



Society for Technical Communication

2017 Summit Proceedings

May 7-10, 2017 • Washington, DC



The papers published in these proceedings were reproduced from originals furnished by the authors. The authors, not the Society for Technical Communication (STC), are solely responsible for the opinions expressed, the integrity of the information presented, and the attribution of sources. The papers presented in this publication are the works of their respective authors. Minor copyediting changes were made to ensure consistency.

STC grants permission to educators and academic libraries to distribute 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. All product and company names herein are the property of their respective owners.

© 2017 Society for Technical Communication
9401 Lee Highway, Suite 300
Fairfax, VA 22031 USA
+1.703.522.4114
www.stc.org

Design and layout by Tina M. Kister, *Nanatoo Communications*.

Table of Contents

2017 Society for Technical Communication Summit	iv
Overview	iv
Conference Committee	iv
<i>Arcelay, Rosie</i>	
Workflows: Your Treasure Map to Great Content	1
<i>Bonsignore, Alisa</i>	
Introverted Entrepreneurship	6
<i>Bowerman, Kate; Joan Carter; Melissa Kulm; and Karen Marginot</i>	
Work Global, Live Local	8
<i>Breker, Melissa</i>	
Collaboration, Communication, and Credibility: Building Stronger Content Teams.....	13
<i>Costa, Vivianne</i>	
A Layman's Crash Course on Analytics in Help Content	19
<i>Coules, David</i>	
Disrupt Your Own Self: Streamlined Publishing Through the Cloud with HTML5	26
<i>Cressey, Eric</i>	
User Experience Strategies for Winning Application Programming Interface (API)	
Documentation	31
<i>DeLuca, Todd</i>	
How May I Help You? Why Service is a Smart Way Forward	35
<i>Elle, Edna</i>	
Accessibility Made Easy: Automating and Optimizing Accessible Outputs from DITA XML38	
<i>Gardner, Michelle M.</i>	
Walking in Your Users' Shoes: Create Journey Maps for an Improved User Experience..	49
<i>Guren, Leah</i>	
Six Super Success Hacks.....	54
<i>Harvey, Michael</i>	
Documentation Support for an Internet of Things (IoT) Product: A Case Study.....	57
<i>Jones, Adam</i>	
Avoiding the \$36 Comma: Clever Editing Strategies Can Reduce Translation Costs	62
<i>Kerr, Debbie</i>	
Learning Styles and the Cancer Experience	67

<i>Kleinsmith, Mark</i>	
Video Provides the Edge.	75
<i>Kramer, Meredith</i>	
To Manage or Not to Manage: That is the Career Path Question	79
<i>Kreger, Jessica</i>	
Proving Return on Investment (ROI): Agilely Cutting Costs in Help	82
<i>Maddox, Sara</i>	
A Tech Writer, a Map, and an App.	84
<i>McCue, Shannon A.</i>	
Case Study: Finding Your Documentation Utopia in an Agile Environment.	88
<i>Mincey, Chrystal R.</i>	
Knowing Your Client’s Language	92
<i>Navarro, Elizabeth</i>	
Everyone’s an Editor	94
<i>Parkinson, Mike</i>	
Perfect: How to Turn Words and Data Into Powerful Graphics	99
<i>Perlin, Neil</i>	
We’re Going Mobile! Great! But What Does That Mean?.	103
<i>Peters, Ear</i>	
Section 508: Are You Ready for Americans with Disabilities Act (ADA) Compliance Standards?.	106
<i>Pfeilsticker, Steve</i>	
Supporting Customer Advocacy with Intelligent Knowledge Management	112
<i>Prabhakar, Rahul</i>	
How Social Media Can Be Part of Your Set of Technical Communication Skills Tools.	118
<i>Prentice, Scott</i>	
Regular Expressions for Tech Writers.	123
<i>Proff, Allie</i>	
My Android Dreams of Electric Cats: Are You Capturing Your User’s Emotive Analytics?129	
<i>Ruggeri, Kathleen and Lisa Adair</i>	
Improve Content Quality with One Process Change	136
<i>Saxena, Shikha</i>	
The Art of Writing in Agile.	141

<i>Keith Schengili-Roberts</i>	
Is Darwin Information Typing Architecture (DITA) Right for You? Scenarios for Considering a Move to DITA.	150
<i>Semp, Monique</i>	
Leveraging Structured Authoring/DITA Techniques When All You Have Are Unstructured Tools	156
<i>Simon, Kelley</i>	
Creating a Free Intranet Using Drupal	166
<i>Stevenson, Sean</i>	
Getting to Compliant: Responding to Government Requests for Proposal (RFPs) for the Technical Communicator	172
<i>Strimling, Yoel</i>	
So You Think You Know What Your Readers Want?	177
<i>Tincher, Louise</i>	
The Wonderful World of Proposals.	186
<i>Todd, Becky</i>	
Engineering Content Champions.	189
<i>Vega, Laurian C.</i>	
Novices and Expert Users, Not Novices or Expert Users.	193
<i>Wilson, Jane</i>	
Creating User Documentation in an Agile World	198
<i>Woelk, Ben</i>	
Follow the Yellow Brick Road: A Leadership Journey to the Emerald City.	202

2017 Society for Technical Communication Summit

Overview

The 2017 Society for Technical Communication Summit was the STC's 64th annual conference. The theme of the 2017 conference was *Gain the Edge to Get Results*. The Summit was held from May 7-10 at the Gaylord National Resort in National Harbor, Maryland – just minutes from downtown Washington, DC.

STC's annual Summit is the premier conference for technical communication education and networking, and attracts over 600 attendees and 40 exhibitors. Attendees are professional technical communicators at all levels, from beginners to seasoned veterans, and includes technical writers, editors, illustrators, managers, information designers, architects, content strategists, instructional designers, usability and user-centered design practitioners, researchers, professors, and students.

Learn more at <https://summit.stc.org/conference/>.

Conference Committee

The Conference Committee was responsible for managing the call for proposals, reviewing all proposals, prioritizing the submissions in each topic

area, coordinating speakers, and ensuring a well-balanced and comprehensive program.

Members of the 2017 committee:

- Molly Jin, Director of Education and Meetings, Society for Technical Communication
- Todd DeLuca, Conference Committee Chair, Black Knight Financial Services
Headshot_DeLuca
- David Caruso, Program Manager, National Institute for Occupational Safety and Health, CDC
- Track managers:
 - Jamie Gillenwater, Career & Leadership
 - Michael Opsteegh, Design & Testing
 - Li-At Rathbun, Writing & Communication
 - Roger Renteria, Technology & Development
- Reviewers:
 - William Butcher
 - Charles Crawley
 - Rachel Houghton
 - Annette Riley
 - Jennifer Staley
 - Lloyd Thompson-Taylor





Arcelay, Rosie

Workflows: Your Treasure Map to Great Content

As technical writers, creating content for software products can be an adventure. With workflows, you and your team can navigate the jungle of bad communication to find the treasure of user-focused content. This knowledge will empower you and your team to minimize the mini-waterfall effect you sometimes see during Agile processes. One of the challenges for technical writers who work in Agile teams is to complete documentation tasks by the end of a sprint. Sprints can become mini-waterfalls, where the documentation work begins after the developers have finished their code work and when quality assurance (QA) begins testing. While the mini-waterfall effect can be difficult to overcome, you can minimize it by using workflows.

What is a Workflow?

A workflow is a treasure map that you can follow to mine content. Workflows provides a bird's-eye-view of the process your team is creating or improving. They also help you to determine the type of content that your users will need.

Workflows are tools that help you to trace the path that users take through a software product or training process. This makes workflows similar to customer journey maps. A customer journey map "is a diagram that illustrates the stages that a customer goes through while engaging with a business to complete a process." (Laplante, 2017). They are useful for user experience (UX) and content marketing research. The difference between customer journey maps and workflows is that workflows expand the concept of customer journey maps to include backend processes.

Workflow Benefits

These are the benefits of having a workflow:

- Workflows provide a starting point for dialog: you can go to your developers and say "This is how I understand the software's workflow goes. Can you tell me if this is accurate?" Developers or QA

will help you shape and pinpoint how the process works.

- Workflows reveal the kind of content you'll need to create and where in the process it should go. For example, during a sales process, you know that confirmation emails are necessary to inform the customer that their order and payment were successful. Is there a confirmation email going out? Do we need an email for when the process fails? These are the kinds of questions that you can ask from looking at workflows.
- Workflows are valuable as an internal reference tool. In my company, technical support and customer service use my workflows to supplement the training content they receive from the teams that create the main documentation.

Plan your Expedition

The first question to ask yourself is, "Does this need a workflow?"

While workflows come in handy at any stage in the design process, it's best to use them at the beginning. For example, if you are working with an existing process that is getting additional steps, you can use workflows to document that process. There may

be situations where you won't need a workflow. For example, minor bugs or updates that do not change the software process do not require a workflow. Major software process changes, such as new user interfaces (UIs), UX workflow changes, or additional API calls will require a workflow. This helps to establish design expectations.

If you have determined that you need a workflow, follow these steps:

1. Map an existing or a new design.
I recommend using flowchart notation. Flowcharts use different shapes to represent actions in a workflow. For the sample workflows in this paper, I use the Start/Stop (oval) and Process (rectangle) shapes, along with direction arrows.
2. Consult the experts to finalize the workflow order.
Look to your team's planners and designers for assistance. In my team, developers and QA analysts help me shape and pinpoint how the

process works. In other types of teams, your guide could be the project manager.

3. Study your workflow process map to discover what kind of content is needed.
For example, an email confirmation step would indicate that you will have to compose email text.
4. Create the content and publish it.
Use the workflow as reference material.
A user visits the store website.

Sample Workflow: Purchasing a Software License Through Your Online Store

1. Map your Quest: Map an Existing Design

Let's use a simple workflow for purchasing a software license through an online store and see how we can mine it for content.

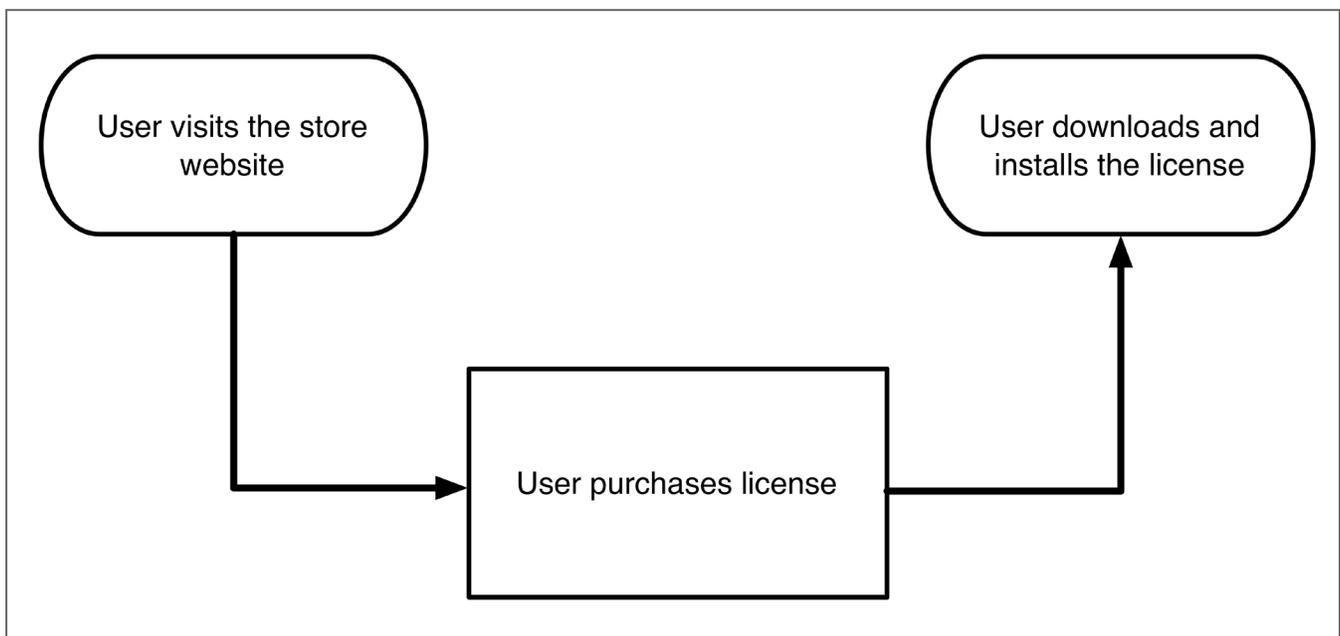


Figure 1. A simple workflow for purchasing a software license through an online store.

