn about this particular version, they need our specific manual. They should use our manual to adapt their existing mental models to accommodate our product and its correct use. But that would leave us with the reasons why technical communication still fails...

The better alternative is to adjust the documentation as much as possible to the users' existing mental models – which is basically saying: "Know your audience", but by now I've shown why we need to know them and how they create meaning.

What does that mean specifically? Applying lessons from user experience design, we can:

- Serve the inertia of mental models: Be consistent and conservative in your writing. Don't invent new terminology unless absolutely necessary. Stick to the conventions and habits of your users.
- Anticipate the users' roles and tasks in mental models: Frame documentation in terms of personas and tasks which users can identify with. In an accounting system, offer navigational paths for personas such as accountants and auditors and for tasks such as daily accounting and end-of-year accounting, but menu by menu.

Another area that takes users' mental models seriously is minimalism: Minimalist technical communication, as described by Hans van der Meij and John M. Carroll, means much more than just presenting the bare necessities. It takes "the need for meaningful activity and sense making" seriously by providing users with "an immediate opportunity to act" (van der Meij 19) and by relating descriptions and instructions to the users' actual tasks. Van der Meij and Carroll offer the following principles and guidelines, among others, to ensure that documentation meets users on their turf (see van der Meij 21):

- Choose an action-oriented approach.

- Provide an immediate opportunity to act.
- Respect the integrity of the user's activity.
- Anchor the tool (i.e., the product and its description) in the (user's) task domain.

Taking these leads from user experience design and minimalism, we can offer users dots which are easy and natural to connect and derive meaning from.

REFERENCES

- Cortada, James W. Information and the Modern Corporation (Cambridge, MA: MIT Press), 2011.*
- Fiske, John. Introduction to Communication Studies, 2nd ed. (New York, NY: Routledge), 1990.*
- Gleick, James. The Information: A History, a Theory, a Flood (London: Fourth Estate), 2011.*
- Johnson-Laird, Philip N. "Mental Models, Deductive Reasoning, and the Brain." The Cognitive Neurosciences. Ed. Michael S. Gazzaniga. (Cambridge, MA: MIT Press), 1995: 999-1008.*
- Johnson-Laird, Philip N. "The history of mental models." Psychology of Reasoning: Theoretical and Historical Perspectives. Ed. K. Manktelow and M.C. Chung. (New York, NY: Psychology Press), 2005: 179-212.*
- Konnikova, Maria. "Hunters of Myths: Why Our Brains Love Origins." Scientific American (7 April 2012). <http://blogs.scientificamerican.com/literally-psyched/2012/04/07/hunters-of-myths-why-our-brains-love-origins/>*
- Nielsen, Jakob. "Mental models". Nielsen Norman Group (18 October 2010). <http://www.nngroup.com/articles/mental-models/>*
- Raszl, Ivan. "Interview with Rob Janoff, designer of the Apple logo." Creative Bits. (3 August 2009). http://creativebits.org/interview/interview_rob_janoff_designer_apple_logo*
- van der Meij, Hans and John M. Carroll. "Principles and Heuristics for Designing Minimalist Instruction." Minimalism Beyond the Nurnberg Funnel. Ed. John M. Carroll. (Cambridge, MA: MIT Press), 1998: 19-53.*
- von Foerster, Heinz. Der Anfang von Himmel und Erde hat keinen Namen. (Berlin: Kadmos), 2008.*
- Weinberger, David. Too Big to Know. (New York, NY: Basic Books), 2011.*
- Winograd, Terry and Fernando Flores. Understanding Computers and Cognition: A New Foundation for Design. (Reading, MA: Addison-Wesley), 1989.*

AUTHOR CONTACT INFORMATION

Kai Weber
 Senior Technical Writer
 SimCorp GmbH
<http://kaiweber.wordpress.com/>
[@techwriterkai](#)

AUTHOR BIOGRAPHY

Kai Weber gets a kick out of making users look good thanks to technical communication. More formally speaking, he's a Senior Technical Writer for SimCorp in Frankfurt and Copenhagen. Kai has been writing, editing and translating software documentation since 1988. After receiving an M.A. in American Studies, Kai went full-time as a technical writer in 1999 and has focused on financial and banking systems since 2001. He enjoys coaching and training other writers and speaking at conferences, such as the STC Summit, TCUK, and tcworld.

FORMALIZING THE TECHNICAL COMMUNICATION AND USER EXPERIENCE RELATIONSHIP

Lori Fisher

We must move beyond collaborative synergy and matrixed teaming of Technical Communication and User Experience to a more formalized relationship, where dedicated User Experience skill becomes required staffing for Technical Communication projects. The user experience must drive the optimal information experience and the required information architecture and design.

IS COLLABORATION ENOUGH?

For many years, we have talked about the synergy of technical communication (TC) and user experience (UX) skills, and the collaboration needed across such teams. On smaller and less mature projects without dedicated UX staffing, technical communicators often take a lead role in shaping the information experience, generally focused on the usability of the information. On larger teams, one or more of the TC team might have training in user experience or there might be a dedicated UX professional on the development team. In these cases, there is often collaboration to include an assessment of the usability of the information as part of the information development process.

It is time to move beyond synergy and matrixed teaming to a more formalized relationship, where dedicated UX skill becomes required staffing for any TC project, and further, where size and scope allows, where UX formally leads the information design effort. It is time to move beyond validating the usability of the information to a model where user experience drives the up-front information design. Putting the user experience first will drive a focus on the optimal information experience and the required information architecture and design to support that user experience.

MOVING TOWARD THE NEW MODEL

The new model integrates user experience and technical communication skills on one team with user experience leading the design phase. To move toward this model on smaller teams, this may mean recruiting a single individual with both TC and UX as part of their formal training. Some university programs today, such as University of Washington's Human Centered Design and Engineering program (among others) provide academic training that crosses both fields. Hiring managers must begin to look for individual contributors and contractors with this profile.

Larger technical communication teams can form integrated multidisciplinary teams, where a portion of the TC resource is allocated to staffing formal UX skills. Well-designed and well-written content is still imperative and the need for content development skills does not go away. But content must be delivered in the context of the overall user experience.

How does a manager or team lead justify bringing in user experience skill to a writing team, instead of a technical communicator? Simpler products require less documentation - simpler tasks, simpler user interfaces, and simpler navigation reduce the need for complex documentation. Automation of tasks can also reduce the need for documentation. Strategic information architectures such as progressive disclosure, specifically embedded assistance, require tight integration of ID and UX teams to implement. The goal of "walk up and use" for software products will be a result of building information and supporting affordances into

the UI, again requiring that tight partnership of TC and UX. For many tasks, the information experience is still a key component of the overall user experience – especially for up & running, getting started, and many other tasks. The technical communicators' work does not go away, but it must be designed in the context of the user experience.

If the company has one or more UX professionals in a separate organization or department, the Technical Communication team should seek, at a minimum, to formalize the collaboration of UX and TC specific to the Information Experience. For example, user testing should include scenarios that exercise the information, and customer and user group interactions should include feedback on the information.

However, in the latter example, there is opportunity in such a company to take the next step and fully implement the new model, by proposing that UX and TC be formally combined, organizationally, within the reporting structures. This is a big step and may seem threatening to some technical communication managers. But if we see the two skill sets, UX and TC, as complementary and both aligned around the user experience, then we must move forward to establish a formal relationship of the two. Each skill set still has its own roles and work to do. There is no “work elimination” suggested here. Content continues to be critical to the user experience. But making UX responsible for the design of the information experience will only happen if the teams are organizationally connected in a reporting structure, or at least a structure that formally defines a relationship between UX and TC that does not depend on volunteer teaming.

CASE STUDY

A recent reorganization at IBM provides a case study of many of the principles described above. One division within IBM reorganized the reporting structure from matrixed collaboration of TC and UX to a formal relationship between Technical Communication and User Experience. Originally, individual TC and UX departments reported to the same upline Director, but in parallel structures. Collaboration and focus on the information experience was encouraged as a part of matrixed teaming, but the effectiveness depended on the initiative and outreach of the TC teams to engage the UX skills as needed. A primary focus of that collaboration was validating user satisfaction with the information design, both before and during iterative implementation. While progressive in the sense of engaging the user in the information process, the focus was on a satisfactory information experience as the goal (in contrast to designing the ideal user experience that included the information to support that experience).

Recently, the organizational structure has been changed to merge the TC and UX skills for a specific product area into one combined team, with a UX manager leading the group. The result is an increased focus on defining the user experience first, then identifying the information design and architecture that best support that experience, rather than designing the information from an "information first" perspective and then validating the usability or ease-of-use of that information experience.

One of the first projects delivered by the new integrated team was a radically new approach to information and interface design which dramatically improved the clarity of tasks and task sequences for customers, and also drove the design of interaction points in the product itself. These information deliverables are being referred to as “interactive conceptual models,” and are both an interactive user interface but also a way of conveying Getting Started information formerly published in text format in the Information Centers.

New users of a product often struggle to piece together the big picture of capability and don't know how to complete key tasks. The design of the interactive conceptual models began with a user scenario for getting started with a particular product, and a challenge to the integrated team to dramatically improve the initial experience, to “jump start” the user on key tasks. In a traditional TC model, the technical communicators

would have worked on providing additional content for key tasks, or documenting new “Getting Started” tasks. The UX team might have run usability testing on scenarios that tested the new content.

Instead, the new integrated team spent a few weeks exploring various graphical, interface, and text approaches to the problem. They then iteratively designed a set of interactive models to enable users to visually explore concepts and procedures to more quickly gain a greater understanding of complex tasks and complete key tasks. Users provided interim feedback that shaped the following design iterations. The “information” conveyed by the models had generally never been documented, and the dynamic interactive aspect of the models would have been nearly impossible to recreate with text. Instead of approaching the problem as an information problem, the team approached this as a user experience problem and designed an interactive informative experience to address it.

The new information design dramatically improved understanding of Getting Started tasks by the customers, and was validated in user testing as more effective than traditional information. For example, one key technical sales leader said this new approach was “the best documentation inside or outside of IBM.” In usability testing with 35 customers at a key customer conference, 100% of the customers rated it “better” or “much better” than the documentation provided today by other vendors. 98% stated it was “likely” or “very likely” to help them achieve a task. Additional quotes included “The interactive documentation is very useful” and “If we have this chart, even my 9 year old son can do it.”

In addition to the work on the interactive conceptual models, we have seen similar success on other projects and areas of focus by the integrated teams. Examples of focus areas include:

- **Up & Running:**

A TC/UX team followed up on key findings from user research to provide the types of set up and installation information that better matched user needs. UX led an effort (including design and code) to include a deployment topology as part of the installer itself. The team jointly created a Quick Start Guide and installer planner. These collaborative, multi-pronged efforts ensured users could successfully get up and running more quickly than in the past.

- **Task Launchers:**

An innovative task aid called the Task Launcher was created by a team of a Senior UX and a Senior TC lead, to address up and running across platforms by helping users find key tasks organized by application lifecycle phases. The Task Launcher has been delivered in eight products.

- **Embedded Assistance:**

Our integrated TC/UX teams have made significant strides in architecting embedded assistance into the product interfaces (instead of documenting help or Information Center topics) across the portfolio of products in our division. The joint team also created a DITA XML customization for coding embedded assistance. TC and UX have collaborated to create embedded assistance in 51 products across one division in a single year, doubling from the prior year.

SUMMARY

Technical communicators will be most successful at providing excellent information when the information is designed and planned as part of the user experience. The information experience (IX) is a subset of the overall user experience (UX) and must be designed in that context.

- The information experience must first be designed, and that experience then drives information design and architecture.

- As we move to embedded assistance and other forms of progressive disclosure tightly integrated into the user experience, formal UX skills become a critical element of the success for the technical communication team.
- Formal training in HCI with a focus on user research and interaction design must become a required skill set for each TC team
- Well-designed and well-written content is still imperative and the need for content development skills does not go away. But content must be delivered in the context of the overall user experience.
- For larger teams and organizations, a formal structure and staffing can ensure the UX/IX relationship is effective.

RESOURCES

Halvorson, Kristina. "The Discipline of Content Strategy." www.alistapart.com, No. 274 (December 16, 2008), and http://www.ue.com/articles/discipline_content_strategy/ republished May 26, 2010.
<http://www.alistapart.com/articles/thedisciplineofcontentstrategy/>

A classic: Mayhew, Deborah J. "User experience design: The evolution of a multi-disciplinary approach." *Journal of Usability Studies* 3.3 (May 2008): 99-102.

Redish, Janice and Carol Barnum. "Overlap, Influence, Intertwining: The Interplay of UX and Technical Communication." *Journal of Usability Studies* 6.1 (May 2011): 90-101.

REFERENCES

Interactive conceptual models:

<http://pic.dhe.ibm.com/infocenter/bigins/v2r0/topic/com.ibm.svg.im.infosphere.biginsights.product.doc/doc/c0057605.html>

http://pic.dhe.ibm.com/infocenter/bigins/v2r0/topic/com.ibm.svg.im.infosphere.biginsights.product.doc/doc/acc_mda.html

http://pic.dhe.ibm.com/infocenter/bigins/v2r0/topic/com.ibm.svg.im.infosphere.bigdata.sda.doc/doc/acc_sda.html

<http://pic.dhe.ibm.com/infocenter/bigins/v2r0/topic/com.ibm.svg.im.infosphere.biginsights.analyze.doc/doc/c0057518.html>

Getting Started with IBM BigInsights video:

<http://youtu.be/59NmBXNlcoc>

AUTHOR CONTACT INFORMATION

Lori Fisher
 Director, Information Management User Technology
 IBM
 555 Bailey Avenue
 San Jose, CA 95141
 408.463.3573

AUTHOR BIOGRAPHY

Lori Fisher is Director of Information Management User Technology at IBM's Silicon Valley Laboratory, leading a multi-disciplinary organization with worldwide responsibility for user experience, information development, globalization, and accessibility for the information management product portfolio in IBM's Software Group. She is a leader in the corporate-wide Information Development Advisory Council and co-chair of the IBM-wide User Experience Executive Council. She developed and taught two of the core courses in a certificate program in Advanced Technical Communication at University of California Extension for almost two decades. She has served on the STC Nominating Committee, chaired the STC Quality SIG, held multiple elected positions on the local Silicon Valley STC chapter Administrative Council, and judged in various STC competitions. She served a 2-year term as Secretary of STC on the international STC Board of Directors. She is a Fellow of STC in the Silicon Valley Chapter.

GAMIFICATION IS HERE: BUILD A WINNING PLAN!

Marta Rauch

Enterprise gamification is growing at an impressive rate. Market research predicts that by 2014, over 70% of companies will have at least one gamified product, and by 2015, over 50% will gamify innovation. (Gartner Press Release, April 12, 2011: <http://www.gartner.com/it/page.jsp?id=1629214>) This rapid change brings opportunities for technical communicators who understand gamification and can create gamified user assistance. This article describes enterprise gamification, provides examples, includes resources for further learning, and suggests strategies for effective gamified user experience.

WHAT IS GAMIFICATION?

Gamification uses game techniques in non-game situations to motivate people and drive behavior. Game techniques are often grouped in these three categories:

- **Game dynamics** - motivate behavior. Examples include scenarios and rules.
- **Game mechanics** - help users achieve goals. Examples include teams, competitions, rewards, and feedback.
- **Game components** - track users' progress. Examples include quests, points, levels, badges, and collections.

When implemented effectively to solve business needs, these techniques can make enterprise tasks as fun and enjoyable as games.

EFFECTIVE FOR SOCIAL MEDIA

Gamification effectively motivates people in social media. For example, *foursquare* encourages frequent checkins at fun locations, and Klout users check in daily to check scores and review leaderboards.

PROVIDES RESULTS FOR EDUCATION

Gamification also shows excellent results for education. For example, Khan Academy provides millions of worldwide students access to effective online instruction. Partially due to the effective gamified interface, it is even being incorporated into public school curriculum.



Education Example: Khan Academy (<http://www.khanacademy.org>)

ADOPTED BY THE U.S. GOVERNMENT

The United States government also recognizes gamification's benefits. According to the webpage of the Office of Science and Technology Policy (<http://www.whitehouse.gov/blog/2011/11/23/games-grand-challenges>), twenty-three U.S. agencies are incorporating gamification, including the National Institutes of Health, Department of Energy, US Army, and NASA.



Example: United States Government

BENEFITS TO ENTERPRISE

As a recent Forbes article pointed out, gamification has become “a CEO’s best friend.” (<http://www.forbes.com/sites/danwoods/2012/05/14/gamification-grows-up-to-become-a-ceos-best-friend/>)
When gamification is implemented effectively, enterprises can:

- Increase user engagement
- Motivate customer participation
- Influence behavior
- Drive adoption, learning, loyalty
- Improve quality of service
- Increase efficiency
- Reduce time, costs
- Drive profits

IT’S BIG MONEY!

The market for gamification has enjoyed dramatic growth, surging from 155% in 2011 to 197% in 2012. Market research firms expect the market to jump from \$100M in 2011 to over \$2.8B in 2016. With 47% of implementations currently focused on user engagement, enterprise is the largest market segment, consisting of 25% of the gamification market. (Source: M2: <http://www.m2research.com/gamification.htm>)

ENGAGING MILLENNIAL EMPLOYEES

Millennial workers, also called “Gen Y,” are currently 25% of the US workforce. In 2014, this number will grow to 36%, and, in 2020, to 46%. ([“Maximizing Millennial” UNC Kenan-Flagler Business School](#))

According to game designer and industry expert Jane McGonigal, a unique trait of millennial workers is that they have 10,000 hours of gaming by age 21. (<http://www.youtube.com/watch?v=dE1DuBesGYM>) This amount of experience qualifies Gen Y as expert gamers, based on the “10,000-Hour Rule” described in Malcolm Gladwell’s *Outliers*.

Due to this expertise, this generation is most engaged when using game-like user interfaces. To motivate this workforce, employers must provide attractive enterprise gamification solutions for corporate innovation and internal processes.

ENTERPRISE GAMIFICATION EXAMPLES

These examples highlight a variety of enterprise gamification products and solutions.

ORACLE

Several enterprise gamification initiatives are spotlighted in the following sections. (Note: The statements and opinions expressed here are the author’s own, and do not necessarily represent those of Oracle Corporation.)

THE IMPORTANCE OF GAMIFICATION

At Badgeville’s Engage 2012 industry event, Nick Giannasi, VP of Life Sciences Product Strategy, Oracle Health Sciences, spoke to attendees about the importance of gamification.

(<http://badgeville.com/resources/events/engage2012/oracle-on-importance-of-gamification>)



Badgeville Engage 2012

ONLINE FORUMS

The Oracle Forum (<https://forums.oracle.com>) is an effectively gamified community. Community members are encouraged to answer questions posed on the forum, and are given points for providing helpful answers. Top contributors are recognized on a leaderboard, with achievements shown by points and badges. The result is that all types of participants are motivated to contribute to the community.

GAMIFICATION FOCUS GROUP

To gain understanding of what motivates users to participate in Oracle’s online communities, the Oracle Applications User Experience group conducted a focus group to gather input from community members.

Through guided discussions, they collected ideas and opinions on motivation, and received feedback on how game mechanics could encourage participants.

ORACLE APPS UX GAMIFICATION ALL HANDS DAY

The Oracle Applications User Experience (UX) team held a worldwide design jam, in which groups held a friendly, gamified competition to create effective new types of gamified user interfaces. Participants agreed that the event was a resounding success. According to Ultan O’Broin, it was “a great way to learn about gamification, build team spirit, and create an innovative, contemporary user experience in a very agile way.” (https://blogs.oracle.com/userassistance/entry/oracle_applications_gamification_worldwide_ux)

SOCIAL NETWORKS

To share information about gamification activities, employees use Twitter handles and a designated hashtag: @GamifyOracle #GamifyOracle. Employees also post a gamification blog at <https://blogs.oracle.com/gamification/>.

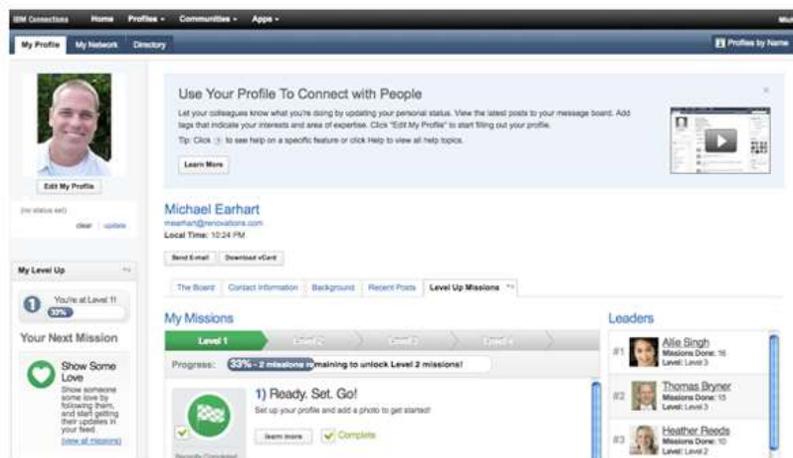
KNOWLEDGE SHARING

Initiatives such as the ones described here provide valuable opportunities to learn and develop gamification techniques. Oracle employees share their learning about gamification at conferences such as STC Summit, LavaCon, Intelligent Content, Gamification Summit, Enterprise Gamification, ACM CHI, and Human-Computer Interaction International (HCII).

IBM

IBM’s Level Up for IBM Connections incorporates gamified strategies such as onboarding tasks that bring users quickly into the product. Gamification components include missions, levels, points, badges, and a leaderboard. (<http://www.youtube.com/watch?v=rro5G0yLyeE&feature=relmfu>)

As reported by *Business Insider*, IBM’s Social Laboratory successfully gamified documentation translation by awarding points to employees who contributed to an internal translation project. Employees were motivated to participate through opportunities to contribute to their own charities. The return on investment (ROI) included improved accuracy, reduced project time, and millions saved in translation costs. (http://articles.businessinsider.com/2011-07-20/tech/30034737_1_ibm-executive-ibm-employees-sametime)



Example: Level Up for IBM Connections

SAMSUNG

Samsung's online community includes a gamified loyalty program that motivates users to contribute to the community. Players can unlock badges and levels by completing quests to learn about Samsung products. (<http://www.samsung.com/us/samsungnation/>)



Example: Samsung Nation

MICROSOFT

To help customers learn about new features in Microsoft Office 2007 and 2010, Microsoft's Office Labs used a gamified approach. The Ribbon Hero and Ribbon Hero 2 games have helped [millions of users](#) learn about the new version of this product. (Microsoft Office Ribbon Hero 2: <http://www.ribbonhero.com/>)



Example: Microsoft Office Ribbon Hero 2

CISCO

To give network customers an enjoyable way to learn to use binary numbers, Cisco created its popular Binary Game. According to Cisco's Learning Network, the game has been played "about a million times around the world" to help users convert 8-digit binary numbers to decimals. (Cisco Binary Game: <https://learningnetwork.cisco.com/docs/DOC-11119>)



Example: Cisco Binary Game

Cisco takes it to the next level by providing games to help network administrators gain networking skills to prepare for professional certification. (<https://learningnetwork.cisco.com/docs/DOC-3820Cisco Mind Share Learning Game>)



Example: Cisco Networking Game

The Cisco Learning Network provides a [Games Arcade](https://learningnetwork.cisco.com/community/connections/games) where users can download learning games for Cisco products. (<https://learningnetwork.cisco.com/community/connections/games>)



Example: Cisco Games Arcade

GAME CHANGERS FOR TECHNICAL COMMUNICATORS

To provide successful user experience for enterprise gamification products, technical communicators need to adapt processes and apply new methodologies. Some sample considerations are described in the following sections.

UNDERSTANDING USER MOTIVATION

The strength of gamification lies in how it motivates users. To achieve these benefits, communicators must understand what motivates their users to perform tasks that address business needs. One way to gain this understanding is by meeting with customers, similar to the Gamification Focus Group hosted by Oracle Application User Experience.

GAMIFIED USER ASSISTANCE ARCHITECTURE

Gamification brings users through a range of activities, starting with onboarding and continuing through mastery. Sample architectural considerations include:

- Planning the player's journey
- Providing onboarding
- Organizing content by quests, challenges, and levels
- Maintaining motivation throughout the player's lifecycle, from novice through mastery

USER ASSISTANCE PATTERNS

Gamification borrows many user assistance patterns from games. A good way to understand these patterns is to play games. Sample patterns include:

- User interface strings for components such as points, levels, quests, badges, and leaderboards
- Onboarding tutorials
- Embedded tips and help

GAMIFICATION TERMINOLOGY

Gamification terminology includes components originating from games, such as:

- Badges
- Quests
- Levels
- Leaderboards

GAMIFIED MESSAGES

In gamification, messages are a critical part of motivating users and are integrated within the game structure. For example, messages:

- Define goals and rules of the gamified system

- Introduce strategy and tactics
- Guide players toward mastery
- Provide feedback, encouragement, motivation, and rewards for leveling up, meeting goals
- Link to a game FAQ

WRITING STYLE

The goal of gamified products is to make tasks fun and rewarding. To achieve this, the writing style is informal, fun, friendly, and minimal. For example, ideal strings are the length of a tweet. Verbs should appeal to a wide range of player motivation.

SCHEDULING FOR PLAY TESTING AND ITERATIONS

Gamified projects undergo several iterations of play testing and feedback. When scheduling gamification projects, it is important allow time for multiple iterations of the product and its user assistance. An example of play testing is shown on the Microsoft Playtest site, <http://www.microsoft.com/en-us/playtest/default.aspx>.

ADMINISTRATION AND UPGRADE CONSIDERATIONS

Gamified products include administrative tasks for configuring components such as levels, rewards, and badges. When planning user assistance projects, consider the requirements of gamification administrators.

Because players are so motivated to achieve goals, upgrades can be disruptive. When scheduling upgrades, alert players in advance to complete current quests before the upgrade.

APIS

Gamified products often use APIs (application programming interfaces). Technical communicators can add value by understanding how to work with and document APIs

ACCESSIBILITY

If a product is required to be accessible, ensure that gamified user assistance conforms to the company's accessibility guidelines. Sample game accessibility considerations are shown at The Able Gamers Foundation's *Includification* website, <http://www.includification.com/> <http://www.ablegamers.org/>.

LOCALIZATION

Localizing games involves more than translation. Allow time for effective localization that considers the target country and culture. Sample localization considerations for gamification:

- Rewards: The same reward does not appeal to all cultures
- Quests: Must be customized to the country and corporate culture
- Leaderboards: Being #1 is not desirable in all cultures

CONCLUSION

Gamification brings exciting changes to enterprise products that engage and motivate users. To stay in the game, technical communicators must prepare for 2014, when over 70% of companies will have at least one gamified product, and for 2015, when over 50% of companies will gamify innovation. The leaders in this field will understand, develop, and deliver effective gamified user assistance.

RESOURCES

Rauch, Marta, *Gamification is Here: Build a Winning Plan*,

<http://www.slideshare.net/MartaRauch/gamification-is-here-build-a-winning-plan>

<http://intercom.stc.org/2012/12/gamification-is-here-build-a-winning-plan/>

Werbach, Kevin and Dan Hunter: *For The Win: How Game Thinking Can Revolutionize Your Business*

<http://wdp.wharton.upenn.edu/books/for-the-win/>

Werbach, Kevin *Gamification course taught online by Kevin Werbach through Coursera:*

<https://www.coursera.org/course/gamification>

Wu, Michael, *What is Gamification, Really?* <http://lithosphere.lithium.com/t5/Building-Community-the-Platform/What-is-Gamification-Really/ba-p/30447>

Gartner Press Release, Egham, UK, April 12, 2011 <http://www.gartner.com/it/page.jsp?id=1629214>

M2 Research <http://www.m2research.com/gamification.htm>

McGonigal, Jane, TED talk, *Gaming Can Make a Better World*

<http://www.youtube.com/watch?v=dE1DuBesGYM>

Kim, Amy Jo, [Gamification 101: Design the Player Journey](#)

O'Broin, Ultan, *Games at Work, Part 1:* https://blogs.oracle.com/userassistance/entry/games_at_work_part_1

Games at Work, Part 2: https://blogs.oracle.com/userassistance/entry/games_at_work_part_2

Webb, Erika Noll, *Gamification* <https://blogs.oracle.com/gamification/>

Oracle Apps Gamification Worldwide UX Design Jam

https://blogs.oracle.com/userassistance/entry/oracle_applications_gamification_worldwide_ux

Webb, Erika Noll, *Making Work Engaging at Oracle* <http://www.gsummit.com/session/enterprise-user-experience-making-work-engaging-at-oracle/>

Why IBM Represents the Future of Social Business http://articles.businessinsider.com/2011-07-20/tech/30034737_1_ibm-executive-ibm-employees-sametime

Gamification Gets Down to Business http://articles.businessinsider.com/2011-07-20/tech/30034737_1_ibm-executive-ibm-employees-sametime

Microsoft Ribbon Hero <http://www.gamification.co/2011/09/15/bringing-games-to-the-enterprise-how-microsoft-ribbon-hero-makes-productivity-engaging/>

Cisco Information Through Play <http://www.nomimes.com/casestudies/hunt/>

7 Examples: Put Gamification to Work

<http://www.informationweek.com/thebrainyard/slideshows/view/232901489/7-examples-put-gamification-to-work>

API (Application Programming Interface) http://en.wikipedia.org/wiki/Application_programming_interface

Game accessibility <http://www.game-accessibility.com/index.php?pagefile=visual>
<http://www.gameaccessibilityguidelines.com/guidelines/full-list/>

Game localization <http://gamelocalisation.wordpress.com/>

AUTHOR CONTACT INFORMATION

[Marta Rauch's LinkedIn Profile](#)

[@martarauch on Twitter](#)

ABOUT THE AUTHOR

Marta Rauch is an STC Associate Fellow and principal information developer and ID lead at Oracle. She participates in gamification initiatives, has worked at two game companies, and completed the 2012 Gamification course through Coursera and Professor Kevin Werbach of the Wharton School. Marta has presented on gamification at an STC webinar, LavaCon, Intelligent Content, and STC Summit.

With 20 years of experience in technical communication, Marta is a member of the STC Silicon Valley Chapter, and has received 15 STC awards for individual and team projects at the regional and international level. Her articles have appeared in STC Intercom, IEEE's electronic library, and the Center for Information Development Management *Best Practices*. In 2012, she was listed as one of the Most Influential in #TechComm and #ContentStrategy. Marta holds a BA from Stanford University, a teaching credential, and a Certificate from UC Extension in Managing the Development of Technical Communication. Marta is a certified Gamification Designer.

USEFUL, SALEABLE, AND BUILDABLE: THE ROLE OF UX

Michael Hughes, PhD

User experience professionals who feel they are not getting the respect they deserve might not have positioned themselves organizationally with people who value what they contribute. Or they might not be sensitive to what value is being asked of them by their organizations. A focus on defining user-centered requirements that are grounded in business objectives and that fall within engineering capabilities can enhance the perceived value of the UX professional.

Steve Farber, leadership guru, gives the advice “Do what you love in the service of those who love what you do.” Whenever I hear UX professionals complain that they continually have to promote the value of what they do, I wonder if they are serving the right people. If people in your organization are not seeing the value you add, maybe you haven’t positioned yourself where you can add the most value.

In this paper I’ll explain how my role has evolved from that of a usability specialist to that of a user experience (UX) architect. In making that transition, I have increased my impact on product strategy and I have established a higher perceived value in the organizations I work for. Essentially, I will discuss how my emphasis and contribution has shifted from just making the product usable, to defining a product that is useful, saleable, and buildable.

THE EVOLUTION OF A UX ARCHITECT

Like many other UX professionals, I got into UX through usability testing. And like many people in usability testing “in the day,” my involvement with a product usually came late, literally weeks before a scheduled launch or even after a product or web site was already in the market. My major challenges back then involved problems with UI design that could not get adequately fixed because of time constraints or because the problem had been baked too deeply into the technical architecture of the product. I went through so much pig lipstick in those days that I was on Miss Piggy’s speed dialer.

Then I landed a job with a different company and made the formal transition into UX. This let me move further upstream in the development process. My early UX experience involved working in the design phase, producing detailed Visio wireframes based on a product manager’s Product Requirements Document and then delivering those wireframes to be coded into a working user interface by the developers.

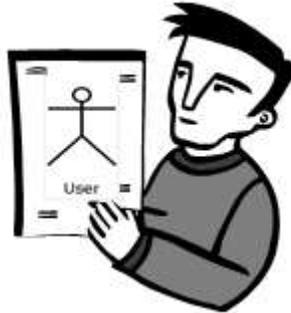
As ideal as that sounds, I started encountering new difficulties. In particular, there was one product manager, “Dave,” that the design and development folks did not want to work with. For one thing, he would look over the shoulders of the QA testers and start complaining that the product wasn’t what he had envisioned when he wrote the requirements. Eventually, he started sitting with developers during coding so that he could shape the product to his expectations.