The simple purchase steps are:

1. A user visits the store website.
2. The user purchases the software license.
3. The user downloads and installs the license and begins using the product.

If we analyze each step further, we can ask a few questions about the processes. For example:

1. A user visits the store website.
 - Does the user have to log in or create an account?
 - Can the user purchase the license from the landing page, or does the user need to navigate to a specific page to purchase the license?
2. The user purchases the software license.
 - How does the user pay for the license?
 - Does the store need the user's IP address or other information to complete the purchase?
 - Does the user receive a purchase confirmation email?
3. The user downloads and installs the license and begins using the product.
 - Does the system download the license automatically, or does it require manual installation?
 - Does the purchase confirmation email include instructions, expectations, or how to contact customer service? What happens when something goes wrong?

2. Consult the Experts: Finalize the Workflow's Order

No expedition is complete without guides to assist you in your journey. The software developers are the visionaries. The map is already in their heads: all you have to do is document it. However, QA is probably your best resource, as they have to test the finished product from start to finish. They will know whether the paths that the developers created are successful or filled with perils. Good Product Owners or Project Managers will listen to your proposals and consider your awesome workflow ideas, especially when you sell them on how they benefit stakeholders. Finally,

stakeholders also make good content guides. After all, they are your primary content consumers.

There are several ways to obtain the information that you need to write a workflow:

- Participate in the software design phase, either as a contributor or a notetaker. This works particularly well if you're on the same Agile team. Gather your information during developer hack sessions and determine the best course of action to solve the particular problem you are trying to tackle.
- Talk to your developers. Sometimes, asking your developers to explain their vision is the best way to gather this information. You can ask them direct questions. Let them know there may be more on the way as you begin the design stages of your workflow.

3. Start your Treasure Hunt: Study Your Process Map to Discover the Content Needs

Let's say that after you have spoken to the software developers and your QA analysts, your most recent workflow looks like Figure 2.

The workflow shown in Figure 2 has more steps in the process and includes the names of the API functions (the backend processes) that enable those steps.

From this workflow, I observed the following content needs:

- If the UI is new, create strings (UI text). Strings in an existing UI may need a review or an up-date.
- Create or review API call documentation for at least six calls.
- Create or review the email confirmation text.
- Create a how-to document or video that guides the user through how to purchase, download, or install the software license

Study this workflow and look for any other content needs. These needs can vary, depending on your technical writing skills and experience. The important skills you are learning here are to anticipate your content needs and turn them into tasks for your sprint or backlog.

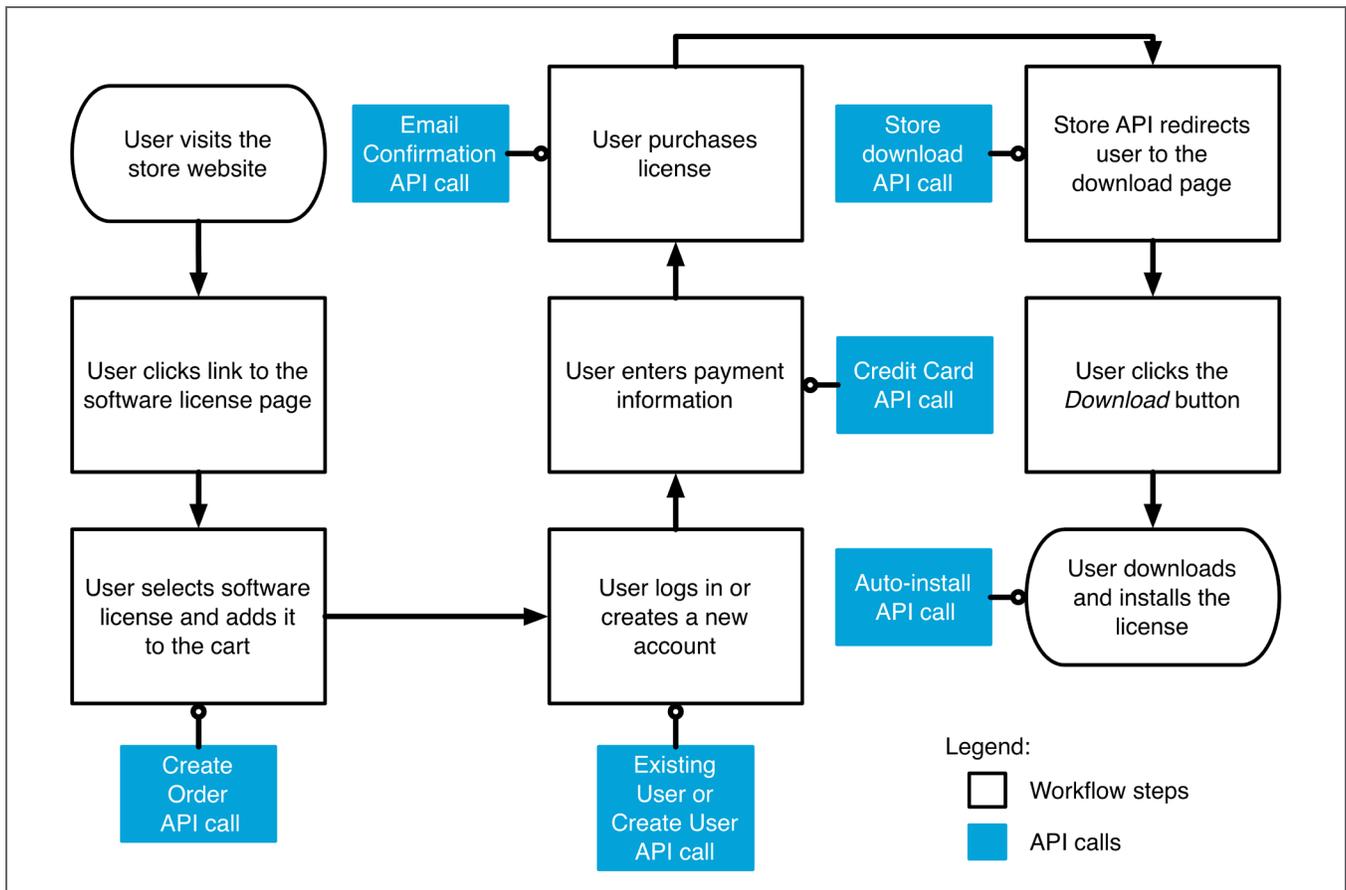


Figure 2. The simple workflow is now more complex, thanks to input from developers and QA.

4. Create the Content and Publish It

Now that you have your to-do list items ready, add them to your backlog or sprint. Then you can create the content and follow your normal publishing procedures. At the end of the sprint, share your workflow with stakeholders. Visual learners will enjoy seeing the process in one diagram.

Let the Adventure Begin!

With your newly-obtained workflow knowledge, you are ready to begin! Create your map and consult the experts. Then start your treasure hunt, discover your content needs, and create and publish your content. Those content spoils are one workflow away from discovery!

Cartography: Workflow Tools

Because workflows are flexible, you can create them in any format. I personally have used handwritten workflows on paper or whiteboards. Once I finalize a workflow, my tool of choice is Omnigraffle. I've also used Google Drawings to sketch workflows. A Google search for "diagram software" or "flowchart software" yields results for additional software options.

Resources

- Groki, Megan. "How to Create a Customer Journey Map." UX Mastery (16 September 2014). <http://uxmastery.com/how-to-create-a-customer-journey-map/>
- Laplante, Mary. "Product Content: The Hidden Gem in the Content Marketer's Tool Kit." BrightTALK (1 March 2017). <https://www.brighttalk.com/webcast/9273/245323>

Author Contact Information

Rosie Arcelay
Technical Writer II
cPanel, Inc.
2550 N Loop West
Suite 4006
+1.713.529.0800

Author Biography

Rosie Arcelay works for cPanel, Inc. as a technical writer on an internal development Scrum team. She writes API content for product developers and training content for customer service and sales. Besides writing, Rosie likes to work on her family's (and other families') genealogy, edit a theater blog, play video games, and binge-watch shows online. She has been an STC Member since 2011.



Bonsignore, Alisa

Introverted Entrepreneurship

You've heard the joke: How do you identify an extroverted engineer? They stare at your shoes instead of their own. But this is part of the huge misconception that introverts can't — or won't — interact with others. The truth is that introverts are perfectly capable of conversation, they're just drained by interaction. We don't normally associate introverts with successful entrepreneurship. But introverts can be successful in business; you just need to make the most of the traits you have.

Finding Balance

Are you an introvert? You're not alone, particularly within the community of technical communicators. For many of us, "introvert" and "entrepreneur" conjure images of vastly different people. The former is seen to be quiet, the isolated worker, while the latter is imagined to be an outgoing dynamo.

But introverts can absolutely become successful entrepreneurs.

Introversion doesn't mean a life of isolation any more than entrepreneurship means that you're always "on" and in sales mode. Life is about balance.

Not Something That Needs to Be Cured

Most introverts have spent their entire lives being encouraged to come out of their shell. They're rarely the popular people. But they have deep insights, rich inner lives, and a deep understanding of others.

As Susan Cain says in her book *Quiet*, "Don't think of introversion as something that needs to be cured."

Advantages of Introversion

Introversion has surprising advantages for communicators.

- Introverts are listeners. When you're not doing all the talking, you have more opportunity for listening. That means that we tend to have a deeper understanding of all parties' needs and wants.
- Introverts tend to be empathetic. This is an excellent skill for a writer, content strategist, or any-one who focuses on the user experience. We understand the reader or user's pain, and address it.
- Introverts are concise. We don't want to talk any more than we must. Our meetings are short and efficient. Our writing is lean and pointed.
- Introverts do their research. We love to research and prepare. We will come into professional meetings with knowledge and insight.

Make the Most of What You Have

When playing to your strengths, and finding workarounds for your weaknesses, it's possible to find success as an introverted entrepreneur.

Resources

Cain, Susan. Quiet: The Power of Introverts in a World That Can't Stop Talking (New York, NY: Crowne Publishers), 2012.

Author Contact Information

Alisa Bonsignore
Writer, Strategist & Speaker
Clarifying Complex Ideas
Pleasanton, CA
+1-408-256-0621

Author Biography

Based in the Bay Area, Alisa spends her days clarifying complex ideas, translating technical and clinical information into understandable language tailored to the needs of healthcare, network security and healthcare IT clients around the globe. She uses humor and real-world scenarios to form the basis for her talks about professional development. Alisa was elected to serve as Director of STC (2016-2018), and is past Chair of the Education Advisory Panel for STC.



Bowerman, Kate; Joan Carter; Melissa Kulm; and Karen Marginot

Work Global, Live Local

As a remote worker, how do you become an integral part of your team, enjoy your team's culture, grow your career, and find balance? Four technical communicators with over 30 years of success as remote communicators share tips about building and maintaining effective virtual working relationships. This includes staying visible and engaged with your stakeholders and teams, participating in and fostering a healthy team culture, and staying focused, particularly when working from a home office.

Note: Within the context of this paper, “remote” refers to any location where you’re not co-located with your team. For example, a technical communicator working remotely from home or from a satellite office.

Stay Visible and Engaged

Remote technical communicators often struggle with the “out of sight, out of mind” trap. Working in a non-office location can make it far too easy to stay heads down and focused on work. The problem? Your team forgets you’re there and has no clue about what you’re working on.

Communication Essentials

Keeping the lines of communication open is critical to staying visible. It’s important when you’re co-located with your team, but it’s even more important when you’re remote.

Informal methods of communication, like instant messaging, can be a great way to get to know your coworkers, work on team projects, and show that you’re easily available to answer questions and resolve issues.

Formal methods of communication can help showcase your work and your contributions. Salesforce Chatter is a news feed tool that’s great for asking questions,

having discussions, and making announcements. Wikis and email are also useful.

Publicize your availability. Consider holding “office hours” —a dedicated hour when “your door is open” and team members know that they can instant-message or call to talk.

Being Your Best Virtual Self

When you dial into remote meetings, it’s essential that you have a good headset and webcam. It’s critical for ensuring that meeting participants can easily hear and see you. The quality of your equipment also reflects on your professionalism.

Equipment

Consider investing in a set of headphones with a microphone, instead of relying on your computer’s microphone. The computer microphone tends to pick up noise from the computer fan and from your typing. Pre-check your webcam video quality. Some computers have pretty good webcams, and you can safely use the one built into your computer. But if the image is low fidelity, consider buying an external webcam.

Consider running a test meeting to check your Internet speed for performance in online meetings. For example, Google Hangouts can require 25-50 mbps for best audio and video quality.

If you work from home, have an Internet backup location like a public library or coffee shop.