All of this was going on in a company with an expectation that products were to be *built to requirements* and the wireframe was the functional specification for how the UI should look and operate. The situation was made particularly painful because Dave was often right and his meddling resulted in noticeable improvements. Doh!

While everyone else was wondering how to get rid of Dave, I was wondering how we were managing to **not please** him. This was his product, these were his requirements, he was right there in easy reach, and he was certainly not shy with his opinions. How, then, were we missing a target that big?

With the help of another UX designer who had recently introduced me to Axure, a quick prototyping tool, we started to change the design culture.

Instead of acting like building architects producing detailed blueprints, we started acting more like police sketch artists ala “His nose was big,” “*Like this?*” “No, longer, and not as wide.”



UX Architect as Sketch Artist

In our new model, the design phase became highly interactive and iterative. We would quickly put up prototypes and ask, “What if it looked like this and acted this way?” Wow, did that get some good conversations going! And because our wireframe tool generated what at least acted like working prototypes, we could do usability testing to get real user feedback and test competing design ideas. By the time we started coding the product, Dave knew exactly what he was going to get and either loved it or had accepted the compromises he knew we would have to impose.

UX DOCUMENTS AS REQUIREMENTS — NOT JUST DESIGN ARTIFACTS

In my current team, we have taken this model of a highly interactive and iterative design phase to an even higher level—letting the UX design artifacts replace the requirements documents altogether. Part of our push has been the need to accommodate our development methodology of SCRUM. SCRUM is an Agile development process which emphasizes minimum design documents and which values “learn as you go” as opposed to “If we think long and hard enough ahead of time, we will anticipate everything and will build the optimal solution.”

Our primary UX design documents that we use to define requirements are *Scenarios* and *Wireframes*. Essentially, our scenarios are two-act plays. We label Act I “Problem” and Act II “Solution.” The Problem section can be as short as one sentence or as long as several paragraphs, depending on the complexity of the requirements scope. Problem sections can reflect the shortcomings of an existing process, functionality that is missing from a current offering, or can capture complaints we have received about our current product’s performance.

What is important is that they establish the *context* in which the required function will add value for the customer. It is often an early warning if we have difficulty positioning a requested feature within the context of a user problem it would solve or alleviate.

The Solution section is a narrative that describes an alternative experience to the one described in the Problem section—one brought to a happy ending by the product feature to be built. It is not a detailed specification, but rather a loose contract between the product manager and the developers about how the solution will look and act—in the context of an authentic user situation.

We often supplement the narrative text with low-fidelity wireframes. We use Balsamiq Mockups because it renders a hand-drawn look which reminds everyone that this is essentially a conceptual piece. For example, a rough drawing of a calendar date selector tells the stakeholders there will be an easy way for the user to enter a date; it tells the developer to insert the appropriate date selector widget from our approved widget library.

These low-fidelity mockups have the advantage of being quicker to produce than prototypes, but they are not as useful for usability testing. And that illustrates another shift from my role as usability tester to UX designer. The focus at the front end of the project is less about “Would this be *usable*?” and more about “Would this be *useful*?”

For example, in the early stages of setting requirements for a new report, I am more interested in capturing what information a user would need to see in that report and how the user would want to manipulate the data, e.g, learning which attributes users would want to filter by. It is later in the development phase that I shift my emphasis more to how intuitive and usable are the widgets and the UI in letting the user run, schedule, and manipulate the report.

STARTING THE GOOD FIGHTS

The combination of the narrative of the scenario and the rendering of the wireframe gives us an artifact that we can put in front of users and stakeholders and ask “Would this have value, is this what you want?” We can also put it in front of developers and ask “Can we build this and for how much?”

And that’s when the “good fights” begin. My experience has been that stakeholders and developers can think they are in agreement over traditional requirements and specs when they really aren’t. The combination of a scenario and one or two supporting wireframes, however, surfaces any disagreements quickly and in concrete terms that can be resolved. These new artifacts become a Lingua Franca, i.e., “...a language systematically used to make communication possible between people not sharing a mother tongue, in particular when it is a third language, distinct from both mother tongues.” (1)

The following is a screenshot from our wiki page where we post our scenarios, and it shows an illustrative example of a scenario and wireframe. Stakeholders have access to these pages and can “follow” them to be notified any time we update them as well as comment directly on them.

Ticket Rating
 No recommendations | Updated today at 5:06 PM by Michael A. Hughes | Tags: None | Add tags

[Edit](#) [Page Actions](#)

Problem

Mark, a SOC supervisor, creates a monthly management report to describe how well his center has performed. He has data on SLA and SLO performance for his areas of responsibility, as well as other performance indicators. But he does not have data on customer satisfaction.

Solution

Jeff, a Security Analyst for ABC, has just finished resolving an open ticket he has been working with the SOC. When a ticket is opened, a feedback widget is provided (needs to be able to link from either the Ticket Detail or on the Ticket Manager). Jeff rates the performance as a 3 out of 5 and adds a comment saying that he felt there were too many iterations before the policy request was successfully implemented.

At the end of the month, Mark runs a customer feedback report and gets the following statistics:

- Total tickets closed
- Total feedback forms filled in
- Average rating
- Comments

He gets the above aggregated for all of the SOC people he manages and he gets the same statistics for each SOC person in his area of responsibility.

Accessing from Ticket Manager:

The screenshot shows a 'Ticket Manager' interface. At the top, there are filters for 'Select filter', 'All issues', 'All', 'All', and 'All'. Below the filters is a table with columns: Ticket ID, Manager, Created, Issue Type, Issue, and Status. The table lists several 'Security Incidents' and 'Service Requests'. A 'Rating' modal window is open over the table, containing a 'Comments' text area and 'Submit' and 'Cancel' buttons. A yellow callout box points to a comment icon in the ticket list with the text 'User clicks comment icon'.

Scenario and Wireframe

CONCLUSION

My journey in user experience has led me from being involved at the tail end of the development cycle and asking “Is this usable?” to being at the very front end of the process helping to define requirements around “Is this useful to a customer?” “Could we sell it if we had it?” and “Can we affordably build it?” One of the biggest differences this has made is on the impact that UX has on the business: We are major players in setting strategy that influences product profitability.

The engineering manager who hired me into my current UX position defined engineering productivity as “revenue divided by the number of engineers.” And he continually warned us against getting overly involved in development projects that could not show a positive impact on revenue.

So by all means, keep your usability “chops,” but use them in a supporting role as you pursue the real prize, namely, defining products customers will want to use, the business can sell, and engineering can build economically.

REFERENCES

[1] *Viacheslav A.C.*, The problem of the Caucasian Sprachbund in *Pieter Muysken, ed., From Linguistic Areas to Areal Linguistics, 2008, p. 31. ISBN 90-272-3100-1*

AUTHOR CONTACT INFORMATION

Michael Hughes
User Experience Architect
IBM Security Services
mhughes2@us.ibm.com
1.404.236.3281

AUTHOR BIOGRAPHY

Michael Hughes has a PhD in Instructional Technology from the University of Georgia and an MS in Technical and Professional Communication from Southern Polytechnic State University. He is a past president and a Fellow with the Society for Technical Communication. He conducts user research and designs user interfaces for web-based security products for IBM.

WEB DESIGN AND DEVELOPMENT

[Every Page is Page One](#) page 197

EVERY PAGE IS PAGE ONE

Mark Baker

For readers who arrive at your website or help system via search, links, or social media, Every Page is Page One. Readers quickly dismiss pages that don't work for them, so to successfully hold their attention and meet their needs, every one of your pages should work as page one.

The key characteristics of a good Every Page is Page One topic are: Standing alone functionally, having a defined and limited purpose, establishing context, conforming to a type, staying on one level, assuming the reader is qualified, and linking richly. To serve users who seek information on the Web, we need to learn to create good Every Page is Page One topics.

On the Web, every page is page one. That is, when you search for information on the Web, whether you use a search engine or follow a link, you land on one of the billions of pages that the Web contains -- any one of them -- and that, for you, is page one.

This, by itself, is simply a fact about the way the Web works. There is no "Start Here" page for the Web. At whatever point you dip your toe into it, that is your page one. We can't avoid this. There is no other way to read the Web. Whether you are a reader or a writer, and whether you like it or not, this is the way the Web works. Every page is page one.

Writers are taught that it is their job to organize content for the reader, and the most common way to organize content is as a hierarchy or sequence, as in a book. Only one page is page one. The other pages descend from and rely upon page one, and all the pages that stand between them and page one. Authors construct pages that are page 16 or page 187, or page 2596. But while their page 187 may be a brilliantly conceived and executed page 187, it does not work as page one for a reader who lands on it as the result of a Google search.

Book-like organization does not work on the Web, where a reader can land on any page out of the blue. The Web is a hypertext media, a navigable network of interconnected pages. But no Web page comes before or after another Web page, except in the particular sequence chosen by the reader. You can't write a Web page as if it were anything other than page one.

INCLUDE IT ALL. FILTER IT AFTERWARD.

Some writers believe that books work better than the Web, because they are better organized and contain fewer distractions. Yet we have known for years that people seldom read the manual [1], and it is clear that users are turning to Google as their first place to look for help with technical problems [2]. Why do users prefer the messy web over the well-organized book? Because, in David Weinberger's words,

We seem to be making a cultural choice---with our new infrastructure's thumb heavily on the scale---to prefer to start with abundance rather than curation. Include it all. Filter it afterward. [3]

That is, the user prefers to access a large volume of content and filter it for themselves, rather than relying on a writer to do it for them. Why?

- **Because they can.** In the paper world, filtering a large volume of information was time consuming and difficult. We relied on writers to do it for us because we had to. Today, the Web presents us with a large number of filters we can use to easily filter content for ourselves. Google search and its

ranking algorithm is a filter, as are social networks like Twitter, Facebook, and LinkedIn, and content aggregation and duration sites like StackOverflow.

- **The long tail.** The Web contains far more information on most products than you could ever hope to include in your documentation. A lot of that information may only be of interest to a few people, but add up all people who want one item of that obscure information, and it will probably encompass most of your user base. People have learned that information that is not in the user manual can probably be found on the Web. Thus they turn first to the Web.
- **New information.** The information about a product tends to keep growing long after the product has been released. It is not until it is in use in the real world that all of a product's potential, its limitations, and its problems come to light. Many manuals were put to bed before the product development was even finished. The Web contains a lot of new information that was not available when the manual was written, and that is where the user goes to look for it.
- **The voice of experience.** On the Web, you get to talk to people who have actually done the task you are trying to do. People know that the official documentation does not always tell the whole story, for all sorts of reasons. (For instance, the maintenance procedure was probably written before the bolts on the assembly had a chance to seize, and before the plastic parts had turned brittle.) People trust other users like themselves to give them the real story. They may have access to few such people in their neighborhood, but on the Web they can find them half a world away.

For all these reasons, people are turning to the Web when they have a technical problem with your product. And they have a choice in the content they choose to consult. If your content does not work well on the Web – if it does not work as page one for every reader – they are going to be reading other people's content instead. Putting a conventional PDF or help system up on the Web is not going to win back your audience. You need to give them real workable Every Page is Page One topics.

CHARACTERISTICS OF EVERY PAGE IS PAGE ONE TOPICS

Every Page is Page One topics are neither rare nor strange. There are millions of great examples on the Web. Wikipedia alone provides countless great examples of Every Page is Page One content. The issue for technical writers is to adjust their thinking and their habits from the old book model to the new Web model. One way to do this is to look at the key characteristics that good Every Page is Page One topics display.

STANDALONE

An Every Page is Page One topic stands alone. It does not depend on a previous topic or presume a next topic.

The word “standalone” is often used to describe what we mean by a topic in topic based writing, however, there are two distinct kinds of topic based writing that are practiced in technical communication today, and both use the word “standalone” to describe a topic, but they mean somewhat different things by “standalone”.

Andrew Brook wrote a blog post on the subject[4], in which he compared topics to electrons:

1. A topic is to a document what a subatomic particle (such as an electron) is to matter. It is the basic component in a document. Each topic can and must stand alone.
2. Combinations of topics are like atoms. They form a section of a document containing a group of related topics. This corresponds to a book within an online help TOC, or a chapter within a book.

3. Groups of sections are like groups of atoms, or molecules, for example, a water molecule. These correspond to an entire document.
4. Groups of documents form a library, which is like the various molecules combined together to form the complex matter, or compounds, that we encounter every day, everything from plastic to clothes to hamburgers.

Here the word “topic” is used to describe a building block intended be used to make something larger and more complex. In this case, “stand alone” means to have an independent existence, but not necessarily to function alone. To expand the scale a little, a brake caliper is a basic component of a car. Does the brake caliper stand alone? Certainly it can stand alone on the shelf at the auto parts store, but it serves no useful function until it is attached to a car. In this sense, it does not stand alone — it can only perform its function when integrated into a larger system.

A topic that is designed as part of a larger information product, that is intended to build sections and chapters of a book, is not an Every Page is Page One topic. An Every Page is Page One topic is not like a brake caliper on a car. It is like a car in a taxi fleet. It may be part of a collection of other topics, but it is fully functional by itself. It is standalone in the sense that it functions alone.

SPECIFIC AND LIMITED PURPOSE

It is natural for the author of a topic to have a specific limited purpose in mind when they write. That well defined purpose guides them as to what details are salient to their topic. When an author writes a recipe, for example, they have a specific, limited purpose which they understand and which guides how they write the recipe. That purpose is to enable an experienced cook to cook a particular dish. It is not to teach cooking, or enable a complete novice to suddenly cook a complex dish. For that, other resources are required. The recipe has a specific and limited purpose, and teaching cooking is not part of it.

However, in technical communication, manuals, chapters, and topics are often not written with a specific and limited purpose. There is a common tendency for tech docs to grow and bloat over time as more and more material is included. But even in the first draft, many tech docs are not written to serve a specific and limited purpose. Why not?

One reason is that the tech writer’s task is often defined simply as “provide documentation for the Widget.” As a purpose, this is neither specific nor limited. All too often, this vague and general purpose turns the writing task on its head. Rather than asking, what facts do I need to include in this topic in order that it fulfill its purpose, the author is left looking at a collection of facts to be included in the documentation, and asking, how do I create a set of topics or chapters such that all these facts will be included? None of the individual topics serve any defined purpose other than to be a container of facts to be documented.

To write a good Every Page is Page One topic, therefore, you must begin by defining the specific and limited purpose that topic is designed to serve for the reader. Never add anything to a topic just to find a place for it. Either that new information serves a specific and limited purpose of its own, in which case it merits its own topic, or it serves no purpose at all and should be omitted.

CONFORM TO A TYPE

There are familiar patterns to many kinds of topics. Recipes, car reviews, and API reference topics, all follow similar patterns. These patterns help the writer to ensure they are supplying a complete topic and allow the reader to recognize and consume the topics more quickly.

Note that when we talk about a recipe or an API reference having a type, we are not talking about generic concept, task, and reference topics, which are something quite different, and often more granular. In this

context, the type of the topic is something very specific. A recipe has the name of the dish, the list of ingredients, a cooking procedure, the number it serves, and sometimes things like a suggested wine pairing, nutritional information, and advice for people with allergies. This pattern is unique to recipes, and it tells you how to write a recipe and what to expect from one.

While many topics naturally conform to a type, there can be a lot of natural variation between how different authors implement that type, including different organizations of the topic, and different inclusions or exclusion of optional parts of the topic type. For a systematic authoring project, you should create your own rather stricter definition of the common topic types to ensure greater consistency and completeness in the topics of that type that your authors create.

These topic types are not arbitrary. They come from the information needs of the readers. This is consequence of having a specific limited purpose. When a topic has a specific and limited purpose, it is likely that all topics that serve a similar purpose will contain the same kinds of information, often in a similar order. Once you have defined the specific and limited purpose of your topic, you will often find that you have gone most of the way to defining the topic type as well.

ESTABLISH CONTEXT

Because the reader may come from anywhere, and often arrives at a topic through an imprecise mechanism such as a Google search, a topic should clearly establish its context in the subject domain. If you have ever landed in the middle of a help system from a search and found that you had no idea where your search had landed you, you have experienced the problem caused by lack of context.

That a topic must establish its context is a necessary consequence of it being standalone. If a topic is to stand alone, the reader must be able to come to it from anywhere and be able to tell where they have arrived. If the topic does not provide that context, then the reader must seek elsewhere to orient themselves and the topic, ipso facto, does not stand alone.

A topic must therefore begin by placing itself in context. This may sound dangerously like the sort of front matter that minimalism has been encouraging us to leave out. It's not. Indeed, providing context is necessary to minimalism's goal of getting to action. You can't act effectively if you don't know where you are. A good Every Page is Page One Topic orients the reader quickly and then gets on with the job at hand.

STAY ON ONE LEVEL

Changes of level are a necessary part of any course of study. As you study a subject, you sometimes need to dive down into the details in order to get a practical illustration of a general principle. When you are working on some detail, you sometimes need to understand a more general principle that explains why the detail works the way it does (and not the way you expected it to).

When any individual reader will need to change levels, however, is very much dependent on their personality and their circumstances. Some prefer to absorb the big picture before acting. Some prefer to immerse themselves in the details and integrate their understanding of the whole at a later time. Some only want to follow procedures without understanding. Some are in a hurry to complete an individual task. Some are preparing for a project and want a broad view so they can plan the best strategy overall. The decision about when to change levels, therefore, is really best left to the individual.

Books don't work that way, though. In books it is the author who decides when to present the big picture and when to delve into details. This is a necessary consequence of the linear structure of a book. Most books are designed to be read in a particular order, and that means that they will change levels at the time and place of the author's choosing.

But the Web allows the reader to choose when to change levels. Indeed, if your content is on the Web, you have no way to prevent readers from making this choice for themselves, so you might as well support them. A good Every Page is Page One topic, therefore, stays on one level, and leaves it to the reader to decide when they want or need to change levels. A great Every Page is Page One topic provides them with links that help them access the lower or higher level topics when they want them.

ASSUME THE READER IS QUALIFIED

I noted above that a recipe does not attempt to teach the reader how to cook. That is outside of its specific and limited purpose. Rather, a recipe is written on the assumption that the reader is qualified to read the recipe and to cook the dish. It assumes, for instance, that they know how to measure a cup of milk or how to sauté a mushroom.

Books sometimes attempt to serve everyone, no matter what their qualifications, though not usually with great success. Such books can, of course, be incredibly tedious for experienced users. They can also be tremendously uneven, treating one subject in detail and another at a very high level. Sometimes, this is a result of accretion over time. The writer gets a complaint that a customer could not complete a certain task, and they are ordered to add a detailed procedure to the docs at that particular user's level of knowledge. The procedure on the next page, however, about which no complaint has been received, remains at a high level as originally written.

A topic can't be all things to all people. There is not enough room. Add everything into one topic, and it ceases to be a topic. Each topic must stick to its specific and limited purpose, and that means it must be written on the assumption that the reader is qualified to undertake the task it describes.

Those assumptions, about what the reader already knows, and what they need to be told, form the basis of the definition of the topic's type.

Of course, your topic collection may be used by readers of different levels of qualification, and may need to provide topics for those different qualification levels. An individual topic assumes that the reader is qualified according to its type and its purpose. The topic collection includes topics written for people of different levels of qualification. Ideally, these topics link to each other so that readers who are not qualified to read the current topic can easily find the topics they need to improve their qualifications.

AS LONG AS IT NEEDS TO BE

One of the first questions writers ask about topic-based writing is, how long should a topic be. The answer to this question can depend on what kind of topic you are creating. If you are creating a building-block topic for reuse, you may want it very small so that you can maximize reuse. But such topics usually don't stand alone in the sense that an Every Page is Page One topic requires: they don't function alone. An Every Page is Page One topic needs to be big enough to function alone. It needs to be as big as a car, not as big as a brake caliper, and not as big as an ocean liner either.

The best guide to the length of an Every Page is Page One topic is that it needs to be long enough to fulfill its specific and limited purpose. Having a well-defined purpose, a well-defined topic type, and making sure that your topic assumes the reader is qualified and stays on one level, will give you a correctly sized topic.

(As for topics sized for reuse, it is an entirely separate matter if you want to construct your Every Page is Page One topics from reusable building blocks. Just don't confuse the two, or assume that a topic sized for reuse is going to work as an Every Page is Page One topic for the reader.)

LINK RICHLY

An Every Page is Page One topic should link richly to other topics. The reader reaches an Every Page is Page One topic by selecting it from a set of search results or by following a link. In a perfect world, that topic would fulfill every reader's every need and that would be the end of their research. In that case, the topic would not need to link to other topics. But we don't live in that perfect world. The readers of our topics will frequently want to move to other topics, and a good Every Page is Page One topic facilitates that movement with rich linking.

If we don't provide rich linking, the reader will go back to Google and search again, and very possibly end up somewhere outside our content set altogether. If we want to keep our readers in our content, we need to provide them links to within our own content set that will meet as many of their information needs as possible.

The need for rich linking, and the linking we should provide, follows from the other properties of an Every Page is Page One topic.

- An Every Page is Page One topic stands alone, but it does not stand in complete isolation. It does not depend directly on a particular previous topic or a particular next topic, but it depends indirectly on all the topics around it on related subjects. It is part of an information web, and it needs to be connected to that web through links.
- An Every Page is Page One topic has a specific and limited purpose. The reader may have a larger or more complex purpose. The individual topic fulfills one part of that broader purpose (as the taxi that takes you to the airport completes a well-defined journey that is part of a larger journey). To facilitate the reader in moving on with their larger purpose, the topic should link to related topics along as many different paths as possible.
- An Every Page is Page One type establishes its context. Context is where it fits in the universe of the subject it discusses. But because it has to establish its context briefly, it can't explore it fully. Therefore it should provide links to topics that explore that context more fully. Also, because readers sometimes land on not quite the right topic, the context helps them recognize where they have landed, and should help them get to where they really need to be through links.
- An Every Page is Page One topic has a well-defined type. That type provides that some things will be described in the topic itself, and others will be passed over without explanation. A recipe, for instance, will instruct the reader to sauté, but will not explain how to do it. A link from the word sauté to a topic on how to sauté makes the recipe topic more useful and more usable for the inexperienced cook.
- An Every Page is Page One topic assumes the reader is qualified, and because some readers may not be qualified, in some particular way, it should provide links to the material that will help them qualify themselves.
- An Every Page is Page One topic is as long as it needs to be. To prevent it from being bloated from its proper length, by people wanting you to insert additional bits of material they think are important, it should use links to reference any ancillary and related content.

CONCLUSION

On the Web, every page is page one. Writing an individual page-one Web page may be easy enough, but when you need to write a whole set of topics to cover a large subject, it is easy to fall back on the book form. But readers don't read technical communication the way they read novels. They never have, and they certainly don't today when the first (and often only) place they look for technical help is on the Web.

To serve these readers, we need to write Every Page is Page One topics. To do that successfully, we have to take a disciplined approach to each topic we write. The basic properties of a good Every Page is Page One topic outlined above provide a foundation for that discipline.

REFERENCES

- [1] Novick, David G., Ward, Karen, "Why Don't People Read the Manual?" (Proceedings, SIGDOC'06, 2006)
- [2] Parnin, Chris and Treude, Christophe, "Measuring API Documentation on the Web" (Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering, 2011)
- [3] Weinberger, David, *Too big to know: rethinking knowledge now that the facts aren't the facts and experts are everywhere, and the smartest person in the room is the room*, Basic Books, 2011, p. 176.
- [4] Brooke, Andrew, "Topical Doc", *A Tech Writer's World*, June 2011, <http://techwriters-world.blogspot.ca/2011/06/topical-docs.html>

AUTHOR CONTACT INFORMATION

Mark Baker
Principal Consultant
Analecta Communications Inc.
Ottawa
Canada
613.422.9400

AUTHOR BIOGRAPHY

Mark Baker is a content strategy and structured authoring consultant who helps companies move their technical content to the Web in a way that really works on the Web. His book, *Every Page is Page One: Topic-based Authoring for Technical Communication and the Web*, will be published in the fall by XML Press. He is also the creator of the SPFE ("spiffy") architecture for structured authoring and publishing, which is described at SPFE.info.

Mark has been practicing and implementing structured writing since the SGML days, and gave his first paper on topic-based authoring at SGML 95, under the title "Component Based Information Development". His previous positions include Manager of Information Engineering Methods at Nortel and Director of Communications for SGML pioneer OmniMark Technologies. He blogs on topic-based authoring for the Web at <http://everypageispageone.com>. He tweets as @mbakeranalecta.

PROGRESSIONS

[Cultivating In-House Sources of Documentation](#) page 207

[Is Your Content Ready for Localization?](#) page 211

[Managing Technical Knowledge: Contributing Factors to Knowledge Loss](#) page 215

[SME — ME — E Screencast creation and collaboration](#) page 221

[Teaching a College-Level Editing Class](#) page 229

[The Retirement of Technical Communicators — Making Retirement an Adventure](#) page 235

[There's Nothing Simple About Simplified Technical English](#) page 239

[Using Mentoring Programs to Collaborate With Industry](#) page 241

[Why Technical Writing Is Always About Training](#) page 245

CULTIVATING IN-HOUSE SOURCES OF DOCUMENTATION

Eileen Palsson

Nothing can replace knowledge of our end users and direct communication with them. However, we can also collect valuable details and feedback from within our own organizations. In this session we will discuss five in-house sources of input that authors can draw on: support technicians, developers (R&D), trainers, new employees, and translators.

We will discuss what each of these groups can offer and how to communicate with them and motivate them to contribute. Not all organizations have all these departments or roles. However, your company may have a similar source that you can draw on. Even in the absence of direct communication with users, there are valuable resources available to assist in your efforts to produce the most effective user documentation.

SUPPORT TECHNICIANS

Support technicians are in possession of exceptionally useful information about the most common user questions or misconceptions. They know which areas are lacking in documentation and which areas are more strongly represented in their support cases. They often have very specific suggestions for improvement of documentation.

You have a juicy carrot to offer when it comes to collaborating with Support: Every contribution they make towards improved documentation has the potential to make their job easier.

MAKING IT KNOWN THAT YOU WANT TO COLLABORATE

- Meet with the support manager and ask for ideas.
- Tell Support what's in it for them (for example fewer cases).
- Get mentioned in their online newsletter, blog, or whatever other internal communication system they use.
- Some support departments have their own body of documentation. Explain how they would be better served by working with you to spread the information via the official product documentation, and referring users to that documentation when appropriate.
- Suggest multiple ways they can provide input and feedback:
 - Email
 - Phone call
 - Stopping by your office
 - Sending a link to a support case where they think better documentation could have helped
- If support technicians are timed or their performance strictly measured, feedback to information developers should somehow measure in.

DEVELOPERS (R&D)

Subject matter experts (SMEs) have deepest *technical* knowledge about your products. Your review process should include SME review.

MAKING IT KNOWN THAT YOU WANT TO COLLABORATE

- If you are a member of an agile development team, there are daily stand-ups or daily scrum meetings. Be vocal about what you're working on and ask others to check the work.
- Testing new features must include reviewing the documentation for technical accuracy and suitability from a user perspective. If it is not already, try to have this added and enforced as part of the normal course of product development.

TRAINERS

Trainers, too, know the products well. They also know which areas users have trouble with.

MAKING IT KNOWN THAT YOU WANT TO COLLABORATE

- If they don't do so already, have trainers show users how to refer to product documentation. For example, when a trainee asks a question, a trainer can say, "Well, let's look for that information in the help, shall we?"
- If trainers refuse to use product documentation, find out why – and then fix the problem!
- Training material can be a great source of input for overview topics and tips & tricks. Ask trainers to provide you with the latest training modules or slideshows.
- Tech writers and trainers should collaborate closely. At many companies the two groups share content. The knowledge goes both ways and benefits both sides.

NEW EMPLOYEES

New employees are often very enthusiastic and ready for anything. Their experience with the product may be very similar to that of other new users, and you can make good use of their fresh perspective. They can point out areas where the documentation is unclear. Typically they point out where overviews are lacking or misleading.

Talking to you can also provide an outlet for a new employee's frustrations – which in some respects may be the same as user frustrations.

MAKING IT KNOWN THAT YOU WANT TO COLLABORATE

- Introduce yourself to new employees who will be learning "your" product.
- Suggest that product documentation is useful when they are learning the product or when they encounter a difficulty or a question.
- Let new employees know how their comments about the documentation can be valuable to the company in improving the user experience.

- Provide your contact details and a note about what you're interested in on a business card.

TRANSLATORS

Translators who are involved in localizing your documentation and/or user interface are language experts who can point out inconsistencies and unclear explanations. This is valuable input, especially if your target audience includes non-native speakers of the language you are writing in.

Examples of inconsistencies that translators notice:

- Synonymous use of *select*, *click* and *choose*.
- Synonymous use of *pane*, *area*, *part*, *group* etc.
- Incorrect (and misleading) punctuation of dependent clauses.
- Unnecessarily complicated use of different tenses.

MAKING IT KNOWN THAT YOU WANT TO COLLABORATE

If your products are being localized, make it known to the translation agency or individual translators that you are open to feedback regarding the original text. Even if this can't affect the current version (where localization is already in progress), it can have great impact on future versions. Think long term!

WHEN YOU RECEIVE CONTRIBUTIONS OR FEEDBACK

- Make it pleasant for the contributor.
- Appreciate the input. A sincere thank-you goes a long way. Thank them for specific information. Thank them for pointing out an area of missing or unclear information. Thank them for their assistance in improving the documentation.
- Try not to let your ego get in the way; you are all on the same team, working for the best user experience.
- When you make changes based on received input, send a link or a copy of the result to the contributor.
- If you know that it will take awhile to implement a suggestion, let the contributor know that it's on your radar. Follow up and let them know when it's fixed and in which version or release they'll see the result. If possible, include a copy of the result, a link to the new help topic, or other concrete proof.
- Even if you don't change anything, take the time to explain why, so that contributors do not get the impression that you've simply ignored the input. Show them that you hear them.
- Recognize contributors – give them credit somehow, for example by mentioning it as a plus at retrospective meetings or praising them to their manager.
- **Make these things *habits*. They soon become part of your routine, and they pay off!**

ADDITIONAL TIPS

- Monitor support channels, FAQs maintained by others, related user forums and so on.
- If you don't mind interruptions, keep treats in your office. People may be more likely to drop by with suggestions.
- Resort to bribery! Candy! Money!
- One company offered \$200 per help topic that was written by SMEs outside of work time.

FINAL ADVICE

- Keep repeating the same consistent message – that you welcome input and corrections.
- Always express appreciation, not defensiveness.
- Invest time. It is worthwhile to cultivate these sources in the long run.

AUTHOR CONTACT INFORMATION

Eileen Palsson
Senior Technical Writer
ReadSoft AB
Box 1224
SE-16428 Kista, SWEDEN
+46 73 3378639

AUTHOR BIOGRAPHY

Eileen Palsson, Senior Technical Writer and senior STC member, has worked in the field for over 16 years, first as a lone writer and now as part of a team of five writers producing and maintaining software documentation in eight languages. At ReadSoft AB in Kista, Sweden, she is instrumental in obtaining the cooperation of different company departments in reviewing product documentation during its development and providing feedback on its efficacy after publication and distribution.

Eileen has a B.A. in German Language and Literature from the University of Michigan. She is also fluent in Swedish. As a speaker Eileen has trained with Stockholm International Toastmasters since September 2010. She has obtained both the Competent Communicator award and the Competent Leader award, and is currently serving as President of ReadSoft Toastmasters.

IS YOUR CONTENT READY FOR LOCALIZATION?

Mary Dulin

Are you writing with a global audience in mind? Can your content be quickly understood and clearly translated to meet your deadlines? Word choice, sentence construction, graphic design, and layout can all affect the quality, cost, and turnaround time for your translations. This paper describes some of the issues to keep in mind when writing content that will be translated, so that your content can be translated quickly, accurately, and at reasonable cost.

To build a quality home, you need a firm foundation, solid construction methods, and appropriate finishings. Design details are captured in blueprints, which enable the builder to construct the home as designed. For successful translations, you need a similar approach. Following a well-defined template provides a solid foundation. Word choice and sentence construction directly affect the translation effort and cost. Your “finishings” – numbers, symbols, graphics, etc. – must be appropriate for the audience. For your translation vendor to understand your requirements, you need to provide the “blueprints” – source content, software strings, fonts, graphics, style guide, glossary, and any other helpful information.

LAY A FIRM FOUNDATION

Just as a home needs a solid foundation, good technical writing needs a solid base. Following a template ensures that your content has a consistent layout - one that can be replicated in other languages.