Distraction-Free Background

When you work remotely, it's important to give a professional appearance. Be aware of your surroundings and how they come across to the people that you're meeting with. Make sure you're someplace quiet with reliable phone and internet connections. What's on the walls? Is the room tidy? You don't want your coworkers to see your dirty dishes when you work from the kitchen table.

You also want to make sure that you're not causing audio or video distractions. Hold meetings in a quiet place, not on the Metro or in a crowded coffee shop. If you have pets, keep them out of the room and quiet if possible. A little bit of your personal life is humanizing, like a cameo from your kitty or pup. But remember, a little goes a long way.

Running good meetings is useful skill whether you're local or remote. But when you're remote, meetings can take on more importance. You have fewer interactions with your coworkers, so the impression that they have of you is based on the encounters that you do have.

Educating Your Team

If you're attending meetings, work with the organizer or host to make sure that remote attendees can fully participate. Ask the organizer to check that the conference room and technology—like the microphones, webcams, video feed, and speakers—are working properly.

Train your team to be considerate of remote participants. For example, educate them to eliminate side conversations and to position themselves near a microphone and video camera. Don't be afraid to ask who's talking or ask the speaker to move closer to the mic. Let people know how to help you.

If your company uses video, then turn on your webcam so that your team can see you. It makes a big difference and literally helps you stay visible.

Time Zone Awareness

When you conduct meetings, make sure that the meeting time works for attendees, especially when people are meeting across time zones. It's frustrating

to always have to stay late or meet at an inconvenient time, especially when the burden falls on the same person or group. Sometimes it's not possible to find a good time for everyone, but you can try trading off inconvenient times. Share recordings and notes from the meeting so people who couldn't attend are kept in the loop.

If you're the meeting organizer, make a point of thanking the people who are attending during what are odd hours for them. That goes a long way towards making people feel visible and appreciated.

Participate In and Foster Team Culture

One of the best parts of working with a team is the opportunity to participate and help foster a healthy and fun team culture.

Be Social

As a remote team member, you may feel that your location limits the chance to experience this. But it doesn't have to be so! There are ways to enjoy a healthy work culture both with your team and locally in your home town.

Host Virtual Get-Togethers

Invite colleagues across your organization to monthly virtual happy hours or morning coffees. All you need is a virtual space (think Google Hangouts, for example) and a way to get the word out. Some teams even use themes, set ground rules (such as no work talk at the happy hour), and invite family and friends. You can set up open invitations for groups as well as virtual, informal one-on-one coffees with colleagues to establish and build good relationships.

Be a Joiner

Look for opportunities to join interest groups outside of your organization, such as through STC or local meetups. Your organization may have employee resource groups that can provide opportunities to network within your company. They may even offer a chance to contribute or lead special initiatives, which can further increase your personal brand.

Lunch Locally

Speaking of STC, check out the monthly lunch series that many local chapters offer. Consider yourself a local ambassador for your company, and offer to host local lunches when colleagues or clients travel to your area. One host bought the flags of the countries her visitors came from. She invited her coworkers to dinner and displayed the flags on the table. A truly thoughtful welcome!

Volunteer Together

Some companies offer volunteer time to their employees, but even if yours doesn't, volunteering is a great way to get to know others outside of your regular sphere and give back to your community. Look for volunteer opportunities within and outside of your community. Share stories and pictures with your team and company, and become an inspiration to others.

Build Good Working Relationships

Even though you may be miles or even time zones away from your team, you can build good working relationships with them. Ask about preferred meeting times. Discuss opportunities to visit your company's home or regional offices, when possible.

Plan Visits

Plan your visits well in advance of your arrival. Arrange travel around your team's in-office presence. Publicize your visit! Add face-to-face meetings to your calendar and think strategically about who to meet with. Sometimes it's helpful to meet with folks that you work with sporadically, like UX designers or video production engineers. Schedule sit-downs with PM and managers. Work with your manager to set up a team brunch or lunch or happy hour. It's a good way to get to know your coworkers.

Wine and dine for success! Plan to connect with coworkers over lunch, dinner, coffee, walks, or plan a team volunteer event.

While You're Visiting

Tell folks where to find you. Remind people that you're in town, even if you'll be out of sight.

Don't expect to get major project work done. Focus instead on in-person project tasks, such as:

- Side-by-side doc reviews
- UX design reviews
- Brainstorming sessions
- After You Get Home

Keep up the momentum! Follow up on the meetings that you had onsite. Solidify relationships.

Stay Focused

Why work remotely? For all of us there were lots of good, convincing points that supported working from home:

- A widely distributed team.
- A horrendous commute.
- The ruinous cost of gas.

There are a lot of immediate advantages to working remotely:

- You can relocate to a cheaper real estate market.
- You can be more involved in child or elder care.
- You can sleep much later.

To make remote working work for you, consider developing ground rules to help you give your employer its due.

For example:

- Create your own, dedicated space for work.
- Groom for a public appearance.
- Move around throughout the day.
- Nurture your social life.
- When your workday's over, turn it off.

Set Aside Office Space

If at all possible, create an office in your home, ideally, one with a door. Educate everyone you live with about your space. Give them the opportunity to respect your boundaries. Minimize distractions. Leave the kids, cats, dogs, and roommates on the other side of the door. See Figures 1 and 2.

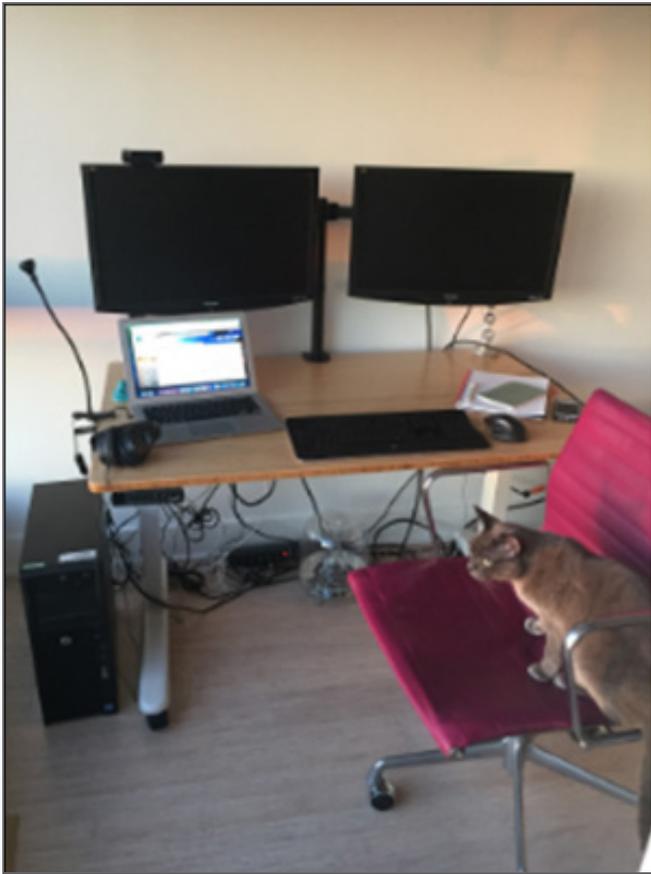


Figure 1. Home office example

Groom for Success

Get ready for your day as if you were going to the office—or at least going out in public. That is, shower, dress, and groom. Dressing doesn't mean suit and tie; just abandon your pajamas in favor of something with a little more decorum. Grooming sets you up to deal professionally with your colleagues. Not only that,

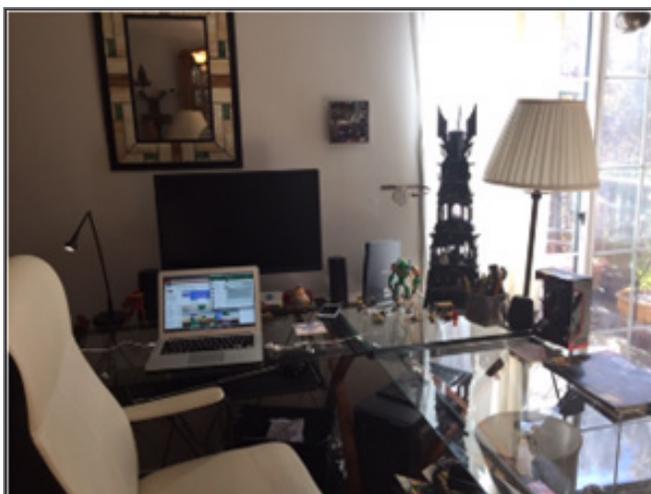


Figure 2. Another home office example

when you look good, you feel good. And when you feel good, it shows in everything you do.

Move It! Move It! Move It!

To keep your spirits up and your body limber, get out of your chair and move around every hour or so. Imagine you're on-site at your company. You'd likely be getting up regularly to talk with colleagues, grab something to drink, or step out to lunch. As a remote worker, those built-in exercise opportunities vanish, so it's up to you. Consider building breaks into your mid-morning and afternoon routine. Use them to walk around for 15 minutes. If you're not otherwise running active errands at lunch, make a point of moving around for 20 minutes or so.

Here are a few tips for working movement into your day:

- Google “desk exercises” for ideas about how to move while sitting.
- Download a desktop tool for scheduling movement breaks throughout the day.
- Use a kitchen timer to time your breaks and keep yourself honest.

Regular movement boosts your productivity, focus, and creativity, and, most of all, your sanity.

Nurture Your Social Life

Make a point of friendship and socializing. If this doesn't come easy, take classes, volunteer, find ways to be around and interact with other people.



Figure 3. Caption: Move it! Move it!

For most of us, a typical work day uses up a minimum of 8 of our 12 to 16 waking hours a day—at least half. It's easy to become isolated, especially in the colder, darker months.

Be in the world. This will also help to preserve your sanity.

Turn It Off

When your work day is over, make sure it's over. Give yourself a post-work evening. For that matter, give yourself a morning and a weekend. Just as you minimize distraction during your working day, invite it in when the day is done. If you have an enclosed office turn off the lights, shut the doors, and leave it.

To Sum It Up

These are guidelines that work for us. You may be someone who concentrates better with noise. If you are, take your work to a sunny cafe. That'll also knock out nurturing your social life. Adjust according to what works best for you. But, really, no pajamas!

Resources

Axtell, Paul. "What Everyone Should Know About Running Virtual Meetings." *Harvard Business Review* (April 14, 2016). <https://hbr.org/2016/04/what-everyone-should-know-about-running-virtual-meetings>

Cullinan, Renee. "Run Meetings That Are Fair to Introverts, Women, and Remote Workers." *Harvard Business Review* (April 29, 2016). <https://hbr.org/2016/04/run-meetings-that-are-fair-to-introverts-women-and-remote-workers>

Erard, Michael. "Remote? That's No Way to Describe This Work." *New York Times* (June 18, 2016). <http://nyti.ms/1UfvRGq>

Larson, Quincy. "The Economics of Working Remotely." *Freecodecamp* (August 24, 2016). <https://medium.freecodecamp.com/the-economics-of-working-remotely-28d4173e16e2#.65nd1d21v>

Salesforce. "Trailhead: Virtual Collaboration at Salesforce." https://trailhead.salesforce.com/module/manage_the_sfdc_way_virtual_collaboration

Author Contact Information

Kate Bowerman
Staff Technical Writer
Salesforce
The Landmark @ One Market
Suite 300
San Francisco, CA 94105
+1.415.715.7701

Joan Carter
Staff Technical Writer
Salesforce
The Landmark @ One Market
Suite 300
San Francisco, CA 94105
+1.559.369.4322

Melissa Kulm
Lead Technical Writer
Salesforce
The Landmark @ One Market
Suite 300
San Francisco, CA 94105
+1.571.419.4168

Karen Marginot
Staff Technical Writer
Salesforce
The Landmark @ One Market
Suite 300
San Francisco, CA 94105
+1.703.463.3244



Breker, Melissa

Collaboration, Communication, and Credibility: Building Stronger Content Teams

Technical communicators are regularly seen as master translators, converging ideas to communicate value to customers. Yet, often in teams, we fail to translate the full value of the work we do. We face last minute requests, insufficient information and resources, and incorrect assumptions about requirements. Although as team members, we depend on each other to get work done, we miss opportunities to be effective and deliver results. From quality and time delays, to firefighting and divided teams, there are significant impacts on results, relationships, and performance. In this paper, I'll share insights from my experience of working with different teams across government, technology and non-profits, and show you how you can improve the strength of your content team and maximize their results.

Getting Started

"A team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they are mutually accountable" (Kazenbatch & Smith, 1994). [1]

We've all heard the statement "the whole is worth more than the sum of the parts"

Teams allow us to accomplish things that we can't individually. From specific tasks to projects, research indicates that people who work together are more likely to be happy with their jobs, while companies who support teams enjoy increased profitability.

When it comes to content teams, we're often so busy doing the work, that we neglect the importance of governance and well designed processes that ensure the success of people working together.

Or, from what I've seen in content teams:

- Lack of a clear plan.
- Poor communication.

- Poor collaboration.

The next sections will uncover why content teams struggle, and some ideas you can use to improve your team planning, communication, and collaboration.

Why Do Content Teams Struggle?

Content teams can struggle for a variety of reasons. Teams need the right mix of skills to address different project requirements. Often, it comes down to the following frustrations:

How Do I Build a Strong Content Team?

Stage 1: Create a Plan (or Know Where You are Today)

Improving team performance is not always easy. Before you make any changes, you'll need to know how your team is performing right now and how ready it is to shift perspective.

Content team frustrations	See the following stage
Lack of clear planning. Unclear vision, goals, and objectives for success. Missing skills and uncertain roles. Ineffective leadership and management. Ad hoc processes and approaches to project needs.	Stage 1: Plan
Poor communication and interpersonal conflict between team members. Unclear benchmarks for metrics. Mismatched work styles around problem solving and decision-making.	Stage 2: Communicate
Ineffective leadership and management. Uncertain relationships. Missing skills on the team.	Stage 3: Collaborate

Table 1. Content team frustrations

How does your team rate? Do you see gaps in how your current content team performs? Where are the greatest opportunities for improvements moving forward?

A clear plan can address most of the common frustrations you may face as a content team. Take time to go through each of the statements on the grid, and rate your team performance.

Determine Vision and Goals

Teams allow us to innovate faster, see mistakes more quickly and find better solutions to problems. [4]

Goals help leadership drive success metrics, create stronger communication between team members, and set the ground rules for collaboration.

To build a strong team, you'll need to first understand your teams' vision and goal as it relates to larger organizational goals. By clarifying these goals, you can determine the fundamental "why" of the group in the first place. Then, set about to understand the immediate high priority goal that will determine success.

Understand Stakeholders' "What's in It for Me (WIIFM)?"

When was the last time you sat down to think about other people on your team? By understanding the context in which they operate, you can clarify what's important to stakeholders based on their mental model of who they are, what they care about, and what they do.

Being empathetic to others will help improve communication and uncover team skills required to meet project needs.

Listen, not only to what they communicate, but also to the things they don't. Understand their preferences when it comes to work style and how they consume and share information. Think about their needs and who they influence, and who they are influenced by.

Clarify Roles

For many of us, juggling multiple projects and roles is the norm. It means that we jump from activity to activity, which has an impact on productivity and happiness at work. Be clear about your role requirements and set boundaries

Common Frustrations	Team Requirements	Content Team Considerations
Planning	Planning and preparation	Mission, goals, and purpose are clear and shared via a clearly laid out and written plan. The primary goals of the team are goals endorsed by members. There is transparency with personal and professional agendas.
Leadership and Management	Leadership and Support	Management is active and visibly supportive. There is trust within the team. The team has timely support and resources. Each team members have a voice in discussions. There is a mutual feeling of trust.
Metrics	Vision, goals, objectives	Goals or expectations are realistic. There are clearly defined goals and objectives. All members know and can share metrics for success. There are clear and agreed upon timelines for results. Members understand what the team must produce.
Communication	Feedback and sharing	There's clear communication and feedback shared between team members. There are clear consequences for members who do not deliver work to the agreed upon timelines and standards. There are regular and efficient team meetings that make good use of available time.
Work style	Interactions	Team members are considered productive. Team member feel respect for each other. Existing policies and procedures are clear for the team (including conflict resolution). Team members are supportive and understanding.
Ad hoc processes and approaches	Process	There is a clearly defined process to complete work. There is transparency regarding triggers and outcomes to move content through the process. The lifecycle of content is considered (including management and maintenance). There's a desire for continuous improvement, including tasks that are less than enjoyable.
Team skills	Roles	Members' roles and responsibilities have proper definition (including leadership and accountability). Commitment to people-building is emphasized. Strengths and weaknesses of team members are known. Appropriate team training is provided at all levels. There is a clearly defined reward system.

Table 2. Requirements for highly performing teams. Adapted from Walter J. Michalski [2], Lydia Kavanagh, and David Neil [3].

Goals	Requirements	Pressures	Organizational objectives
Blind spots	Risks	Conflicts	Personal objectives
Problems	Passions	Work styles	Communication styles

Table 3. Stakeholders' WIIFM.

Clear roles support better content processes. By understanding the roles involved, you can map your activities to clarify project requirements and dependencies. Clear roles also helps you to determine skill gaps and requirements for project success.

Stage 2: Communicate

Set Shared Expectations

All teams manage a certain level of chaos and uncertainty. Through clear expectations, you can reduce conflicts and assumptions, set standards, and improve alignment.

Keep people informed and be transparent about how decisions get made, when, and how. Let them know about your needs when it comes to timing, material, and other support. Identify risks in advance to reduce firefighting and last minute requests.

Be clear about what's acceptable and unacceptable. Sometimes it's hard to share unpleasant things, but from a team perspective, it's important. Model the behaviors you'd like to see and set up clear rules of engagement with others. This may take practice and patience because team members can set up rules to defend their own reality and may be set in their ways.

Set Benchmarks

Prepare Key Performance Indicators (KPIs) to measure success. Uncover, discuss and validate what great performance means for you as a team.

Understand the difference between the overarching goal of impact versus project output. Delivery is not necessarily success; it's about measuring the things that matter, that have impact.

Set a Shared Language through a Communication Plan

Communication plans are a great way to share what, how, and when information is shared and who is responsible for communicating information. Do a web search for "project communication plan template" to see some samples for your projects.

Establish a shared understanding regarding terminology and how it's applied in a group setting. Shared knowledge allows more effective and efficient communication.

Facilitate conversations so the team works more effectively together. Be a resource for others to help create synergy within the group.

Know Yourself

The way you work within a team is influenced by your personality. It's important to be clear about your preferences, qualities, and attributes, as well as those of your team members. This alignment of work styles will improve communication and help the leadership and management of your project.

Competent role performance can be connected to the relevant member's "people style," according to Robert and Dorothy Bolton. When assigned team roles aren't the right fit, there can be a significant impact on team capability and trust. [5]

Assess your own strengths and weaknesses. Know what matters most to you and to the group, then look at how you're spending your time. Track your time to see how your activities align with both sets of priorities.

Although there's no one effective tool for personality assessment, tools like the Myers-Briggs Type Indicator (MBTI), the Five-Factor Model (FFM) and the DiSC® profile assessment can help you understand your

natural preferences and how you relate to other people.

Stage 3: Collaborate

Build Relationships

In their book *Execution*, Larry Bossidy and Ram Charan share insights around the impact of high-integrity leadership on building relationships and trust.[6] Clear and careful communication is at the heart of how leaders build high-trust organizations.

A great relationship is built on how you relate to the other person involved. Start with active listening to reflect what you're hearing in conversations. Ask questions so you can see the other persons' position, facts, and intentions clearly. Then, apply the right balance of urgency, intensity, and focus to build trust and improve your relationships.

To create a stronger content team, you'll need to build social capital and credibility. Listen and choose the right approach to relationships to create the right environment within team.

Stop Selling

Focus on creating genuine connections and use a clearly defined process to track and measure project requirements.

Stop focusing on what you want them to do. As Mark Goulston and John Ullmen indicate in their book, *Real Influence*, "You need to see their vision and make it a part of yours. You need to make them want to work with you to achieve amazing outcomes. It's the secret for building long term commitments and for reaching big goals." [7]

Co-Own Ideas

Shift problem solving to asking questions. Come to meetings with curiosity and questions that could support the team in creating a new perspective. Ask "What if?" and open ended questions to ideate around possibilities. A co-creation approach shows respect and trust and considers possibilities outside the confines of problem solving. It also allows other people to take a leadership role within the group, while supporting team skills that may be missing.

Conclusion

As Zig Ziglar once said, "Research indicates that workers have three prime needs: interesting work, recognition for doing a good job, and being let in on things that are going on in the company.

There are many ways to translate the value we do as technical communicators. By strategically preparing for change, we can understand our internal audience, create a common vision, and use the right approach to help align stakeholders.

References

- [1] Katzenbach, Jon R. and Smith, Douglas E. *The Wisdom of Teams* (Boston, MA: Harvard Business School Press), 1993.
- [2] Michalski, Walter J. *40 Tools for Cross-Functional Teams: Building Synergy for Breakthrough Creativity*, (New York, NY: Productivity Press), 1998.
- [3] Kavanagh, Lydia and Neil, David. "Working in Teams: A Practical Guide" edX course (December 16, 2016) Web link.
- [4] Duhigg, C. "What Google Learned from its Quest to Build the Perfect Team." *The New York Times Magazine* (February 25, 2016) Web link.
- [5] Bolton, Robert and Bolton, Dorothy. *People Styles at Work: Making Bad Relationships Good and Good Relationships Better*, (New York, NY: AMACOM; 2 edition), 2009.
- [6] Bossidy, Larry and Charan, Ram. *Execution: The Discipline of Getting Things Done*, (New York, NY: Crown Business), 2002.
- [7] Goulston, Mark M.D and Ullmen, John Dr. *Real Influence: Persuade Without Pushing and Gain Without Giving In*, (New York, NY: AMACOM), 2013.

Other Books

- Kohn, Stephen E., and Vincent D. O'Connell. *9 Powerful Practices of Really Great Teams*, (Wayne, NJ: Career Press), 2013.
- Edmondson, Amy C. *Teaming: How Organizations Learn, Innovate, and Compete in the Knowledge Economy*, (San Francisco, CA: Jossey-Bass), 2012.
- Willett, Alan. *Leading the Unleadable: How to Manage Mavericks, Cynics, Divas, and Other Difficult People*. (New York, NY: AMACOM), 2017.

Author Contact Information

Melissa Breker
Facilitator, Strategist, Speaker
Breker Strategy Consulting
Vancouver, BC, Canada
melissa.breker@gmail.com
+1 604 619 7743

Author Biography

For over ten years, Melissa has been working with, and helping teams create content with impact.

She works with creative agencies, customer experience teams, product developers, communication teams, and marketing strategists to inspire and evolve teams to generate results.

From technology to government and non-profit organizations, Melissa takes a collaborative, systematic, thoughtful and analytical approach to the work she completes. As a content coach, speaker, and workshop facilitator, she's developed courses for the University of British Columbia, Content Marketing Institute, and MarketingProfs.

When she's not out running workshops, she's spending time with her lovely family in beautiful Vancouver, British Columbia.



Costa, Vivianne

A Layman's Crash Course on Analytics in Help Content

If your help content lives on an HTML page, then you can easily measure how well it's performing, who it's reaching, how they get to it, and what you need to do to make it more, well ... helpful. Luckily, you can do this for free with Google Analytics. This paper provides a step-by-step breakdown of what you need to do to develop a Google Analytics monitoring strategy that defines the important metrics you need to be measuring and how you can use it to improve your content strategy. It also covers the data that is relevant to report to your leadership team and stakeholders to promote the importance of your work and provide insights to product teams.

One of the main deliverables of an analytics monitoring strategy is a clear understanding of the specific data your team needs to focus on when looking at your analytics reports. By developing the right key performance indicators (KPIs) you will be able to obtain real value from the overwhelming array of data the application makes available.

The easiest way to develop an analytics monitoring strategy is to structure it into steps:

1. State your overall site goals
2. List strategies and tactics your site adopts to achieve those goals
3. Define conversions, the key actions users take on your site to fulfill your goals
4. Identify metrics and KPIs that will allow you to track conversions
5. Set up Google Analytics to fulfill your needs
6. Report on your main KPIs

Step 1: State Your Site Goals

Before identifying what to measure, you need a clear understanding of the purpose of your help site. Begin by writing down the business goals your website supports and how your site specifically aligns to them.

For virtually all help portals, the overall business goals supported are to decrease customer support costs and increase customer satisfaction. And you do so by enabling self-service.

Example

Business goals:

1. Decrease support costs
2. Increase customer satisfaction

Help portal overall goal:

1. Enable self-service

Step 2: List Strategies and Tactics

Create a list of strategies that support your overall goals. Then list the tactics your site uses to help achieve each of those strategies.

Examples

Strategies:

1. Improve information findability

Tactics:

- a. Improve search results relevancy
- b. Make website navigation more intuitive

2. Increase customer trust of documentation

Tactics:

- a. Improve content quality
- b. Recommend relevant content as the user types the support ticket title

Step 3: Define Conversions

Think about the workflows your website's users follow, and which actions in those workflows fulfill your website's overall goals. Those actions are called conversions.