- Adhere to a template. Localized content can be easily formatted when template styles are used consistently.
- Limit the number of styles in your template. A clean, simple template allows your translation vendor to quickly format the localized content with a minimum of effort, which keeps your translation costs down.
- Avoid manual formatting such as tabs or extra line breaks. Define template styles that provide the necessary indentations and spacing.
- Avoid style overrides. Overrides may go undetected by the translation vendor.
- Use a table to present alphabetized content, such as a glossary or list of terms. Terms that are presented in a table can quickly be rearranged into alphabetical order in the localized language.
- Use the features of your authoring tool to create cross-references and tables of contents. Using the automated software features allows your localization vendor to quickly recreate the cross-references and tables of contents in the localized text.

CONSTRUCTION IS KEY

Good construction is essential, whether you are building a home or writing user instructions. Follow these writing tips to improve the likelihood that your content will be understood and translated correctly. Using consistent verbiage, present tense, and active voice also reduce the number of words that need translated, which keeps translation costs down.

- Use short, simple sentences. Longer sentences are more likely to be misunderstood and possibly translated incorrectly.
- Choose words that are precise, with one or few meanings. Words that have multiple meanings or that can be either a verb or a noun are easily misunderstood.
- Use active voice and present tense. Such sentences convey clear messages using the fewest words.
- Write so that the subject and verb are near the beginning of the sentence. This standard construction is easily understood and translated.
- Be consistent! If a warning statement appears more than once, write the warning exactly the same way every time. Refer to a product the same way every time. If a procedure or step appears more than once, write it the same way every time. Using repetition presents a consistent message to the user and has one of the greatest impacts on reducing translation costs!
- Do not use slang or jargon. Americans may understand “keep a fire extinguisher handy,” but such an expression may not translate easily or clearly.
- Do not use contractions. To facilitate translation, write “do not” rather than “don’t,” for example.
- Define abbreviations and acronyms on first use. Use abbreviations cautiously. You may know what “Cat. No.” means, but your translator may not. Additionally, a comparable translation may not fit into the same space.
- Do not use Latin abbreviations. Use “that is” instead of i.e., and “for example” rather than e.g.
- Avoid awkward constructions such as “his/her” or “battery(ies).” Such construction is awkward in English and even more challenging to translate, as many languages change grammatical construction or gender depending on whether the subject is singular or plural.
- Avoid verb forms that end in “-ing. Some languages do not have a similar verb form, so the translator must rework the entire expression.
- DO include optional words such as “that” when they add clarity. Include words that indicate relationships to make your message clear.
- Allow room for text expansion. The general rule of thumb is to allow for up to 30% expansion; software strings may increase by up to 50%. Keep text expansion in mind especially when crafting headings, tables, and callouts. You should also consider text expansion when designing a multi-language document in which each language must fit into a predefined space, such a one column or one page.
- Do not embed text in graphics. Simplify the translation process by using numbered callouts which correspond to a legend, or define text boxes around your graphic to hold callouts. Another option is to use layered graphics and isolate translatable text on one layer.

APPLY APPROPRIATE FINISHINGS

The finishings on a house must be appropriate for the style of the house and its location. Technical content must have the proper “finishings” (items such as measurements, currency, contact information, and graphic images) so it is meaningful for the reader.

- Provide metric equivalents for all weights and measures.

- Write dates in a format that is universally recognized. While the U.S. commonly uses the month-day-year format, many countries commonly use the day-month-year format. Determine a format that will be understood by all your users – you may want to spell out the month, use month and year only, or follow some other convention that is common for your industry.
- Include contact information (address and telephone numbers) in a form that is useful for an international user. Toll-free 1-800 numbers are often valid only within the United States or North America. Provide an appropriate telephone number for users outside of the U.S. Spell out the words “telephone” and “fax” in contact lists.
- When listing cost, such as in a price list, be sure to specify the currency. Note that the dollar sign (\$) alone does not indicate U.S. dollars. For example, (USD) indicates U.S. dollars, while (AUD) indicates Australian dollars and (CAD) indicates Canadian dollars.
- Be aware that many symbols do not have a universal meaning. For example, the # symbol can mean “number” or “pound” in the U.S., but its meaning may not be clear outside the U.S. Also, when referencing “pound,” be sure to specify whether you mean weight or British currency.
- The double quote (“) that is used to represent “inch” in the U.S. may have little meaning to your global users. Provide an appropriate metric measure.
- Choose colors carefully, as colors may have religious or cultural significance.
- Use caution when designing icons. Try to use generic symbols that are internationally recognized. The *Microsoft Manual of Style*, 4th Edition, recommends: “Do not use hand signs: nearly every hand sign is offensive somewhere.”
- Put index markers and other hidden text at the end of sentences or paragraphs. Hidden text that breaks up a sentence fragments the text in the translation memory and makes the translator’s job more challenging.

PROVIDE COMPLETE BLUEPRINTS

A builder needs detailed blueprints to plan and build a structure. Your translation vendor needs to know all the requirements for translating your content accurately.

Be sure to provide all the relevant collateral that your translation vendor needs to complete the job as quickly as possible. In addition to the source content, you may need to provide:

- Fonts. Be sure to specify (and provide, if necessary) all the fonts needed. If the font is not available in certain alphabets, work with your vendor to determine an appropriate substitution.
- Software strings. If your content includes software strings that are also translated on a device, be sure to provide the string translations to your vendor, so the translations in the text match the actual translations on the device.
- Graphics. If your graphic images are inserted by reference, be sure to provide your localization vendor with all graphic images.
- Style guide and style sheets. Provide your localization vendor with a copy of your style guide and style sheets.
- Work with your localization vendor to create locale-specific style guides that specify conventions such as units of measure, time and date formats, and address formats for specific locales. You may need

variations of your English content to meet unique requirements in the U.S. versus England or Australia, for example. Your Portuguese content may vary depending upon whether your audience is in Portugal or Brazil. The locale-specific style guides specify these details.

- Glossary. Provide a glossary of key terms to help the translators understand your product.
- Trademarks and service marks. Provide a list of trademarked terms and service marks, and specify how they should be handled - these terms are typically not translated.

REFERENCES

"Authoring for Translation and Localization." (Quick Reference Card). Vasont Systems.

Coe, Mariana. Human Factors for Technical Communicators (John Wiley & Sons, Inc), 1996.

"Content Internationalization Top 10 List." (White paper). Lionbridge.

"Creating graphics with localization in mind". (White paper). Lloyd International Translations. Tarporley, Cheshire, UK.

"FrameMaker tips for localized projects". (White paper). Lionbridge Life Sciences.

Kohl, John R. The Global English Style Guide: Writing Clear, Translatable Documentation for a Global Market (Cary, NC:SAS Institute Inc.), 2008.

Pietrangeli, Lisa. "Internationalizing Your Content: Authoring with Localization in Mind. Step One: Start with the Words." (White paper). Language Intelligence, Rochester, NY.

Weiss, Edmund. "Twenty-five Tactics to 'Internationalize' Your English." Intercom, May 1998

AUTHOR CONTACT INFORMATION

Mary Dulin
Owner
Sound View Technical Writing
Everett, WA 98203
MaryDulin@hotmail.com

AUTHOR BIOGRAPHY

Mary Dulin earned a certificate in Technical Writing and Editing from the University of Washington in 1996. After prior careers as a social worker and computer programmer, Mary found her niche as a technical writer, helping users learn about new technology.

As an award-winning technical communicator, Mary has developed technical publications for products ranging from networking hardware to gene expression software. She is currently a freelance technical writer and the owner of Sound View Technical Writing, where she specializes in authoring medical device user instructions using FrameMaker.

As the Technical Publications Manager at Physio-Control, Mary was a hands-on manager responsible for authoring the publications for several defibrillator product lines and localizing the content into 32 languages.

Mary is a Senior member of STC. She has served in various positions for the Puget Sound chapter, including Competition Chair. She is a long-time judge in the local and international competitions.

Mary has worked with various localization vendors. She is currently developing medical device operating instructions that will be localized into 38 languages.

MANAGING TECHNICAL KNOWLEDGE: CONTRIBUTING FACTORS TO KNOWLEDGE LOSS

Liz Herman, PhD, PMP

Knowledge loss, or organizational forgetting, is often overlooked in knowledge management frameworks, yet it costs organizations money, personnel, efficiency, and customer service. The purpose of this mixed model case study was to understand, using a model of forgetting as the conceptual framework, where and why loss occurred and to examine performance implications when technical knowledge is not managed. Contributing factors included culture, leadership support, and limited follow-through from lessons learned.

Recommendations include establishing a formal disposal process to remove outdated knowledge from knowledge management tools and establishing incentives to encourage employees to contribute knowledge, which can lead to higher staff engagement of those tools and improved customer service. Research was conducted and completed in 2009. A follow-up with the case under study was made in 2012 to explore its ongoing management of technical knowledge.

Knowledge loss, or forgetting, is a fact of life. People forget things. Most of the time, knowledge loss is accidental, but some of the time, it is purposeful. People might choose not to accept new knowledge and, through this choice, forget the new knowledge because of disuse or because of the very act of subverting or dismissing the knowledge (Martin de Holan et al., 2004). In personal situations, knowledge loss can cost people an extra trip to the grocery store, a missed birthday, or a late fee. In work situations, knowledge loss can accumulate so that it impacts an organization's overall knowledge capabilities (Martin de Holan et al.). Despite studies that demonstrated a clear relationship between knowledge loss and organizational performance, knowledge management frameworks fail to incorporate knowledge loss as a key component or dimension of the framework (Argote et al., 2003; Hsu & Shen, 2005). The problem addressed in this study concerned the low awareness that organizations have of how knowledge loss occurs in their environment, how it impacts their performance, and how its omission in the routine practice of knowledge management by the organization creates knowledge gaps.

ABOUT THE CASE

The contact center was selected as the organization under study because the concept of knowledge loss had not been explored in this particular type of organization and because the contact center is a highly dynamic and knowledge intensive environment currently undergoing prolific changes regarding knowledge complexity and performance measurement, which offered an opportunity to explore the intersection of knowledge management, learning management, organizational culture, organizational change, and organizational performance.

Located in the Midwestern United States, this contact center comprises 1 manager, 2 supervisors, 1 team lead, 8 agents (called product support specialists), 2 knowledge team members, and 4 systems support technicians. This contact center operates as an inbound information services/technical support helpline for state-based educational services software projects. It supports over 120 different projects. Rather than specializing on a handful of projects, the agents are trained to be generalists and support all projects. This contact center does use a knowledge management tool on its agents' desktops called Kaidara Advisor, a product of Kaidara Software, Inc. Use of a knowledge management tool was not required for participation in the study, but it did offer the ability to infer relationships with other evidence in the case.

Qualitative data collection occurred through semi-structured individual interviews with the participants, through onsite observation of live inquiries through the contact center channels and the agents' use of desktop tools, job aids, and knowledge management support structures, and through review of some documentation that included training materials and e-mails from management and other agents. The qualitative data collection efforts were focused on answering these two research questions: Where did knowledge loss occur at the individual and/or organizational level? What were the contributing factors? In total, 20 participants were interviewed and 16 hours of observation were completed.

Quantitative data collection occurred through receipt of both productivity-based and quality-based performance metrics captured for the contact center for a specific period in time. The quantitative data collection efforts were focused on answering the third and final research question: How did knowledge loss influence the contact center's productivity and quality metrics? The quantitative data collected for this study were used as a point of triangulation for the qualitative data. As such, the data were not empirically tested against a hypothesis or null hypothesis.

CASE STUDY RESULTS

The primary research question asked about knowledge loss in the contact center and the operational impacts. In this contact center, knowledge loss occurred primarily because of failure to capture and resistance to unlearning.

- Subject matter experts failed to provide knowledge to the contact center and agents failed to adopt processes to add knowledge to the Kaidara knowledge management system and there was no mutual accountability.
- Neither the subject matter experts nor the agents were incented by the organization to capture and share knowledge. That is, they received no positive reinforcement, no financial incentive, and no performance incentive to contribute.
- Memory decay and bad habits contributed minimally to knowledge loss.

Because attrition was low, participants were able to rely on the tacit knowledge of their peers when they forgot how to complete a task.

Keeping outdated knowledge in its Kaidara knowledge management system was a bad habit practiced by the contact center and sometimes participants did not check Kaidara before providing responses to customer questions, which resulted in callbacks to customers to correct erroneous answers.

- Knowledge loss occurred at both the individual level and the organizational level.

At the individual level, there was attitudinal resistance to change. Subject matter experts did not want to share knowledge because it reduced their power and took away the limited on-the-job recognition that they received. Agents did not regularly take initiative to recommend or write knowledge articles.

At the organizational level, standard operating procedures were lacking for project start-ups and contractual uniformity.

Contributing factors to knowledge loss included culture, leadership support, the breadth of projects supported, and limited follow-through from lessons learned.

- The culture seemed to support the division between the contact center and the program teams, which fostered knowledge loss. Although leadership support was strong at the contact center level,

executive level support seemed limited and there was no executive level customer service position to promote the contact center work and focus on better performance outcomes for operational efficiencies.

- The sheer volume of projects supported contributed to knowledge loss. With 120 projects, memorization was not possible; therefore, it was necessary to have a knowledge management database from which to access knowledge.
- Continued issues with project implementations showed that lessons learned from previous events were not institutionalized or made standard.

Because knowledge was not readily available to the agents, first call resolution was lower than targeted while average handle time and after call work time were higher than targeted. Quality metrics related to call accuracy and customer satisfaction were absent thus leaving a sizable gap in performance knowledge.

CONCLUSION

Managing technical knowledge requires a blended stream of attention to technology, culture, and people (Davenport & Prusak, 2000; Nonaka & Takeuchi, 1995). While technology and leadership can foster knowledge transfer, organizations cannot solely rely on one or the other and this study upheld that premise. The contact center had implemented a knowledge management solution that was linked directly to its ticket tracking software that made it easy for the agents to search for knowledge while they were servicing the customer. Moreover, the leadership support in the immediate contact center was high and knowledge transfer within the immediate contact center was occurring with some frequency. To operate successfully, however, the contact center relied upon knowledge transfer from various program teams within the organization and that was not happening. Impeding this transfer of knowledge to the contact center was a cultural resistance to unlearning and contract standardization that permeated the program teams and the organization at large. Therefore, although technology was in place and leadership support was present in the immediate contact center, the culture worked against the knowledge transfer process, which led to knowledge loss in this organization.

RECOMMENDATIONS

This study offers researchers and practitioners a conceptual framework from which to operate in order to pinpoint areas of knowledge loss. Applying the framework to additional studies or applying the framework in practice may lead to more meaningful understanding of accidental and intentional knowledge loss and create a body of knowledge on ways in which loss can be minimized, whether that be through technology, culture, human capital, or some combination thereof. Further research needs to show more empirically the connection between knowledge loss and organizational performance.

Examining what motivates employees to share knowledge or populate information and data into knowledge management tools is also worthy of further study. Determining success factors and best practices specific to knowledge workers and knowledge intensive organizations can position chief knowledge officers, customer service executives, and management staff to quickly implement an effective reward and recognition program. This could lead to a higher number of employees submitting content into an organization's knowledge management tool and lead to an increased exchange of knowledge throughout the organization. Moreover, organizations may find that they can implement a program at minimal cost because employees desire psychological and social benefits over financial benefits.

CASE STUDY FOLLOW-UP

In 2012, the researcher contacted the case study sponsor to follow-up on the case and determine what changes, if any, had occurred in the contact center related to knowledge management, organizational culture, and employee and customer satisfaction. In a two-hour discussion, the case study sponsor, who is the manager of this contact center team, and a team lead provided the following updates:

- There is now an increased focus on tracking customer satisfaction. Because one contract required customer service satisfaction surveys, that practice has now been implemented across most of the programs supported by this team.
- The knowledge management sub-team has been largely disbanded. Labor hours for this group to maintain the knowledge base were overestimated and the work did not materialize. Two members of this knowledge management sub-team were reabsorbed as contact center agents. The team currently allocates 2.5 full time employees to managing the technical content. They are focusing on quality and greater efficiencies with how to manage this content.
- Kaidara is still in use as the knowledge management tool, but there are discussions to move to Salesforce's Service Cloud product.
- The relationship between the subject matter experts and the contact center team has improved. A contact center team lead meets biweekly with program teams (subject matter experts) to maintain a positive relationship, proactively seek knowledge for the knowledge base, and provide education on entering technical knowledge into Kaidara. In addition, some personnel changes occurring through natural attrition have also improved the relationship.
- Personnel changes have occurred at the executive level. The case study sponsor has had at least two different managers since the initial case study research was conducted in 2009. The current executive manager is more focused on customer service outcomes. Rather than focusing on the organization changes occurring at the executive level, however, the case study sponsor moved the team forward as an entity on its own.
- Contact center agents have established Individual Development Plans (IDPs) and are offered more training. An employee satisfaction survey is being administered every Spring.
- Performance scores are shared with individual employees. They are used as a tool to improve performance, but the team is still working on consistency issues.
- A call recording system is being considered; 20% of the calls would be captured and recorded (audio/video).