Websites have macro and micro conversions. Macro conversions are key actions users take that fulfill the broader goals of your website. Micro conversions are actions that lead to macro conversions.

If you provide online guided tutorials, a macro conversion might be completing a tutorial. A micro conversion might be advancing through one step of that tutorial.

For most documentation and support sites, one of the macro conversions involves the user not submitting a ticket. A micro conversion in that context would be clicking a link to an article that will answer the user's question.

Examples of Workflows and Conversions

Workflow 1:

- Customer searches/navigates to product page > clicks a link to a help topic > reads the topic > does not submit a ticket
 - Micro conversions:
 - Customer searches
 - Clicks a link to a help topic

- Reads the topic
- Macro conversion:
 - Does not submit a support ticket

Step 4: Establish Metrics and KPIs

The next step is to define the metrics that will be used to track the micro and macro conversions on your site. The metrics used to track macro conversions are your KPIs, as they indicate how your site is performing against the overall goals. Metrics used to track micro conversions are important to gain insight on how to improve workflows that lead to conversions, however, they are not your KPIs.

It's very likely that your KPIs will be calculated from raw metrics. Some of those metrics might be provided out-of-the-box by Google Analytics, while others may need to be customized. Identify what you need, and don't be afraid to get ambitious. Use an agile approach to create and iterate on your analytics implementation. Your minimal viable product (MVP) might include only Google Analytics default metrics. If you need more than that, you can iterate on the original implementation.

Examples

Workflow 1:

- **Action:** Customer searches
- **Metric:** Increase on % sessions with search
- **Action:** Clicks a link to a documentation piece
- **Metrics:** Decrease on % Search refinement, Decrease on % Search exit, and Increase on # visits to documentation
- **Action:** Reads the documentation
- **Metrics:** Increase on average time on page and Increase on # visits to documentation from returning users
- **Action:** Does not submit a support ticket
- **Metric/KPI:** Increase on % sessions without tickets

Define Dimensions, Segmentations, and Filters

Google Analytics allows you to add dimensions, and also filter and segment data to better understand user's behavior and get more accurate and meaningful data.

Dimensions

Dimensions allow you to add attributes to your metrics. For example, if a report shows search terms (a dimension), you could add a secondary dimension, like Start Page, that tells you which page the user started the search from. In Figure 1, the metrics are the quantitative data: *Total Unique Searches*, *% Search Refinement*, and *% Search Exit*. The dimension is the attribute: *Search Term*.

Segmentation

Segmentations allow you to isolate and analyze subsets of your data. You can, for example, analyze all sessions where a support ticket was submitted or all sessions where a mobile device was used. You can also create user segmentations, like customers and employees.

Filters

Filters can include or exclude data from a view. The most common view filter in Google Analytics is used to exclude the IP address of a company. You can use this, for example, to remove all traffic from employees from a reporting view.

Step 5: Set up Google Analytics to Fulfill Your Needs

It's time to get your hands on Google Analytics (GA), and I highly recommend pairing it with Google Tag Manager (GTM). This will greatly improve your ability to track site events and create custom metrics that are not tracked out-of-the-box by GA, like support ticketing submissions. Even if you don't plan to make use of events tracking or custom metrics in your MVP, you will most likely need to at some point. Google Analytics Academy provides steps on how to set up both applications.

Once you have GA and GTM set up, it's time to implement your analytics monitoring strategy. Start with the basics. The best way to get this done is to define a roadmap and make the tasks part of your agile backlog. Keep in mind that you may not be able to get very precise at first. Fortunately, you don't have to. Analytics is about establishing a baseline and tracking from there.

Below are some examples of what our analytics implementation for the PROS support site looks like. You will note that in some of the examples I used the [MVP] and [Iteration] tags to give you a better idea of how our implementation has evolved.

Filters

For one of our views, we used Google Analytics IP address filter to exclude traffic from our employees. Your IT department can provide the list of IPs for your company. If you have team members that work remotely, make sure to filter out their IPs as well.

Search Term ?	Total Unique Searches ? ↓	% Search Refinements ?	% Search Exits ?
	12,536 <small>% of Total: 61.48% (20,390)</small>	19.24% <small>Avg for View: 27.85% (-30.90%)</small>	10.86% <small>Avg for View: 19.18% (-43.34%)</small>
1. Search Term 1	58 (0.46%)	16.53%	25.86%
2. Search Term 2	42 (0.34%)	24.73%	14.29%
3. Search Term 3	37 (0.30%)	20.00%	37.84%
4. Search Term 4	34 (0.27%)	11.76%	5.88%
5. Search Term 5	27 (0.22%)	26.32%	7.41%
6. Search Term 6	24 (0.19%)	29.63%	12.50%

Figure 1. The metrics are the quantitative data: *Total Unique Searches*, *% Search Refinement*, and *% Search Exit*. The dimension is the attribute: *Search Term*.

Users Segmentation

[MVP] Since employees also use our support site for product help, we didn't want to completely ignore traffic from them. That's valuable data which allows us to compare the documentation needs of customers and employees. Therefore, we created a view that doesn't have the IP filter and added two segmentations by Network Domain: customers and employees.

Again, your IT department can tell you what your company's network domain is called. After that, we created a segment for employees by only including traffic from our company network domains, and a segment for customers by excluding traffic from our company network domains. We also added the network domains for remote employees that work on the site.

[Iteration] The set up in Figure 2 was good enough for our MVP, but we wanted to get more precise and more granular. We often heard from writers that the employee traffic to a specific product could become inflated by their own activity on the content pages as they were writing or editing articles. As a result, we decided to further segment employee traffic into employees who work on the website and employees who go to the site to get help. We also separated customer traffic into customers who manage support tickets and customers who don't have access to

tickets. This turned out to be a more advanced implementation, requiring help from our development team to send the user role from our site user profile to Google Tag Manager. It also required us to learn how the Google Tag Manager data layer works. Again, don't expect to have this kind of segmentation on your MVP.

Ticketing Deflection Rate

This KPI measures the effectiveness of one of our key workflows to encourage self-service. When the user starts filling out the support ticketing form, he selects product, enters incident title, and clicks a Get Help button. We then present a list of articles that are relevant to his entries. We did not track this in our MVP.

To implement this measure later, we added trackers to the *Get Help* button, the links to articles recommended, and the *Submit Ticket* button.

The formula looks like this:

$$\text{Incident Deflection Rate} = 100\% \text{ clicks on } \textit{Get Help} - X\% \text{ clicks on } \textit{Get Help} \text{ that resulted in a ticket}$$

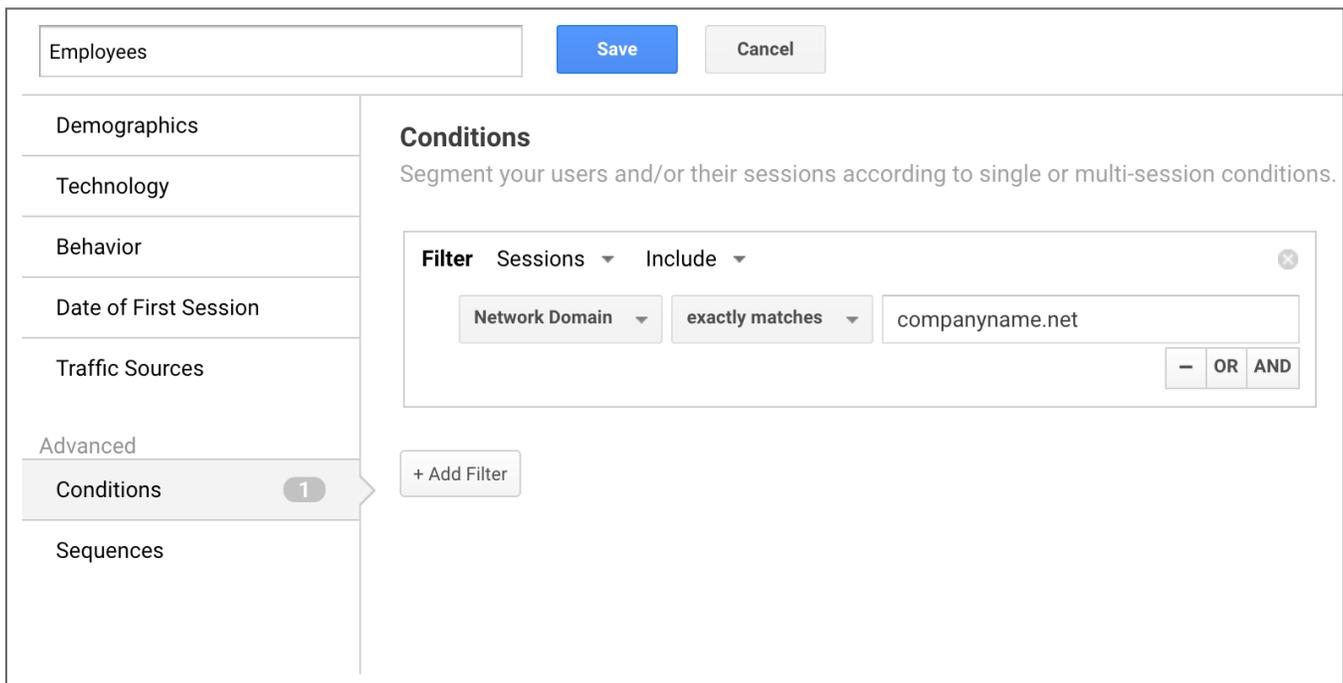


Figure 2. The set up was good enough for our MVP, but we wanted to get more precise and more granular.

Self-Service Rate

[MVP] Since our support site is the entry point for submitting support tickets, we installed a tracker on the *Submit Ticket* button to track submissions. If you don't have access to that data through your site, you could obtain it from your customer support team. In most cases, you can strip out feature requests and service requests, as those tickets are not likely to be deflected with documentation. We used the following formula for our MVP: Self Service Rate = $100\% \text{ Total Sessions from customers} - X\% \text{ of Total Sessions from customers that resulted in a ticket}$.

[Iteration] As we dug deeper into our analytics data, we realized there were a significant number of sessions where users went to the site exclusively to interact with an existing ticket. Our original formula was counting those sessions as successful self-service, meaning it was time to get more precise. As a result, we created a segmentation to exclude sessions where the user only visited existing tickets.

Search Performance

Search data is crucial to improve content findability and close content gaps on a help site. It allows you to identify which topics users are more interested in, measure how relevant your search results are, and identify content pages that users most often start searching from.

Google Analytics provides a rich site search report out-of-the-box, which includes all the search metrics and dimensions we identified in our measurement plan.

Here is how we use search metrics and dimensions to track performance and get insights:

- Sessions with search – as one of our metrics, this tracks how many sessions include searches, which might be the first action users take towards self-service.
- Search refinement and Search exit combined with Search terms – this is used to evaluate relevancy of search results. If customers refine their queries often or exit the site from the search results page, it's usually because search results are not relevant. We can use search terms with a high refinement or exit rate to identify content that needs to be created or optimized for maximizing findability.

- Start page - This is the page from which the user searched. If the start page is a help topic, the user probably wasn't satisfied with the information he found there.

Content Performance

[MVP] The Google Analytics Site Content report helps us track content quality globally and drill down to individual articles that need attention.

There are a few metrics here that are useful for identifying content that needs attention:

- Average time on page - This can be used to identify articles that are visited and then dismissed immediately, which might indicate a misleading title or metadata
- Unique Pageviews - Content with only a few or no visits should be reviewed for improvement or removal. This helps keep content current and accurate, and reduces noise in search results and navigation.

[Iteration] In addition to the standard GA set up, we created content groups to compare aggregated metrics by content type and product. This allows us to do things like identify, for example, the most popular products and content types across the site.

As an example, this sort of information influenced our decisions while considering the removal of system interface diagrams from the support site. Maintaining changes with every release was complicated and we never got any feedback on them from customers. However, by referring to GA, we could see lots of returning traffic for this content.

We use the table Content Drilldown to look at product level navigation. We select a product and can see the most popular content type in that context, most popular articles, etc. This data, along with search metrics, can provide insights to product teams. They might discover, for example, that a highly popular feature in documentation points to an area of the product that needs UX's attention, or that customers are searching for a feature that doesn't exist.

Step 6: Report

Your web analytics report should be sent periodically to your stakeholders and leadership team, even if you

provide a persistent dashboard. Keep in mind that your audience probably knows very little about your metrics or technical content, so a data dump of activity is wasted effort. Select the data that matters the most to them.

Be sure to focus on the metric that shows you are working to meet business goals. In our case, that would be the self-service rate. You can then present your most essential metrics, content, and search data that you want to share with product teams. Some of your metrics shouldn't be included in this report, but rather tracked internally to provide insights to your writing team.

Spend some time planning how to display your data. Make the report visually attractive and easy to digest. Include subheadings or name your KPIs in a way that explains the data in layman's terms. Instead of self-service rate, for example, we use "% of visits that didn't result on a ticket".

Conclusion

Having a web analytics monitoring strategy for your documentation can help you stay focused on meaningful metrics and KPIs. Using these, you will better be able to meet your goals and provide insights about changes that can lead to real-world improvements. Begin your strategy design by writing down the overall goal for your website, along with the strategies and tactics used to achieve that goal. Identify key actions users take on your site to fulfill your goal, then define the metrics that can be used to track each of those actions.

Approach the implementation from an agile perspective. Start small, with a minimal viable product, and improve on it. Don't treat analytics as a side project that you'll get to when you have some free time. Instead, once you have a strategy and know the data you need to collect, make the implementation part of your backlog and get serious about it.

Resources

Carlos, Jenn, Kelly, Travis, and Costa, Vivianne. "Measuring Content Like a Marketer. Getting Started with Web Analytics". Intercom 63.3 (March 2016): 6 -9. <https://www.stc.org/intercom/>

Avinash Kaushik. "Digital Marketing and Measurement Model". <https://www.kaushik.net/avinash/digital-marketing-and-measurement-model/>

Google Analytics Help. Dimensions and Metrics. <https://support.google.com/analytics/answer/1033861?hl=en>

Google Analytics Help. About Segments. <https://support.google.com/analytics/answer/3123951?hl=en>

Google Analytics Help. About View Filters. <https://support.google.com/analytics/answer/1033162?hl=en>

References

Google Analytics Academy. Google Analytics for Beginners. <https://analytics.google.com/analytics/academy/course/6>

Google Analytics Academy. Google Tag Manager Fundamentals. <https://analytics.google.com/analytics/academy/course/5>

Google Analytics Help. About Calculated Metrics. <https://support.google.com/analytics/answer/6121409?hl=en>

Author Contact Information

Vivianne Costa
Senior Product Manager
PROS, Inc.
2831 Turquoise Cir, Davidson, NC 28036
281-990-4320

Author Biography

Vivianne Costa has been a strong customer experience advocate for the past 15+ years, working first in digital marketing, then diving into the product world as senior product manager for a customer help portal.

Starting from scratch in 2001, Vivianne taught herself how to use analytics and earned a Google Analytics Certification. Over years of hard-won experience, she has designed effective analytics strategies for many kinds of web portals, including, most recently, a documentation and support platform.

A passionate advocate for useful analytics, Vivianne is always excited to share how to transform raw data into actionable insights and is keen to help analytics

newbies create meaningful reports that can drive leadership decisions and real-world improvements that directly impact customer experience.



Coules, David

Disrupt Your Own Self: Streamlined Publishing Through the Cloud with HTML5

There is growing unrest in organizations and their audiences – caught in a world of beautiful, functional websites that ultimately lead to the information they were looking for, posted as an oversized file of static printed pages that are cumbersome to use and inconsistent with the experience they just left. This article details this problem, the technologies and methods that are being used to solve it and illustrates with short case studies an emerging movement that is challenging accepted norms about the cost and complexity of producing professional and technical content.

The Problem

It's 2017 and despite the pace of new technologies being introduced – rich, cloud-based application services, Internet of Things, Augmented/Virtual/Mixed reality, machine learning, big data – we see a widening gap between the practices used to create these new, exciting technologies and those being used day in and day out to produce professional material in document form.

This is despite the fact that we continue to fund an enormous volume of documents that we expect people to use – whether these are commercial publications or material that supports products and services.

There are surprisingly fragmented and slow delivery channels for something that should be available instantly. The workflows can be archaic and often we see practices that primarily target a static print format with interactivity as a downstream add-on or simply an afterthought that ultimately no one gets time to finish.

Creating brilliant, interactive material takes knowledge, effort and talent - as I learned for a large part of my career in managing the production of learning products. However, even the most basic expectations that we have for websites is completely absent in

equally or more important assets when it comes to professionally produced technical material or compliance documents.

Whether it is mandatory medical, legal, engineering or policy information, we have not escaped the tyranny of putting up with what are essentially print pages in a file you can view on screen. Why is this? Does anyone really read this stuff? How do you know – do you get real-time data to understand how people are consuming this material? Why not?

We consistently see operations that are still experiencing many of the following pain points:

- Multiple versions stored in many varying formats
- Editors juggling various editions and versions in a non template-driven environment
- Editors concentrating on layout instead of content
- Delays in reaching target audiences
- Little or no automation in the publishing life-cycle
- Staggered timelines depending on type of output (operating systems and versions, device types, app store approvals)
- Re-inventing the wheel for each type of output

- Not reusing or storing valuable corporate information efficiently
- No extensibility for future marketing opportunities and publishing demands
- No proven quality assurance.

There are certainly organizations that have put in place solutions that deal with these issues head-on however the level of investment is typically prohibitive for all but the very large and very specialized of workflows. The procedures and tools introduced tend to target a presumed level of complexity that wasn't really meant for the "rest of us".

Plenty of platforms and services offered by very capable providers are out there offering a myriad of authoring solutions that output to an extensive array of formats, catering primarily to people managing or doing the publishing process.

The downstream complexity that this entails is forcing operations to look to third parties to finalize to the varying different channels and are invariably distancing themselves from the very person they created the content for in the first place.

It is becoming urgent that we address this as a community of writers, publishers and organizations that rely on getting vital information to their audience and that this audience enjoys a level of usability, engagement and functionality that we've come to expect with most other services.

The urgency comes from these expectations and the lack of engagement in digital versions of print pages. We see this reflected in a growing disquiet across all industries in terms of what they and their audience are getting out of their investment.

Importantly, the approach should be something that reaches the smallest of budgets, and is simple to understand in the same way that so many Software-as-a-Service platforms are these days – they make it easy to engage and where necessary, teach people along the way through self-discovery models. It's an experience that reinforces validation of "This is going to work".

Technology Enablers

Fortunately, the building blocks for solving this have been available for some time. HTML5 and supporting

browsers already on our devices is one of these ingredients, and through direct experience, has unleashed the power of the web on our humble document.

For some who have been in the industry for as long as we have, it is a bit surreal how the world seems to have re-discovered XML as an enabler for building new workflows that make it easy to attack these pain points.

While some of these technologies have a long history, when coupled with newer technologies and modern practice in application development, they can bring a level of ubiquity and scalability that serves as the fuel required to deliver a consistent experience across all content – from data visualization and CMS based information on your website through to highly technical works in document form.

Starting with HTML5, released as a stable W3C Recommendation on 28 October 2014, fundamentally important features were introduced, including:

- HTML5 as a markup language in its own right (no longer SGML-dependent) to support ease of implementation
- New structural elements akin to XML, an important building block for end to end automation
- Dynamic pages support for rendering complex content
- Simple tags for rich media elements video and audio
- New input element types to support better interactivity
- Custom Data attributes to enable sophisticated application behavior in a web browser

This further enabled the introduction of new concepts around achieving a level of parity with applications running natively on a device – without the hassle of supporting different operating systems and devices. One key concept is a Progressive Web App – an application that runs in any web browser that is responsive to fit in any form factor. It is also independent of connectivity, working offline or online and it "feels" like an app because it models separation of content from functionality.

This is not an argument against native apps because they are critical where raw compute power is unable to support the level of abstraction that web browsers

afford – e.g. Games. The benefit at stake here is that for most content typically delivered in document form – including embedded interactives and rich media – abstracting to a web browser opens the door to instant delivery models with website-like features. This further brings a level of agility to publishing that is strikingly more efficient compared to current common practice.

Rather than being a disconnected solution, bringing in a philosophy of exposing access to the technology is another trait commonly found in modern implementations. Often referred to as “API-first” (Application Programming Interface), this translates to the web app being something that can be readily mixed or integrated with other apps to create a best-fit solution in a relatively short space of time.

While technically not mandatory, coupling this with Cloud Computing further enhances the delivery model, bringing all of the associated benefits, including:

- Work from anywhere
- World-class availability for service levels and integration opportunities
- Flexibility with scale and localized presence
- Software updates managed for you
- Pay-as-you-go, capital expenditure free
- Security from lost devices

Finally, a sustained mindset for the principles of continuous integration and continuous delivery rounds out how we can capitalize on the benefits afforded by technology that is available today. Put more simply, applying vigilance to an end-to-end automation mindset completes the picture in terms of real benefit that can be gained by applying the above technologies and approaches.

In Practice

With platforms like tekReader, this can be brought into perspective not only in more obvious industries such as engineering and manufacturing, but with self-publishing models being explored at the smaller end of town or, in a larger context, departments with tight budgetary constraints; all with the benefits described above and, in true SaaS manner, allowing traditionally constrained operations to deliver world-class experiences to a global audience.

Self-Publishing Independence for Law Firms in the Mobile Cloud

The legal industry’s trend to embrace more flexible business models is driven by well documented forces: the changing market, modern work practices and ever more demanding clients. The 2016 American Bar Association (ABA) Legal Technology Survey found 93% of lawyers use a smartphone, 70% of whom use their smartphones for legal work while in the office. It also found 51% of lawyers use iPads or other tablets for work.

Responding to these trends, an Australian law firm saw a business opportunity to securely publish to all digital channels using a cloud-based SaaS solution, ensuring the purchasers and readers of its legal publications could get instant access via any device whether laptop, mobile or tablet.

It was vital that while pursuing this option, efficiency of process to contain costs was a mandatory constraint as they looked to expand from a traditional print offering. Equally important was the need to provide a seamless and engaging experience that would represent an innovation in the eyes of their customers as this is a direct revenue channel as opposed to a supporting resource.

Using an API-first application also made it feasible to connect eCommerce via PayPal and provide a complete revenue-earning digital solution.

In summary, this enabled the law firm to use technology to facilitate growth and client engagement with a digital self-publishing model that provided a means of authoring material that both acted as a revenue-earner as well as being an innovation showcase from a marketing perspective.

Publishing Startup in Education

Class2Digital – an educational publishing startup – sought to introduce a business model that would empower university professors to publish textbooks direct to market. Their motto “Teaching is a complex science... Publishing doesn’t need to be” illustrates modern practice in technology as a revitalized enabler.

Of particular interest is that Class2Digital’s requirements were for financial textbooks 300-400 pages in print-equivalent form, full of complex tables and equations, targeting a “Bring Your Own Device”

ecosystem that included mobile as well as tablet and laptop experiences – all in French, as they are based in Quebec. Furthermore, they needed to integrate with their pre-existing eCommerce solution and subsequently, formative assessment content (quizzes), with plans to do more with rich media down the track.

This was all within the context of a small startup, the team having no prior experience with XML or XML authoring tools. This case study has all the hallmarks of the sort of disruption that can be common with cloud-based technology.

Just Getting Warmed Up...

The previous case studies illustrate these enablers along a more traditional line, comparing publishing opportunities and demonstrating efficiency and functionality that supports complex content in a seamless manner.

More interesting are the flow-on benefits once a web-enabled content delivery channel can be assumed:

- Enabling quick integration to create new solutions such as embedding a full reader-based experience for extensive documentation inside a CRM system, driving exceptional levels of responsive customer interactions.
- Analytics – collecting rich information on document use – finally information is available to help feed back into the content development process. Turn your cost center into a revenue contributor by analyzing this data for patterns where people are accessing a product troubleshooting section more than expected.
- Future connection and rendering with the Internet of Things – real-time, customized/ personalized information inline with what used to be static documentation. Mixing this with augmented reality has all sorts of possibilities once content is no longer locked away in a static, printed page equivalent.

In summary, with the right combination of technology and insight into what makes a publishing process work, we can now deliver rich technical content to one single form supported across all devices, saving significant cost and turning the audience experience into something truly inspiring or simply getting out of the way so that people can get on with it.