Although further analysis can be done on the information obtained through this follow-up visit, the case study sponsor has moved forward largely as a self-directed team and implemented new processes like the customer service satisfaction and the biweekly meetings with the program team (the subject matter experts) to move the team to better organizational performance with the way technical knowledge is managed. Despite changes in executive leadership and a somewhat static organizational culture, focus on improving the employee and customer experience has continued and been successful.

REFERENCES

Argote, L., McEvily, B., & Reagans, R. "Managing knowledge in organizations: An integrative framework and review of emerging themes." *Management Science*, 49(4), April 2003.
<http://mansci.journal.informs.org/content/49/4/v.abstract?sid=5a3d6621-4b38-4944-b181-a3516879b0c2>

Davenport, T., & Prusak, L. *Working knowledge. How organizations manage what they know.* Boston: Harvard Business School Press, 2000.

Herman, E. M. *Specialized knowledge area module VI: Organizational change models.* Unpublished manuscript. Walden University, Minneapolis, MN, 2007.

Herman, E. M. *Exploring Knowledge Loss in the Contact Center: Key Loss Areas, Contributing Factors, and Performance Metric Implications.* Walden University, Minneapolis, MN, 2010.

<http://gradworks.umi.com/3379876.pdf>

Hsu, S., & Shen, H. "Knowledge management and its relationship with TQM." *Total Quality Management*, 16(3), 2005. doi: 10.1080/14783360500054111

Martin de Holan, P., Phillips, N., & Lawrence, T. B. "Managing organizational forgetting." *MIT Sloan Management Review*, 45(2), January 2004. <http://sloanreview.mit.edu/article/managing-organizational-forgetting/>

Nonaka, I., & Takeuchi, H. *The knowledge-creating company.* New York: Oxford University Press, 1995.

AUTHOR CONTACT INFORMATION

Liz Herman
65 E Weston Drive
North Liberty, IA
319.430.6051

AUTHOR BIOGRAPHY

Liz Herman is a knowledge management and health policy professional with 19 years of experience in technical writing, content and training development, and strategic communications. She currently manages a health policy and strategy team for General Dynamics Information Technology in Coralville, IA. Her job involves transforming complex healthcare information into accessible, easy-to-understand, bite-sized pieces of business intelligence. Liz has been a member of STC since 1997. She received a Ph.D. in Knowledge Management from Walden University and holds a graduate degree in English from Iowa State University and an undergraduate degree in Journalism from the University of Iowa. She is also an Adjunct English Professor at Kirkwood Community College and DeVry University.

SME — ME — E SCREENCAST CREATION AND COLLABORATION

Christopher Bartis

Screencasts, also known as a video screen captures, are digital recordings of what you see and what happens on your computer screen. The name makes it comparable to "screenshot", but rather than being a static image a screencast is more like a documentary movie, complete with voice narration. Where technical writers mostly got their messages across using text, today's technical communicators work with all types of media.

The elements of the title refer to the SME, the Subject Matter Expert, ME, well, that's actually, *YOU*, and the E is for electronic distribution and publishing. I will discuss how we need to enlist the help of subject matter experts in screencast creation, how you can help them to get started, and some tools that you and they can use. Imagine a scenario where solution experts, whether developers, testers or other colleagues feed the you the technical communicator with raw video input - resulting in an effective, distributed process. Finally, I'll give you a few tips for publishing your completed creation.

SUBJECT MATTER EXPERTS

Traditionally, technical communication is a collaborative endeavor, and screencast creation should be no different. Technical communicators are not expected to have the same level of technical knowhow as experts in their field are, and just as input is needed with written documentation, it is also needed with digitally recorded information. While organizations might be skeptical of screencasting and the time and resources it takes in the first place, the input needed for creating "raw video" is no different than any type or form of input, and in actual fact, after a little time spent on the initial setup, it may be even quicker for the SME. Finding the right SME is as important as ever, here, perhaps with some added requirements that the expert be interested and willing to take part in this new media and learn a few new things him/herself.

THE WORKFLOW

The stages for your SME to help you and start creating raw video:

1. Download and configure two free programs.
2. Create the video.
3. Make the video available to the screencasting expert
4. You, the Camtasia expert, edit the raw video, and add audio narration.
5. Publish the completed screencast.



Figure 1: The SME → ME → E workflow

THE TOOLS

There are a number of recording tools and programs out there, but for our purposes I am going to focus on a couple of products by a company called TechSmith, and one more, freeware, product.

This approach requires that each SME who is going to help you with "raw video input" download and install Jing, a free, easy-to-use screen recording tool. Jing works hand-in-hand with Camtasia Studio, since both products are developed by TechSmith. The main benefits of using Jing is that it can be used as raw input into the Camtasia video creation process and it's free. Your organization only needs to pay for one license, for one program, and you get to be the only expert in the use of that program!

In order to use the free Jing version you must create a www.screencast.com account. This will come in handy in the later stage of sharing the video. To start off, Jing needs a few small configuration tweaks. The most important thing we discovered was that under More > Preferences > Video Format, you should select SWF. For some reason, selecting MPEG-4 (even though that is the format we save our completed screencast in) results in VERY large file sizes when the video is imported into Camtasia.

Enabling video hotkeys is also a good idea. You should specify a Capture hotkey, and make a note that the pre-defined hotkeys: Start F6, Stop F7, and Pause/Resume F8. Pausing/Resuming comes in very handy when recording videos.

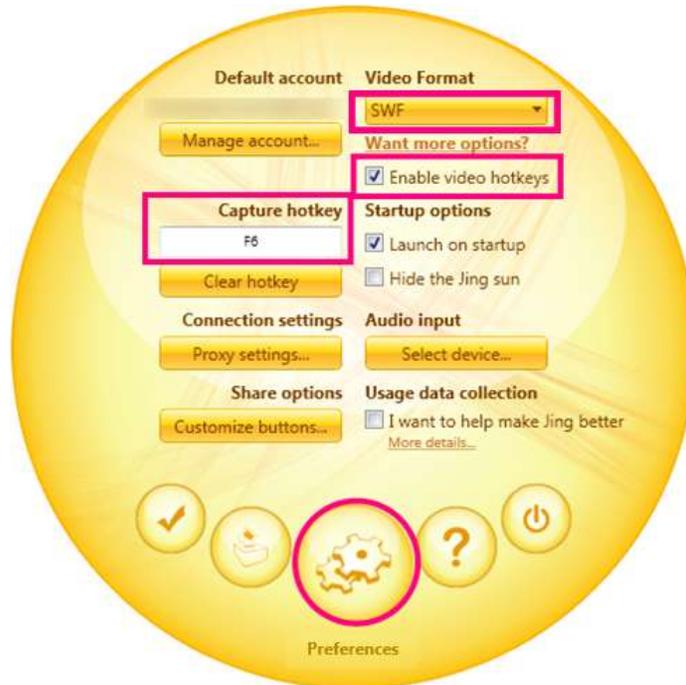


Figure 2: Jing Preferences

Another very useful tool to have, which will help you to automatically set the size of any window, computer program or web browser, is Sizer, available from <http://www.brianapps.net/sizer/>. The notes on their web page state that the new version will support 64-bit machines, but I've been running the current version on a 64-bit Windows 7 machine for some time now and never experienced any difficulties.

Before configuring Sizer, you need to think about your ideal recording size. Here is a list of some standards:

- 4:3 resolutions:
 - 1600 x 1200
 - 1400 x 1050
 - 1024 x 768
- 16:9 resolutions:
 - 1920 x 1080
- 8:5 resolutions:
 - 2560 x 1600
 - 1920 x 1200
 - 1680 x 1050

When choosing a resolution and deciding on one which is best for your needs, one thing to consider is whether or not you will use the Zoom-n-Pan functions in Camtasia Studio. We wanted to publish screencasts at a resolution of 1024 x 768 and realized that in order to Zoom-n-Pan, the original "raw" video needs to be larger (otherwise the areas/content you zoom in on will be blurry). In this case, we chose 1400x1050 as the ideal raw video recording size, allowing us to zoom in, roughly 25%. After installing Sizer, you can configure it, by adding one or more resolution presets.

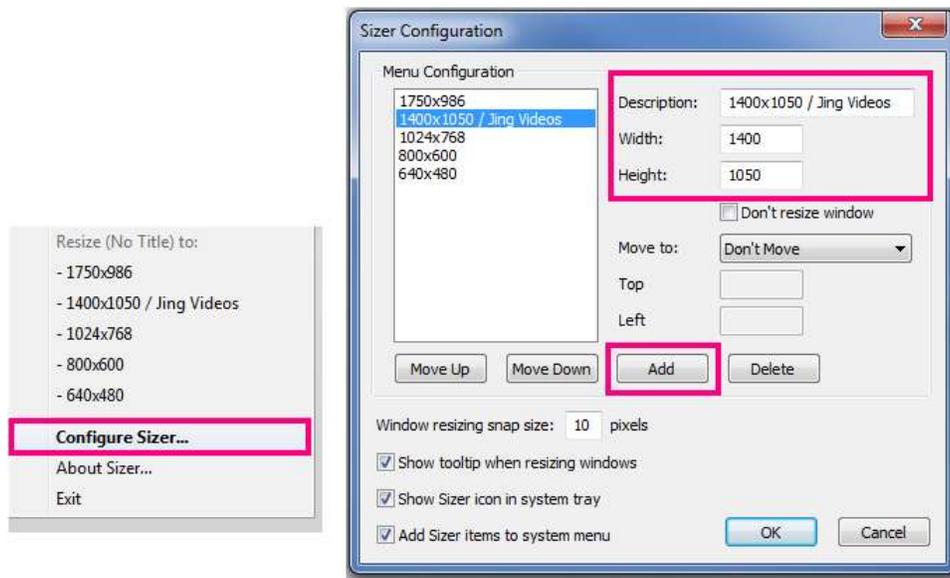


Figure 3: Configuring Sizer

Then when you are ready to start recording, you simply open the program, browser or window you will be recording, right-click the dialog frame and select **Resize > 1400x1050** (or whatever resolution you configured earlier in Sizer).

RECORDING TIPS FOR THE SME

Sit down with your SME and plan the screencast. A script or at least some notes is highly recommended and can simply be a list of steps or items to cover. I'm going to click here, open this, edit this, save my settings, and exit, for example. You might also want to take some time to familiarize the SME with the Jing interface and controls. Show him/her how hovering the Jing crosshairs over the edge of a window locks Jing the resized application and finally how and the recording controller appears, attached to the dialog. I find it important to remind him/her that there is no need to wave the mouse around, circling things with the cursor – it is the job of the Camtasia expert to edit the video and draw circles, underline, zoom in/out pan, etc. If your SME insists on waving that sucker all over the place, you, the Camtasia expert can totally remove the cursor from the final video, which might look a bit odd in some cases, but is still better than making viewers sick. Also, if a video needs to be cut, and the SME is in the middle of the cursor Olympics, it will difficult or impossible to do. Slow, controlled mouse movements are the way to go. As I work abroad, with non-English speaking developers, it is always my voice that ends up in the final video. Seeing as how you will need to edit the video, it is most likely that the voice of the SME will not be present in your case either. Explain to the SME that speaking and explaining what they are doing helps to provide the right video timings and reducing video editing later. This narration can also be used as the basis for the final speaker script and will help the Camtasia expert understand what the SME is trying to demonstrate.

SHARING THE RAW VIDEO WITH THE CAMTASIA EXPERT

When recording is finished, the video can be shared in a variety of ways. While the video can always be saved locally or on a server in a network and then emailed, or put a USB stick, uploading it to www.screencast.com has its advantages. If you choose to use [screencast.com](http://www.screencast.com), make sure the SME checks "Show download link to viewers" and have a look at the other options to see what's best for you.

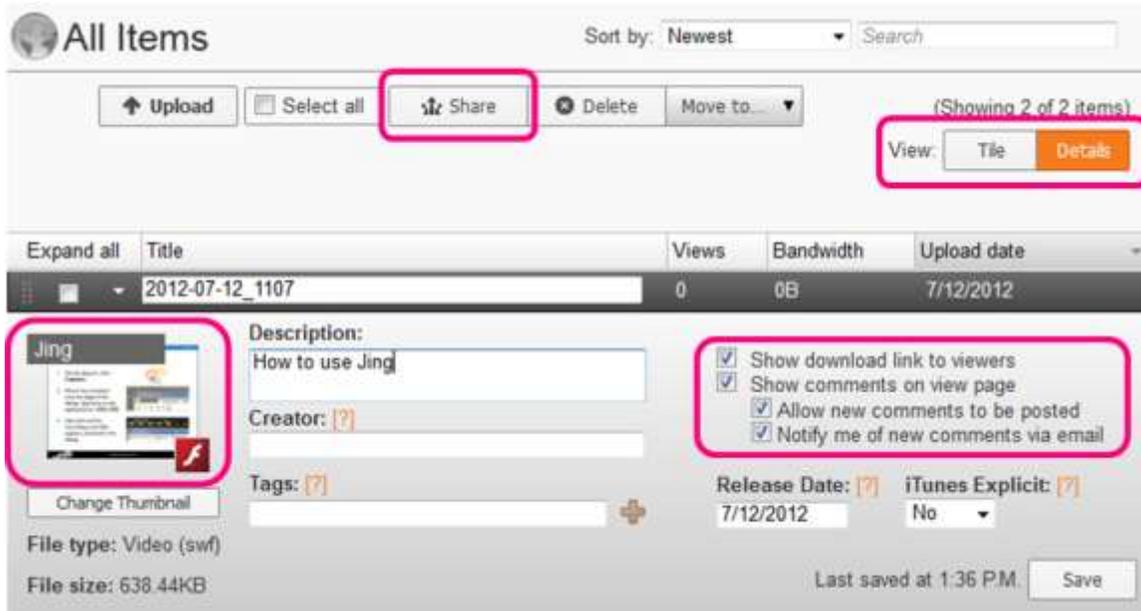


Figure 4: Sharing to Screencast.com

PUBLISHING THE FINAL SCREENCAST

Discussing how to edit your video in Camtasia is beyond the scope of this paper. There are, however, a few things worth mentioning. Any watermark that you initially see because a video was created using the free Jing version will NOT appear once the video is imported into Camtasia. Remember that the original recording dimensions and your publishing dimensions will be different, if you allowed for and used Zoom-n-Pan. For example, our videos captured at 1400 x 1050 are published 1024 x 768, while a video captured at 1750 x 986 would most likely be published at 1280 x 720. Note that there are also presets available for Blackberry, iPhone, etc. Whatever you decide on, it's a good idea to stick to a standard format each time you produce videos, using consistent values, such as always publishing a 1024 x 768, Mp4 video in 4:3 format. By opting to include an HTML page with the video embedded and placing the file(s) on a server, you can easily link to, and embed the screencasts in your online help and even pdfs and other documents.

Setting up customer order solutions

Use this procedure to set up a customer order solution in DOCUMENTS. To process customer orders in more than one language, a separate batch specification is needed for each language.

<ol style="list-style-type: none">Using the Administration module, select Solutions, right-click in the right pane, and click New Solution.With Create from template selected, select Customer order solution, and click OK.For each customer order language:<ol style="list-style-type: none">Create a new document specification, select Copy content from a template and then select one of the available Customer Order document specifications.Open the new document specification and on the Extraction tab:<ol style="list-style-type: none">Delete any fields that do not apply to your needs.Add any necessary fields on the newly created specification, <i>not</i> on the template.Change Extraction > Confidence settings on the fields as necessary. For example, you may want to select Confirm empty or not found for some fields.	
<ol style="list-style-type: none">Create an output method for each document specification.	
<ol style="list-style-type: none">Create a batch specification for the language that contains the new Customer order document specification.	

Figure 5: Video links in online help.



Installing and configuring Knowledge Store

DOCUMENTS Knowledge Store Service results in better extraction of header fields (single fields, for example customer number) and table fields. It is recommended in customer order solutions, and in solutions that process invoices without using the Invoice modules.

Video instructions

To see a video demonstration of the Knowledge Store installation procedure, click the image to the right.



Installing the Knowledge Store Service

Install DOCUMENTS Knowledge Store Service on a PC that has access to a Microsoft SQL Server installation. (Knowledge Store Service supports the same versions of SQL Server as DOCUMENTS.)

Figure 6: Video link in PDF.

RESOURCES

"Screencast." From Wikipedia, the free encyclopedia. [Web link](#)

Lee, Martha F. and Mehlenbacher, Brad. "Technical writer/subject-matter expert interaction: The Writer's Perspective, the Organizational Challenge." [Web link](#)

File:Vector Video Standards4.svg" From Wikipedia, the free encyclopedia. [Web link](#)

Bozarth, Jane. "Nuts and Bolts: Working With Subject Matter Experts" [Web link](#)

AUTHOR CONTACT INFORMATION

Christopher Bartis
Senior Technical Writer
ReadSoft AB
Box 1224
SE-16428 Kista, SWEDEN
+46 73 3378607
christoper.bartis@readsoft.com

AUTHOR BIOGRAPHY

Christopher Bartis, is a Senior Technical Communicator at ReadSoft AB in Kista, Sweden. Chris is part of a team of six technical communicators who produce software documentation in eight different languages. Since joining the company in 2006, he has worked with written documentation, as a technical trainer, and for the last two years has focused on screencasting and video creation

Originally from Connecticut, he started his career as a teacher and has taught English as a Second Language in universities and public and private schools in the US, England and Sweden, where he currently resides. He has an MS degree in Applied Linguistics and a BA degree in English, both from CCSU, and is fluent in Swedish.

TEACHING A COLLEGE-LEVEL EDITING CLASS

E. Jonathan Arnett, Ph.D.

The technical editor's job is changing rapidly. The "traditional" editor's role of marking hard copy with a red or blue pencil has almost disappeared and been replaced by the task of editing multimedia communication products via computer. Thus, the electronically-based skills and literacies that technical editors use in the workplace are what college-level editing classes should teach students.

THE TIMES THEY ARE A-CHANGIN'

Until fairly recently, the college-level class in professional editing took a predictable form; students studied copyediting and proofreading techniques, learned the minutiae of markup, dabbled in document design, reviewed grammar and mechanics, and emerged proficient in all things red-pencil-related, hopefully with a keen eye for detecting errors. Of course, students also learned how to communicate effectively with authors, touched upon soft-copy editing, and may have learned the rudiments of a tagging language, but for the most part, college-level editing classes and textbooks (for example, *Technical Editing* by Rude and, later, Rude and Eaton) focused on skills related to manipulating and producing hard copy.

Now, though, workplace communication media have changed. Soft copy is king, hard-copy markup is nearly moribund, and editors must learn to master the techniques and technologies required to edit multimedia communication artifacts. Accordingly, the college-level class in professional editing must change to reflect this new reality.

DASHING STUDENTS' EXPECTATIONS

In the professional editing class that I teach, most, if not all, students enter the class with misapprehensions about what professional editing actually means. Some students enter expecting a journalism course; there is always at least one member of the campus newspaper editorial staff who hopes to parlay experience editing news stories into an easy "A" or a reporter who wants to move up the ranks to section editor. Other students hope for a semester devoted to reviewing English grammar and mechanics, or they have vague notions that they'd like to be writers and want to learn how to edit their own short stories or novels. The rest of the students enroll because they entertain visions of becoming literary editors and discovering the next J.K. Rowling or Maya Angelou. They think they'll learn how to aid an author with character development, how to craft elegant sentences, or how to refine a poem to reveal its inner, sublime essence.

I do my best to quickly disabuse students of these notions.

The course's first assignment always involves finding out what editors actually do. In this assignment, students search for editing jobs, collect and compile job descriptions, and share their findings with each other. The variety of editorial tasks students discover, a small sample of which appears in Table 1, never fails to amaze them.

Table 1: Sample of Editors' Job Responsibilities

Copyedit texts	Proofread texts	Format documents	Create interactive content
Write captions	Develop learning modules	Transcribe data	Analyze audiences
Acquire book projects	Manage typesetting	Choose backing music	Create advertisements
Implement brand strategy	Produce newsletters	Edit audio	Manage files
Manipulate photographs	Shoot video	Update blogs	Set timelines
Write audio scripts	Negotiate deadlines	Translate documents	Write progress reports
Write feature stories	Choose articles	Ensure continuity	Negotiate contracts
Oversee production processes	Manage single-source tags	Liaise between authors and other editors	Curate user-created content
Evaluate content for accuracy	Oversee art and editing departments	Optimize websites for findability	Liaise between editors and press

COMMON THEMES

Two central questions arise when one considers this sample: If these tasks are what editors actually do at their jobs, what skills do student editors need to gain in order to succeed in the workplace, and how can an editing course teach students those skills?

The answers to both questions lie in the three major themes that emerge from the sample. Editors need

- project-management skills
- interpersonal skills
- technology skills

Project-management skills and best practices for working with authors and other editors can be and are currently taught in college-level editing classes (for example, see Albers and Marsella, 2011; Amare, Nowlin, and Weber, 2011), but technology skills deserve special scrutiny because they constitute the greatest changes to an editing course's content.

In the past, editing-technology skills necessarily involved creating or marking up hard copy documents, but with the spread of computer technology throughout the workplace and the rise of the internet, authors and editors have become used to working in soft copy (Lanier, 2004) and collaborating at a distance (Crognale, 2008). In fact, many modern editors work on purely electronic publications, and few, if any, job ads students have located describe editorial duties that would demand an editor work directly on hard copy.

Accordingly, I suggest that a college-level professional editing course should de-emphasize traditional hard-copy markup in favor of hands-on experience using computer-based editing technologies. In particular, I suggest that editing classes expose students to five common electronic media formats:

- MS Word documents
- MS PowerPoint presentations
- PDF files
- audiovisual media
- websites

MS WORD DOCUMENTS

MS Word is *the* nearly-ubiquitous workplace program; many competitors exist, but MS Word remains the most widely used word processor, and thus, editors must learn how to use it effectively. Specifically, student editors need to learn how to

- create, manipulate, and apply Styles
- use the Track Changes tool for error correction and version control
- create and manipulate comment balloons

MS POWERPOINT PRESENTATIONS

Like its companion MS Word, many popular alternatives—such as Prezi and Keynote—exist to this program, but MS PowerPoint remains the most commonly used slideshow program in the workplace. As such, student editors need to learn how to

- mark up slide shows with the built-in commenting tools
- compare versions of PPTs with the built-in comparison tool
- use the Master Slide tool
- call up and modify slide templates
- choose appropriate fonts and manipulate layouts for maximum readability and effectiveness

PDF FILES

A third type of file common to many workplaces is the PDF, and student editors would benefit greatly from learning how to edit these files. However, teaching students to edit PDF files can be a challenge; students are unlikely to have Adobe Acrobat on their own computers because the program costs \$120 even after a student discount, and the free program they are likely to have, Adobe Acrobat Reader, only contains rudimentary editing tools: a highlighter and sticky notes. Multiple third-party PDF generators and editors are also available, but their functionality is variable.

Even so, student editors need to learn how to

- create PDFs
- mark up PDFs using the basic Acrobat Reader editing tools
- (if possible) comment on and edit text directly using the more sophisticated Annotations, Drawing Markups, and Content editing tools on Acrobat

AUDIOVISUAL MEDIA

In addition to editing skills based on visual elements, modern editors need to be familiar with multimedia presentations that use sound and video. In particular, student editors need to learn how to

- identify and avoid tongue-twisters or words containing plosives, fricatives, or sibilance that can cause sonic artifacts
- provide guidance for voice artists regarding commonly mispronounced words
- edit audio files using common audio-editing tools

- edit video files using common video-editing tools
- budget time for editing multimedia projects

WEBSITES

Because a great deal of modern professional communication takes place on the internet, students must learn to present materials clearly and effectively in electronic media. As such, student editors would ideally learn about HTML 5, Section 508, findability, information architecture, search engine optimization, and taxonomies/folksonomies, but these concerns are likely beyond the scope of a basic editing class. At the very least, then, student editors should learn how to

- read and edit basic HTML code
- read and edit CSS code
- prepare materials for single-sourcing and/or insertion into a content-management system
- tag items with XML
- identify and work with RGB color models as opposed to CMYK color models
- optimize file sizes for web-delivered content, particularly images
- differentiate between presentation standards for hard and soft copy materials

THE REALITY OF THE SITUATION

At present, teaching the editing class described here remains a goal for the future. Hard-copy markup skills are “zombie skills”—nearly, but not entirely dead—so while including a unit on traditional markup symbols still makes sense, there is no need to drill students on intricacies of red-pencil markup all semester long. Also, because of time constraints, it may not be possible to cover all five electronic media formats in a single semester. However, in order to prepare students for real editing jobs, instructors of editing classes must incorporate electronic media to a greater extent than they have before and provide students with hands-on practice in editing computer-based communication artifacts.

RESOURCES

Dayton, David. “Electronic Editing in Technical Communication: The Compelling Logics of Local Contexts.” Technical Communication 51.1 (February 2004): 86-101.

Dayton, David. “Electronic Editing: A Model of User-centered Technology Adoption.” Technical Communication 51.2 (May 2004): 207-223.

Eaton, Angela. “Editing in the Workplace.” Intercom 53.10 (December 2006): 27.

Hayhoe, George F. “The Future of Technical Writing and Editing.” Technical Communication 54.2 (August 2007): 281-282.

Hart, Geoffrey J.S. “Editing Web Pages.” Intercom 50.6 (June 2003): 38-39.

Hart, Geoffrey J.S. “Editing Web Pages: A Second Look.” Intercom 51.1 (January 2004): 36-37.

Hart, Geoffrey J.S. “Special Needs: Editing Tables and Graphics.” Intercom 49.4 (April 2002): 37-38.

Thomas, Shelley. "More than Grammar: Expectations of Technical Editing New Hires." *Intercom* 56.5 (May 2009): 26-27.

REFERENCES

Albers, Michael J., and John F. Marsella. "An Analysis of Student Comments in Comprehensive Editing." *Technical Communication* 58.1 (February 2011): 52-67.

Amare, Nicole, Barry Nowlin, and Jean Hollis Weber. *Technical Editing in the Twenty-First Century*. (Boston, MA: Prentice-Hall), 2011.

Crognale, Heather. "Long-distance Editing." *Intercom* 55.7 (July 2008): 17-19.

Lanier, Clinton R. "Electronic Editing and the Author." *Technical Communication* 51.4 (November 2004): 526-536.

Rude, Carolyn D., and Angela Eaton. *Technical Editing*, 5th ed. (Boston, MA: Longman), 2011.

AUTHOR CONTACT INFORMATION

E. Jonathan Arnett
Assistant Professor of Professional Writing
Kennesaw State University
1000 Chastain Rd. MS 2701
Kennesaw, GA 30144
678.797.2523
jonathan.arnett@kennesaw.edu

AUTHOR BIOGRAPHY

Jonathan Arnett is an Assistant Professor in the Kennesaw State University Department of English, where he teaches technical writing, professional editing, and first-year composition. He received his MA in Technical Communication and Ph.D. in Technical Communication & Rhetoric from Texas Tech University, and his research interests include rhetoric of science and professional editing course pedagogy.

THE RETIREMENT OF TECHNICAL COMMUNICATORS — MAKING RETIREMENT AN ADVENTURE

Bill Leavitt

Retirement in the 21st Century is much different and problematic than it was for our parents and grandparents. People are retiring earlier, and usually have better health and fitness levels than people who retired 50 years ago. As a result, there are new issues that should be discovered and faced to help today's retiree take advantage of all the activities and excitement that is available now. Also, mental and emotional health has become more important for a retiree to have a happy, healthy, successful, fulfilling, active retirement.

“Retirement” is a word with many meanings. Traditionally, it is a time of rest and relaxation—earned from a lifetime of work. To some people, retirement is a time to focus on family and friends—particularly grand children. To others, it is a time to consider relocating to a more desirable climate and perhaps living in a retirement community.

In today's society, people are often retiring in their 50's. These people are obviously younger and usually healthier, and thus may need to be more active.

Twelve years of research and interviewing people planning for retirement and those who have already retired has led to interesting insights into what retirement is like in the 21st Century.

Interviews with pre-retirees indicate that about half of this group was not looking forward to retirement. They didn't have solid plans, were concerned about the unknown, were unsure whether they could afford retirement, and/or hated leaving a job or profession they liked.

Those who had retired were often disappointed with retirement, were bored, felt unfulfilled, were lonely, were depressed, etc. Thus it appears that many of those who are retired have not discovered a retirement that is what they hoped for.

As a technical writer, I approached the issues of retirement as a good technical writer tries to become expert in the subject matter to be written about. In this case, it started with research into why people are satisfied or not completely satisfied with their retirement. This led to the following issues:

1. People often define themselves by what their job or profession is. When they retire, they sometimes become depressed or at least disappointed because they no longer have an important purpose.
2. They have no meaningful plans for the future.
3. They are not physically fit enough to do the activities they would like to do.
4. They are lonely because many or most of their friends were associated with work and are no longer available to them.

It thus appears that a successful retirement needs challenges, purpose, fulfilling and productive activity, and excitement. Particularly for those planning for retirement, they must learn the issues that they may face and prepare adequately for them. For those who have already retired, they must create new goals and search for activities that will provide for challenges, purpose, fulfilling and productive activity, and excitement.

Choosing activities that suit the particular individual is not simple. Everyone is different. Everyone reacts differently to various activities. It may require some time to try activities that may meet their needs. It is common for one to try many different activities before finding one or more that provides the kind of excitement, fulfillment and satisfaction that will make a happy, active retirement.

Considering all the research and interviews, I concluded that to have a happy, healthy, successful, fulfilling, active retirement, you must first be concerned about three kinds of fitness:

- Physical fitness
- Mental fitness
- Emotional fitness (most important)

Emotional fitness is most important because it deals with one's feelings of well being. Next is mental fitness, because some activities have to be chosen to "work" your brain; that is, create new pathways of thought. Increased brain activity can help prevent Alzheimer's disease, dementia, and other types of mental deficiency. Physical fitness doesn't mean being able run a marathon or ride a bike 50 miles, but some of the fun things in life require having enough energy to walk a little or have the energy for evening activities.

Retirement: Life's Greatest Adventure, by Bill Leavitt (STC Fellow), is a new book that addresses all these issues, provides guidance for how to select activities that are fun, fulfilling, satisfying and active, and yet provide for the kinds of fitness that retirees and seniors need to be happy and busy. The book has work sheets to help select activities and goals, and descriptions of over 100 activities that are appropriate for an exciting, active retirement.

The book also deals with many of the modern issues of an active retirement, such as the changing relationship with one spouse that results from spending more time together, deciding whether to move to a more desirable climate or a retirement community, finding new friends that share one's new life style, and issues of part time work for the retired person.

Although the issues facing a retiree are important and serious, they are often presented in humorous and interesting ways to make the book more readable.

RESOURCES AND REFERENCES

Leavitt, Bill. Retirement: Life's Greatest Adventure, Valparaiso, IN: Write On Technical Writing, Inc. 2012.

AUTHOR CONTACT INFORMATION

Bill Leavitt
President
Write On Technical Writing, Inc.
417 Killarney Lane
Valparaiso, IN 46385
219.464.0595
WriteOn55@aol.com
www.RetirementLifesGreatestAdventure.com

AUTHOR BIOGRAPHY

Bill Leavitt is a technical writer living in Valparaiso, Indiana. He earned his degree from Purdue University (West Lafayette) in technical writing in 1964. He has worked in the field of technical writing for 48 years.

After retiring from a large corporation in Chicago twelve years ago, he did technical writing consulting for many companies. His consulting company is named Write On Technical Writing, Inc. He recently taught technical, business and report writing to engineers part-time at Purdue University Calumet. Bill is also a past president and Fellow of the Society for Technical Communication and a long-time member of the Indiana Society of Chicago.

Bill has written books on construction (*Gypsum Construction Handbook*) and local history (*Hail Noble Washington*). Bill recently completed a book about the joys of retirement, entitled *Retirement: Life's Greatest Adventure*.

THERE'S NOTHING SIMPLE ABOUT SIMPLIFIED TECHNICAL ENGLISH

Bernadette Koontz

Discuss Simplified Technical English (STE), its elements, rationale and benefits for using STE, and guidelines for writing in STE as a progression topic.

Simplified Technical English is an international writing specification STE100 defined by the AeroSpace and Defence Industries Association of Europe (ASD) which uses a finite number of specified verbs and terminology that can be understood globally for documentation pertaining to land, sea and air equipment and applications.

The limited vocabulary, verb choice, and style are intended as a controlled language that can be more uniformly understood by other nationalities without ambiguities and complexities associated with the full English language.

By limiting the lexicon to a finite set of verbs, the goal is to decrease ambiguity and nuances across languages for easier, uniform, and cost effective- translations.