References

- Erl, Thomas; Mahmood, Saigham; Puttini, Ricardo. Cloud Computing – Concepts, Technology & Architecture (Upper Saddle River, NJ: Prentice Hall), 2014.
- Humble, Jez; Farley, David. Continuous Delivery – Reliable Software Releases Through Build, Test, and Deployment Automation (Upper Saddle River, NJ: Addison-Wesley), 2011.
- Street, Aaron, “Mobile Technology ABA TECHREPORT 2016.” American Bar Association website. <http://www.americanbar.org/publications/techreport/2016/mobile.html>
- LePage, Pete., “Your First Progressive Web App.” Google Web Fundamentals. <https://developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp/>
- HTML5. (2017, March 31). In Wikipedia, The Free Encyclopedia. Retrieved 07:47, April 15, 2017, <https://en.wikipedia.org/w/index.php?title=HTML5&oldid=773107857>
- Continuous integration. (2017, April 11). In Wikipedia, The Free Encyclopedia. Retrieved 08:02, April 15, 2017, from https://en.wikipedia.org/w/index.php?title=Continuous_integration&oldid=774969583
- Continuous delivery. (2017, March 24). In Wikipedia, The Free Encyclopedia. Retrieved 08:03, April 15, 2017, from https://en.wikipedia.org/w/index.php?title=Continuous_delivery&oldid=771967054

Author Contact Information

David, Coules, CEO
eGloo Technologies
B121/20 Lexington Drive
Bella Vista, NSW 2153
Australia
61.2.8814.7533

Author Biography

With 20+ years experience in technology and professional management, David works closely with the team at eGloo to develop key programs on both services and platforms that enhance benefit to its clients, leveraging the sought-after talent that has built eGloo’s reputation over the last 15 years.

David has held numerous leadership roles in global organizations spanning technology, operations and program delivery - roles that consistently involved digital transformation and working with people to build a lasting and inspirational change.

This includes the introduction of large-scale, XML-based content management systems and following through with multi-channel, eCommerce-enabled web platforms.

A genuine technology enthusiast and a life-long learner, David can also be found at various technology meetups across Sydney. David is co-organizer for the IoE meetup group in Sydney focussing on the Internet of Things.

David holds an MBA from the Macquarie Graduate School of Management in Sydney as well as an engineering degree from Queen's University in Canada.



Cressey, Eric

User Experience Strategies for Winning Application Programming Interface (API) Documentation

Documentation is a critical component of an API's success, but it's often an afterthought for development teams. This paper explores the strategies that API writers can use to work more effectively with developers, product owners, and customers; identify API usability issues; and write winning API documentation.

Introduction

As API writers, we're responsible for a large part of an API's success or failure. Developers count on API documentation to learn new APIs as quickly and painlessly as possible. Being responsible for such a highly visible part of an API user's experience is rewarding and challenging.

In a study of more than 1,500 bugs for Eclipse, Python, and other projects, researchers found that 37% of those bugs were related to usability. Of those usability issues, 27% were documentation problems like typos and missing or incorrect descriptions and examples (Zibran, Eishita, and Roy 2011, pp.152-153).

In another study, researchers asked developers at Microsoft about their strategy for learning an API. 78% of the 83 respondents said that they read documentation and 55% said that they read code samples. When the developers were asked about obstacles to learning, 50 out of 74 cited problems with API resources like documentation and code samples (Robillard 2009, pp. 28-30).

Of all the usability issues that were identified in these studies, like complexity, missing features, and error handling, documentation stands out. It's one of the only aspects of a public API's usability that you can improve after the API is published. If API development teams want to improve usability, writing better

documentation is one of the most straightforward ways to do it.

Development teams don't always think about usability as they design and build their APIs. When usability isn't a team priority, documentation happens later in the process when writers don't have the time to find and resolve usability issues and write the required documentation. This doesn't have to be your situation. With the right strategies, you can work more effectively with your team and customers, identify API usability issues, make usability a team priority, and write winning API documentation.

Working with Development Teams and Customers

I don't think you can write great API documentation without help from your development team and customer feedback. Working with developers, QA engineers, management, and customers is a challenge because each group has different information and priorities around the API.

Working with Management

In agile software development, product managers or product owners shape the overall product vision and determine what features it needs. Product owners

care about growing the business by reducing costs or increasing profits.

Knowledge areas:

- Product vision, business cases for integration, customers introductions, and use cases for features.
- Strategies:
- Use stories to strengthen your arguments. If you want to fix a usability issue, make your case to management with a real customer story. Listeners empathize with personal stories.
- Gather metrics and use them. Get metrics on user retention and customer escalations. Use web analytics to track page flows, visits, time on site, etc. Show that your work reduces costs or increases revenue.
- Spend time cultivating this relationship. Product owners hold great influence over development teams. If usability is important to the product owners, it'll be important to the team, too.

Working with Developers

Developers like writing code and want to spend their time doing that.

Knowledge areas:

- High-level API design information, technical reviews, feedback and direction on samples and tutorials.

Strategies:

- Learn the API, learn to read code, and learn the technology your API is built on.
- Lurk in all the developer meetings you can attend. Help name things and point out complexity or inconsistency when you see it.
- Use real customer stories and bring up usability when you can. A good API is a usable API!
- Make API documentation part of a larger API usability discussion.

Working with Customers

If handled correctly, customer escalations are a great source for recruiting usability study and survey

participants. Make it a habit to speak with customers as often as you can.

Knowledge areas:

- Background information for metrics and personas, use cases, usability feedback

Strategies:

- Take good notes when you talk to customers. Using customer stories to illustrate usability problems is a great strategy for working with management and developers. Write stories down and remember them so you're ready to use them when necessary.

Positioning Yourself for Success

Make your documentation a usability task and communicate the relationship between documentation and usability to your team. Most developers are interested in at least some aspects of API usability because good APIs are easy to use and maintain. Many usability issues must be addressed during API design, so you'll be involved earlier in the development process. That means you can help name things, facilitate usability discussions, and effectively plan documentation requirements that include high-level API design and targeted code samples.

In an agile development environment, you should be busy in every stage of the product development lifecycle. Engage product owners during release planning and get involved in sprint planning. Make sure that you're around for important discussions and that features have documentation or usability tasks associated with them.

Be active and look for opportunities to improve usability. For public APIs, you only get one shot at a good API.

Evaluating API Usability Issues

Since we're positioning documentation as a component of usability, you need to be ready to evaluate APIs for usability issues. The heuristics for evaluating user interfaces also work well for evaluating API usability. Nielsen's ten usability heuristics are a great place to start (Myers and Stylos 2012, pp. 66-69).

Key heuristics:

- Match between system and real world. Working with the API should be intuitive. Names must make sense. Naming things is a famously difficult task for developers, so help your team. If you're writing an API for a product with a user interface, make sure the API is consistent with the user interface.
- Consistency and standards. The API design should be consistent across the entire API. As you're reviewing the API, look for anything unusual or out of place. Get to know the API so you're positioned to do this task (Zibran 2008, p. 258).
- Error prevention. The API should guide the user to correct use and make it hard to misuse. Make sure that defaults make sense and do the right thing (Bloch 2006).
- Recognition versus recall. Clear, understandable names help users understand the API without consulting the documentation (Bloch 2006).
- Flexibility and efficiency of use. Users should be able to efficiently accomplish tasks with the API. User tests, where actual users try to use your API, are a great way to gauge efficiency.
- Help users recognize, diagnose, and recover from errors. Many APIs provide limited or unhelpful error information. Unhelpful errors impact usability and can cause other problems.
- Help and documentation. This is the most important one for writers. People complain about API documentation issues a lot.

Heuristic evaluations are great when you're designing an API, but they're not the only tests you can do. API usability studies, interviews, and surveys are some other excellent ways to capture user information (Grill, Polacek, and Tscheligi 2012, p. 168). Whatever you do, measure everything.

Building API Documentation that Meets User Needs

As with other writing tasks, there's a lot to do before you begin. Working effectively with your team, talking to customers, and evaluating usability issues will prepare you for writing great documentation.

What do developers need in API documentation?

1. Complete, accurate reference information.
2. Code samples and tutorials that demonstrate best practices and basic and high-level API use.
3. Guidance on high-level API design so users know when and why to use the different pieces of the API.
4. Simple organization with a useful search and a manageable table of contents. Provide a search even if you're using a single-page output.
5. Use topic-based content. Don't make the reader think about the type of content they want to read.

Automate Reference Documentation

Reference documentation is painful to maintain by hand and it's often a source for documentation issues. If you can automate it, you'll have more time to focus on other aspects of the documentation. Make sure that the descriptions in the code are helpful and intuitive.

Create User Personas for the API and Use Cases for each Feature

Do your users prefer searching or browsing? Do they have advanced domain knowledge relevant to using the API? The more you know about your users, the more you can tailor your documentation to them.

Use Analytics

You're probably not going to fix a problem you don't know about. Use analytics in your online help to track page flows, page visits, search terms, page visits after search, and more. Try to understand how easily people find what they're looking for. Set up reporting and set aside some time to look at the reports on a regular basis.

In addition to analytics, gather metrics from other sources like management and support. Ask support for the number of API escalations they deal with each week. Ask management the number of new API users each month. When you have all of these metrics you're able to see how things change as you make improvements to the API documentation.

Provide Good Examples

In one study where developers were asked to use an API, nearly every line of code they started with came from example code in the documentation (Meng 2016).

Here are some tips when you're creating examples:

- Make code sample read like prose. If it's hard to read, it's not going to be very useful (Bloch 2006).
- Use a style guide for each language you support. Enforce consistent style for all of your code samples.
- Use snippets to demonstrate basic API functionality.
- Use tutorials to cover a specific aspect of the API.
- Provide code segments from complete applications so developers can see how the API works in the real world. (Robillard 2009, p. 32).

The most common frustration with code samples is that they don't show how to put things together. Your samples should show functionality, explain when and why to use a feature, and explain how the feature fits with the rest of the API to accomplish user goals (Robillard 2009, p. 32).

Conclusion

API documentation is a huge part of how developers learn an API. Unfortunately, it's also a source of API usability complaints. As writers, we can position ourselves to write better API documentation by focusing on usability, improving our team relationships, staying engaged the entire development process, and by collecting and leveraging user feedback to measure our progress.

Resources and References

- J. Bloch. How to Design a Good API and Why it Matters. OOPSLA, Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications (October 2006) 506–507.
- Grill, Thomas, Ondrej Polacek, and Manfred Tscheligi. "Methods Toward API Usability: A Structural Analysis of Usability Problem Categories" Proceedings of the 4th International Conference on

Human-Centered Software Engineering (October 2012) 164-180.

Meng, Michael. "API Documentation: Exploring the Information Needs of Software Developers" (Write the Docs Europe Conference, 2016).

Myers, Brad and Jeffrey Stylos. "Improving API Usability" Communications of the ACM 59.6 (June 2016) 62-69. <https://cacm.acm.org/magazines/2016/6/202645-improving-api-usability/abstract>

Robillard, Martin. "What Makes APIs Hard to Learn? Answers from Developers" IEEE Software 26.6 (November 2009) 27-34. <http://ieeexplore.ieee.org/document/5287006/?reload=true>

Zibran, Minhaz. "What Makes APIs Difficult to Use?" IJCSNS International Journal of Computer Science and Network Security 8.4 (April 2008) 255-261.

Zibran, Minhaz, Farjana Eishita, and Chanchal Roy. "Useful, But Usable? Factors Affecting the Usability of APIs" Proceedings of the 2011 18th Working Conference on Reverse Engineering (October 2011) 151-155.

Author Contact Information

Eric Cressey
Principal Information Developer
Symantec
350 Ellis Street
Mountain View, CA 94043

Author Biography

Eric Cressey is a Principal Information Developer at Symantec. He has been a technical communicator since 2011 and specializes in API documentation and usability, building custom outputs for web help, user interface content, and task automation.



DeLuca, Todd

How May I Help You? Why Service is a Smart Way Forward

As technical communicators, we're typically working behind the scenes and missing an opportunity to establish longer-lasting impressions with our users or consumers (possibly a relationship). It is like being a cook in a restaurant, where the work is done behind closed doors and there is no personal connection is made. However, the server is the one who delivers, makes the connection, and gets recognition for passing on the work of the cook. Providing service and getting closer to the customer is ultimately the higher-value and better long-term path for career advancement. Get out of the kitchen!

Imagine these two scenarios:

- You go out to a nice meal at a fancy restaurant with great service and incredible food, perhaps a high-end steak restaurant.
- You go out for a more casual meal with decent food and fine service.

Who Do You Remember from That Experience?

In both cases, you likely remember the person who served and delivered the food. Not the person who prepared it. It doesn't matter how much effort, time, and energy was put into preparing the food - the customer who eats it doesn't see or appreciate what happened before it was delivered. Most of the time they only care if the food is decent and ready quickly - if the order is accurate and cooked correctly. If the food is great, then it's a bonus. If it's bad, then they really notice and may possibly complain. People don't typically meet or know the chef or cook. Ultimately there's no visibility or connection with the cook for the majority of people. It's the person you interacted with and met in person that you remember (or would recognize later). That's the server. Your overall experience is shaped by the quality of the interactions.