Some companies have mandated that document deliverables produced by vendors be written in STE. Airbus imposes a very stiff \$1 million penalty if a vendor does not write documents for Airbus in STE. The U.S. Army requires their equipment documentation to be written in STE. The Federal Aviation Administration (FAA) also requires airworthiness manuals to be written in STE.

GUIDELINES FOR WRITING IN STE

Guidelines for writing of STE include:

1. Specified limited verb list
2. Short sentences in active voice (usually starts with a action command type verb...e.g. Complete, Examine, Enable, Put, Do, etc.)
3. Use only verbs on approved verb list
4. One topic or idea per sentence
5. Avoid use of gerunds (words ending in "ing")
6. Handout - additional rules-and comparative examples of text written in normal English and in STE
7. The ASD has a board that meets to update rules and verb list

TOOLS & SERVICES

ASD Web address: <http://www.asd-stan.org> for copy of STE Rules

STE Software Checker Tools

- Etteplan/tedopres HyperSTE – Checker Software
<http://www.simplifiedenglish.net/>

<http://www.tedopres.com/hyperste-simplified-technical-english>

- SMART Communication MAXit Checker Software

<http://www.smartny.com/simplifiedenglish.htm>

Writing Services

- TechScribe Service (United Kingdom)

<http://www.techscribe.co.uk/techw/asd-simplified-technical-english.htm>

AUTHOR CONTACT INFORMATION

Bernadette Koontz
Supervisor, Technical Publications
Lycoming Engines
652 Oliver St.
Williamsport, PA 17701
570.327.7337

AUTHOR BIOGRAPHY

Bernadette has been a practitioner in the field of technical writing for over 30 years, with emphasis on document management and heuristics to help companies build a profitable streamlined infrastructure to efficiently produce cost effective, compliant, and timely document deliverables that achieve customer satisfaction and profitability for companies in global and defense markets.

USING MENTORING PROGRAMS TO COLLABORATE WITH INDUSTRY

Herb J. Smith

Mentoring programs can be effective ways for industry to build strong ties with academic programs. Mentoring programs have been around for a long time in a number of academic disciplines, but these programs are a relatively recent development for technical communication programs. Mentoring programs are beneficial for both students and industry professionals who serve as mentors. In short, it's a two-way street. Students find that mentoring programs help extend learning beyond the classroom and industry professionals often enjoy the experience of giving back. Although mentoring can be defined in several different ways, mentoring is defined here as a way for experienced technical communicators to share their experiences with less experienced protégés or mentees. This progression session will cover both formal and informal mentoring models, noting the benefits that mentoring provides for mentors and mentees. The session will also include the results of an informal survey of mentors who commented on their mentoring experiences. Hopefully, the session will help you decide if mentoring is for you.

FORMAL MENTORING PROGRAMS

Several academic programs in technical communication have set up mentoring programs, partnering with local STC chapters. One good example is the mentoring program between the University of Central Florida and the STC Orlando Chapter. This mentoring program provides seniors, graduate students, and recent graduates with STC mentors who provide career guidance. The program is highly structured and includes three positions: program coordinator, mentor, and mentee. The mentee must join STC and the local STC chapter. The STC member mentors the student for two semesters and usually provides feedback on the mentee's resume and portfolio, in addition to providing tips on how to develop a networking plan. At the end of the second semester, the mentor provides feedback to the program coordinator on the mentoring experience. As you can tell, setting up a formal mentorship requires a lot of planning and coordination with a local STC chapter for the program to be successful. There is, however, another type of mentoring program that is easier to set up—the informal mentoring program

INFORMAL MENTORING PROGRAMS

The mentoring program at Southern Polytechnic State University pairs each senior who is in the senior capstone course with a mentor who is a professional communicator. The senior capstone course requires each student to design and then create both an e-portfolio and paper portfolio of that student's best projects. The course also requires each student to complete a career unit. Because the course covers career options, interviewing techniques, resume writing, and the job search process, the professional communicator, serving as mentor, provides his or her mentee with career advice and feedback on the mentee's portfolio. The mentoring program requires that the mentor and mentee complete at least three activities. Two of the activities are required: the initial face-to-face interview and the mentor review of the mentee's e-portfolio. The mentor and mentee decide on a third activity from the following list:

- Feedback on the mentee's resume

- Review of the mentee's portfolio design, or
- Review of the mentee's paper portfolio

Mentors and mentees often are involved in more than the minimal requirements and have the opportunity to build a professional relationship that extends beyond the semester-long course. We have had mentors hire their mentees after graduation. Mentors are also invited to attend an Open House where mentees present their e-folios and paper portfolios to the public.

WHAT ARE THE MENTOR BENEFITS?

An informal survey of mentors revealed several benefits. These are listed below:

- Sharing my experiences with others
- Giving back
- Forcing myself to put in place some of the very advice I was
- giving the student
- Staying current with the tools that are now being used to create
- portfolios
- Helping others start a career

IS MENTORING FOR YOU?

Hopefully, the answer to this question is "yes." Mentoring is a two-way street providing both the mentor and the mentee a number of benefits. Much of the mentoring can also be done through email or via technologies like Skype. Many of the mentoring benefits are intangible and are probably best summed up by the phrase "giving back."

RESOURCES

Bosley, D. "Collaborative Partnerships: Academic and Industry Working Together." Technical Communication 42.4: 611-619.

Dimick, S.A. "Electronic mentoring: Benefits and Rewards." (Proceedings, STC, 1996).

Duin, A.H. and D. Starke-Meyerring. "Professional Communication in the Learning Marketplace: A Call for Partnering." Journal of Business and Technical Communication 17.3: 346-361.

Epp, B. "Developing a Chapter Mentoring Program." (Proceedings, STC, 1996).

Smith, H. "Project Portfolio Survey." (2011).

Society for Technical Communication Orlando Chapter education Committee. "Mentoring Programs." (2011).

<http://www.stc-orlando.org/education/mentor/mentor.asp>

AUTHOR CONTACT INFORMATION

Herb J. Smith
Professor
English, Technical Communication & Media Arts
Southern Polytechnic State University
Marietta, GA 30060-2896
678.915.7475

AUTHOR BIOGRAPHY

Herb J. Smith is Professor of English and Technical Communication at Southern Polytechnic State University where he currently serves as Director of Departmental Undergraduate programs. He has been teaching technical communication for more than 30 years. In 1998, he was Gastprofesseur of Technical Writing at Fachhochschule Merseburg, Germany. He was also principal author of a National Science Foundation grant.

Smith has published widely in most of the major technical communication journals. Portfolios for Technical and Professional Communicators, which was co-authored with Dr. Himes-Korn, provides detailed information about paper and electronic portfolios for technical communicators. Smith has recently written a book chapter on mentoring and internships for students pursuing technical communication degrees.

WHY TECHNICAL WRITING IS ALWAYS ABOUT TRAINING

Mary Ryba Knepper

Many technical communicators “move into” training (expanded in this discussion to include “instructional design:”) to upgrade or expand their career; conversely, trainers become documentation specialists as champions of communicating content. In this progression, I propose that one informs the other; in some instances the line blurs and disappears. Drawing parallels between core principles in technical communication and training enriches the practice of both. They are both committed to serving their audiences in pursuit of a business goal and maintaining high ethical standards. Both rely for excellence on the habit and skill in analyzing an audience, purpose, business situation, and communication context; both require clear development, execution, and review processes; both depend upon collaboration from various sources and areas; they experience the same constraints and offer similar career advancement opportunities. The simultaneous practice of both areas enriches one’s professional experience.

INTRODUCTION

Despite the obvious difference of technical writers' delivering documents on line or through a printer and trainers' face-to-face interaction in a classroom or webinar, technical communication and training are similar enough at their core that applying the principles of one to the other elevates the quality of both. The proof is, first, in observed practice: Many technical communicators move into training to upgrade or expand their careers; trainers can find themselves becoming the documentation arm of an enterprise to support the course work. Additional proof, which is the remainder of this discussion, derives from a comparison of their core principles.

One caveat based on personal bias as both a "technical communicator" and a "trainer": For this discussion, I define "trainer" to include "instructional designer," with emphasis on *the training experience designed for an applied, practical purpose*. Another caveat is I have excluded instructional design for self-paced instruction from the discussion because it rests at the apex of both areas, requiring excellent technical communication and instructional design skills.

SHARING THE SAME PURPOSE: SERVICE

Thinking deeply about one informs the other. Specifically, documentation and training have the same objectives: They serve peoples' needs in specific behavioral ways. If people can do something to accomplish an objective—through a series of objectives that help them reach goals—the documentation or training was successful. The difference is the medium: In one, the medium is a combination of visual and verbal elements in a text, delivered via paper or electronics; in the other, a person uses his or her self as the communication vehicle in conjunction with various media, props, and locations. In creating a technical document, the communicator serves the audience's needs and purposes, not the product's, company's, or context's. If the text serves anyone or thing but the reader/user, it is not technical communication; it is an advertisement, a marketing piece, a legal or administrative document, a review or critique, a thought experiment, a work of art, or something else. In standing before a group, a trainer that focuses on his or her performance has lost the game; the concern should be the learners' performance. Both technical communication and training are at essence selfless exercises. This is the first area of overlap.

The STC official website describes technical communication this way: "What all technical communicators have in common is a user-centered approach to providing the right information, in the right way, at the right time to make someone's life easier and more productive." I could not find an equivalent statement on training from an official website, but one could legitimately substitute "trainers" for "technical communicators."

AREAS IN COMMON

Principles: The two areas overlap in their core principles:

1. Serve the reader/learner.
2. Maintain ethical and professional standards.
3. Serve a broader business function to meet business goals by enhancing performance.
4. Present information to conform with how people learn, in the timeframe they need it, in amounts that can be assimilated.

Process: They overlap in the creation-delivery-evaluation process:

1. Both require a collaborative approach for excellence.
2. Development derives from clear definitions of audience, purpose, and context.

Audience: Demographics; experience and skill levels; knowledge and attitude toward the subject; expectations; physical limitations

Purpose: Instruction; motivation; understanding; persuasion; combination of the above

Context: Physical space and circumstances where the document/learning will be applied; language; business environment; timeframe

3. Both include information collecting, organization, and presentation.
4. Ideally, both use pilots or usability testing, and undergo revision; they should evolve with circumstance; they are never "done."
5. Both should be grounded in solid theory, data, and practice.

Design: Technical communication and instructional design consider the following:

1. Hierarchies and clusters of information, relationships among clusters, and use of distinguishing and differentiating signals (visual or verbal), such as headings, numbering, and graphics
2. Hierarchy of content importance with emphasis on key/critical areas, such as safety, core principles, rules
3. Selection to reduce extraneous information or visual noise
4. Concision to free memory for retention
5. Role of the communicator/trainer vis-a-vis the audience—guide, director, facilitator, mentor

Constraints: Both work within similar constraints:

1. Time
2. Global scope
3. Limited funding

4. Personal limitations
5. Technical limitations
6. Availability and quality of information

Careers: Good work moves professionals in both areas toward management. Technical communicators can advance to managing staff, company knowledge, and information archiving and conveyance systems; trainers can also advance to managing staff and designing/managing company-wide curricula that serves core business objectives. Both share a need to be seen and heard at the "C" level for highest effectiveness.

They share other skill sets and knowledge about

1. Interviewing skills
2. Software skills
3. Specific technical content
4. Project management
5. Finance and business analysis
6. Information design
7. Graphic design
8. Learning theory
9. Information management
10. Website design and management
11. Screen interface
12. Knowledge of technology capabilities and limitations

WHERE THEY DIVERGE

The main difference is in the medium. Training is not for introverts. A passion for the subject, however, can take even the most introverted out of self-consciousness into the "pure service" realm, but the transition can be challenging. Another area of difference is entities can throw any problem at training and expect performance improvement; they are not as fast to assume documentation as a cure.

PARTING OBSERVATION

It can be a stretch keeping one foot in both camps, but the practice of one strengthens the practice of the other.

REFERENCES

<http://www.stc.org/about-stc/the-profession-all-about-technical-communication/defining-tc>

Hackos, JoAnn T. Managing Your Documentation Projects. New York, NY: Wiley, 1994.

Kostelnick, Charles. Designing Visual Language. New York, NY: Longman, 2011.

Riordan, Daniel. *Technical Report Writing Today*. Boston: Wadsworth, 2005.

AUTHOR CONTACT INFORMATION

Mary Ryba Knepper
President
Ryba Associates, Inc.
3036 Norris Freeway
Andersonville, TN 37705
708.738.8178

AUTHOR BIOGRAPHY

Mary Ryba Knepper is president and owner of Ryba Associates, which designs and delivers custom communication training to corporations and scientific communities, and documentation and technical training in rolling mill technology. She has a PhD in English and has taught at various universities. She has been an active member of STC in the Chicago and ET chapters for over 20 years and volunteers in her community.

INDEX

A

A Manager's Guide to Acquiring Technology 121
Addicted to Meaning: Mental Models for Technical Communicators 167
Anthony, Jennifer 145
Arnett, E. Jonathan 229

B

Baby Steps: Automated Publishing Using Arbortext and SharePoint 3
Baker, Mark 197
Bartis, Christopher 221
Bending Without Breaking: Info Dev Flexibility in Agile 123
Bleiel, Nicky 105
Briody, Rita 3
Building and Maintaining Your Personal Brand 141

C

Carliner, Saul 105
Chambers, Carrie 145
Christensen, R. N. Homer 27
Collaborating in DITA: Lessons Learned from the 2012 State of Collaboration Survey 9
Collaborative Single-Source Development of Documentation and Training 27
Content Development and Delivery 1, 3, 9, 27, 37, 43, 49, 61, 67, 79, 87
Content Strategy and Design 91, 93
Creating Learning That Lasts with Interactive Documents 97
Cultivating In-House Sources of Documentation 207

D

DITA Proof-of-Concept Publishing System 37
Doc Sprints: The Ultimate in Collaborative Document Development 43
Dulin, Mary 211

E

Education and Training 95, 97, 101, 105
Elle, Edna 67
Embedding User Assistance with DITA 49
EPUB and Techcomm — Are We Ready? 61
Every Page is Page One 197

F

Fisher, Lori 173
Fletcher, Kate 135
Formalizing the Technical Communication and User Experience Relationship 173
Fox, Alyssa 123

G

Gallon, Ray 49
Gambino, Francis 145
Gamification is Here: Build a Winning Plan! 179
Giammona, Barbara 145
Gillenwater, Jamie 97

H

Hamilton, Richard 121
Herman, Liz 215
Hughes, Michael 189

I

Interviewing SMEs: Covering the Bases and Hitting Homers 145
Is Your Content Ready for Localization? 211

J

Jennings, Ann 101

K

Knepper, Mary Ryba 245
Koontz, Bernadette 239
Koup, Eric 93
Kunz, Larry 131, 141

L

Leavitt, Bill 235
Limiaco, Matso 37

M

Maddox, Sarah 43
Managing and Optimizing Unstructured-to-Structured Conversions 67
Managing Technical Knowledge: Contributing Factors to Knowledge Loss 215
McCorkle, Brett 3

O

Opsteegh, Michael 79
Our Processes Are a Mess! 131

P

Palmer, Laura A. 159
Palsson, Eileen 207
People, Projects, and Business Management 119, 121,
123, 131, 135
Plain Talk: Get the Marbles Out of Your Message 93
Planning and Creating Engaging Infographics 79
Prentice, Scott 61
Professional Development 139, 141, 145, 155, 159
Progressions 205, 207, 211, 215, 221, 229, 235, 239,
241, 245

R

Rauch, Marta 179

S

Shockproofing Your Use of Social Media 155
Siegel, Kevin 105
SME — ME — E Screencast creation and
collaboration 221
Smith, Herb J. 241
Staking Your Claim on the Social Media Frontier 159
Sullivan, Matt 105
Supporting Customers: Onsite and Online 135

T

Teaching a College-Level Editing Class 229
The Retirement of Technical Communicators —
Making Retirement an Adventure 235
The Traditional Executive Summary Versus the Video
Game Proposal One-Sheet 101
There's Nothing Simple About Simplified Technical
English 239

U

Useful, Saleable, and Buildable: The Role of UX 189
User Assistance, Technical Communication, and
Learning 105
User Experience and Usability 165, 167, 173, 179, 189
Using Mentoring Programs to Collaborate With
Industry 241
Using Semantic Technology to Create Process-Driven
Documentation 87

V

van Mansom, Kees 87

W

Web Design and Development 195, 197
Weber, Kai 167

Why Technical Writing Is Always About Training 245
Włodarczyk, Paul 9
Woelk, Ben 155

Y

Yenzer, Steven 135

Z

Zeiss, McKenzie 135