Which One Are You? Cook or Server?

Many technical communicators are more like cooks or chefs. Our focus is on production - either making or helping produce things. Typically, our work is in the kitchen, behind the swinging doors. We gather the ingredients (content), prepare and cook the food, follow recipe (consistent), and plate it for consumption by the customer. So, what?

As long as people get good food and what they ordered. However, people don't see or know what happens before the food comes up and is served, nor do they really care. Our skill and value is hidden away. They don't think about the people making the food, they ultimately just want to eat and solve their immediate hunger problem. And it's the server that they see ultimately serving the food and satisfying their need (solving their hunger problem). They might appreciate who prepared the food, but we rarely see that person.

Often we and others talk about what we do as making or improving things - working the ingredients in such a way to create a great dish. The focus is typically on the craft and production skills. For instance, I and others would say that my team produces product documentation and release notes that our clients

and users consume to know what's changed in the software. We prepare the dish that clients consume. In this case, release notes are like tapas (small and bit size)

What's Wrong with Being a Cook?

Nothing. It's not bad, but production is not a skill that differentiates you from someone else who can cook. Production is also much easier to outsource (done cheaper) or automate, just learn and acquire the skill. Almost anyone with the recipe, training, and some practice can make a comparable meal (prepared to taste similarly). From the best top chains to the lowest (price and quality) - they all have systems, recipes, standards, and training (and do it well). All they must do is source ingredients and plug in people to the system.

Where Is This Headed?

Does this sound like the sort of work that could be automated? Have you heard about robots and machines that now prepare food? As cooks or chefs, we face a similar challenge in the years ahead when it comes to preparing content.

Automation and artificial intelligence and coming and likely to take more and more pieces away. If there is a less expensive or quicker way to do something comparable to what you're doing (or making), then they will go for that option. Outsourcing of technical writing was just a first step and is only expanding.

COMPlimentary

Service is COMPlimentary:

- **Chemical** (triggers feeling or emotion in those you serve)
- **Opportunity** (to build a relationship)
- **Memorable** (memory and feeling lasts longer)
- **Personal** (connects you directly with another individual)

Who Gets the Recognition?

In the best restaurants, who are the people that are paid well and are often there for years? The ones that last and stick around. It's the servers. Cooks and chefs come and go (higher turnover for good ones).

When you have an incredible experience and your expectations are exceeded, who do you want to recognize and give a big tip to? It's those great servers! You should focus on adding service to your repertoire as an important means of connecting with people - it will matter more over time. It's a chance to be intentional to establish a potential relationship. You can make an impact and build influence by doing a very good job and more than what's expected by your potential customers or clients.

Consider an Alternative to Being a Server

If you're not ready to fully jump ship into service (or away from production), you still need to interact and get closer to the consumer. My suggestion? Get out of the kitchen and provide the meal you prepared to customers so they remember you and connect you with both the service and meal). It's unexpected, appreciated, and get that personal, albeit limited, connection. It's a progressive step in the right direction (for you and career).

Be the Server! or Add Delivery as an Add-On Service

The lasting, personal, and memorable relationship you build by providing excellent service is key to your successfully connecting with people. Going forward, your skills and experience will have less impact on your career than the people you serve and connect with. You want to be remembered for who you are, not what you do or what you make.

Be the server! (Or add delivery as a value-add service offering)

Author Contact Information

Todd DeLuca, , Associate Fellow
TechCommTodd@gmail.com

Author Biography

Todd DeLuca has over 15 years of experience as a technical communicator and currently manages a remote team of technical writers in the US and India. His professional background includes graphic design, editing, client communication, and software development documentation. Todd is an STC Associate Fellow and Conference Chair for the 2017 Summit. He was President of the Philadelphia Metro Chapter president and has written articles for the STC Intercom magazine. Todd has a Masters Degree in Technical and Scientific Communication from Miami University (Ohio) and speaks at various regional and national conferences.



Elle, Edna

Accessibility Made Easy: Automating and Optimizing Accessible Outputs from DITA XML

*This presentation is geared toward information architects, technical writers, and managers, who are using DITA XML authoring and rendering tools and who have a **requirement** to produce accessible HTML output. Making accessible HTML output is time-consuming, expensive, and error-prone. It is also probably a business requirement for your company or organization. This session shows how to automate your way to accessible output and “gain the edge” using DITA XML to optimize your HTML results quickly and efficiently.*

Making accessible HTML output from DITA out-of-the-box is difficult because the DTD lacks certain enforcements and because some default attribute settings produce the least desired result. Learn what you can do to enforce your way to accessible output using DITA XML and to optimize your HTML results.

This session describes:

- How to enforce inclusion of accessibility requirements with minimal DTD specializations
- How to move much of your testing from HTML output to XML, where issues can easily be detected in the sources and corrected before rendering to HTML
- How to optimize your output rendering tools and stylesheets
- Where and how to focus your testing efforts
- The most common and avoidable DITA HTML output mistakes (Hint: table rowheaders).

The Two Most Critical Areas: Graphics and Tables

The two most difficult accessibility areas are graphics and tables. Graphics require alternate descriptions, and tables require summaries or captions and markup

about how the tables should be read by screen reader software. Additionally, tables must not be overly complex to avoid screen reader users from becoming disoriented in the table.

Graphics

To meet accessibility requirements, graphics must include text descriptions. Standard DITA does not enforce this requirement, so you want to include rules in your customization that do. Leaving it to the authors to choose the right thing imposes an undue burden on them and can lead to a confusing experience for screen reader users.

DITA has two elements for text description of graphics: `<alt>` and `<longdescref>`, both of which are optional in the DTD. You must include one or the other, a choice depending on your graphic, but never both and never none.

If you chose none, then unless your graphic is part of the template and decorative, you have created an accessibility error. If you choose both then screen reader users do not know which one is the correct alternate description and why they would use one description over the other.

The <alt> element is intended to contain a short line of text that briefly describes a simple graphic such as a button, icon, or logo. In the HTML output, alt is an HTML attribute value that is read only by screen readers. Because it is an HTML attribute, the <alt> XML element cannot include inline markup that will not render as expected.

The <longdescref> element is for complex graphics that require lengthy descriptions. It may include multiple paragraphs, lists, code examples, tables, or other HTML markup - it is an entire topic.

Change the DTD to make it a requirement to have <alt> or <longdescref> as a child of <image> or <object> but not both and not optional. Not requiring either <alt> or <longdescref> makes it easy to skip this critical requirement. If you depend on authoring instructions to optionally include these elements then you have lost the ability to enforce their proper inclusion.

Optionally, allowing both <alt> or <longdescref> or requiring both frustrates users and is incorrect because users cannot tell if they have enough of a description from the alt attribute alone - how do blind readers know if the alt read to them is sufficient if you also provide a link to a longer description? How do they know that they should or should not link to the <longdescref> topic? Additionally, allowing both or requiring both needlessly links users to long descriptions that may duplicate what is in the HTML alt attribute.

So, how do blind users know that there is a more detailed description for the graphic if you have used <longdescref>? You change the HTML output to automatically provide a boilerplate HTML alt in your rendering output that references the link to the <longdescref> description or topic, thus, informing readers that a link to a description follows (Figure 1).

If the writer supplies <alt> rather than <longdescref>, then the graphic gets an HTML alt attribute and it is presumed to be enough.

For most technical documents, a short description in an HTML alt attribute is not the equivalent of seeing a complex graphic such as a flow diagram or a screenshot with callouts. To ensure that visually disabled readers have the information they need without making it cumbersome for visual able readers, linking to a long description is the best option for complex graphics. Therefore, <longdescref> is often the best choice for many graphics.

Remember also to customize the DTD for the <alt> element to disallow inline markup elements such as , <i>, <sup>, <sub>, <u>, <code>, etc. The inline elements, when rendered as an HTML attribute value, cannot include markup - the markup would be removed by the rendering tool and the meaning may not be what you intended.

Enforcing these rules in the DTD ensures that there are clear graphic descriptions, short or long.

Tables

There are two types of tables for accessibility purposes, layout tables and data tables. Screen readers read these two types of tables differently.

Layout tables do not have row and column headers that can be mapped to information in the table cells and screen readers read the content of table in a linear fashion, left-right, top-down. Also, layout tables must not have an HTML caption element or a table summary attribute.

Data tables include the relationship between two or more items, in rows and columns. Data tables must have an HTML caption or table summary attribute and markup that tells screen readers how to read the table.

How do screen readers “know” if a table is a layout table or a data table? Screen readers make this determination based on the markup. HTML <th> and <caption> elements and the summary attribute (deprecated in HTML5 but mandated by many companies including Oracle) are screen reader clues. Using <th> and <caption> elements (optional if summary is used) and summary attribute values in layout tables is incorrect and causes screen readers to navigate those tables incorrectly. Conversely, the lack of the HTML <th> and <caption> elements and summary attribute values causes screen readers to navigate a data tables incorrectly.

DITA it makes it difficult to clearly distinguish between layout and data tables. For example, DITA allows you to indicate an attribute that provides a rowheader in a layout table where there should be no rowheaders, just information read in a linear fashion. However, the problem can be solved by customizing the DTD and segmenting the existing DITA tables in the following ways:

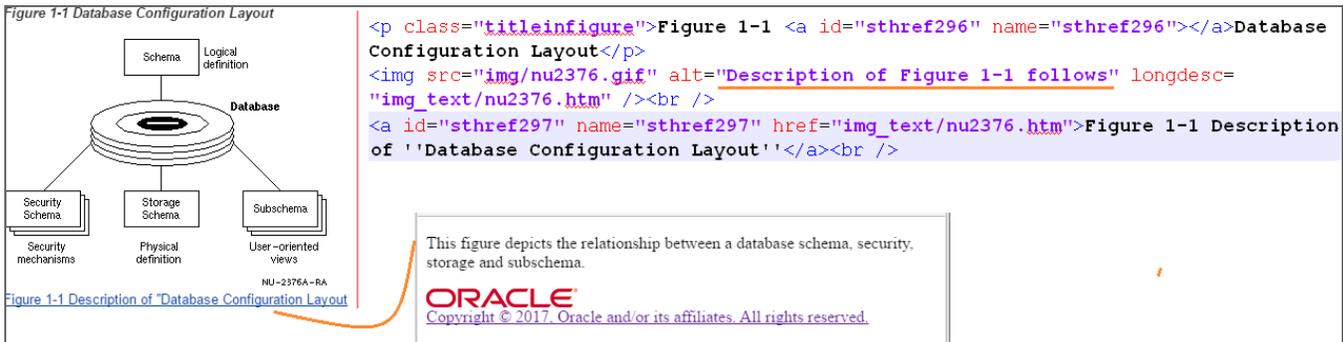


Figure 1. So, how do blind users know that there is a more detailed description for the graphic if you have used `<longdescref>`? You change the HTML output to automatically provide a boilerplate HTML alt in your rendering output that references the link to the `<longdescref>` description or topic, thus, informing readers that a link to a description follows.

For layout tables:

- Use only `<simpletable>`
 - Do not customize `<simpletable>` to add a `<desc>` child element or provide any element that outputs as an HTML `<caption>` element or summary attribute
 - Customize your HTML rendering tool to add `role=presentation` to the HTML that precisely identifies a layout table
 - Disallow `<thead>`, which avoids the table becoming a data table-type table because it has a heading row
 - Disallow the `keycol` attribute which outputs rowheaders that should not be in a layout table
 - Allow only two `<stentry>` elements in a `<strow>` element, this ensures a default linear read and avoids the temptation to span rows or columns
- Customize `<row>` to require two or more `<entry>` elements to avoid a single-column layout table
 - If you require table legends or keys, specialize an outputclass attribute value to produces an HTML `<tfoot>` element and, in your table summary, add boilerplate that alerts “This table has a legend”

These changes allow you to always make `<table>` into data tables that have proper screen reader markup when rendered.

Understanding DITA’s Default Table Rendering

The most important thing to understand about DITA tables is that all versions assume the `<table>` `rowheaders="norowheader"` value as the default. This means that all data tables, unless marked as `rowheaders="firstcol"`, read only top down. For most tables, this is NOT true. Usually, the first cell in each row is also important to reading the table - meaning the first cell in each row is usually the rowheader for the rest of the content in that row. This caused almost 100% of the tables in one of our major product releases to be marked not accessible!

Instead of having all authors “fix” the rowheader value, change the default rendering behavior to process as if you have set your table to `rowheaders="firstcol"`. If you do this, most of your tables will “read” correctly by default. Oracle’s HTML DITA rendering tool assumes `rowheaders="firstcol"` as the default and respects `rowheaders="norowheader"` only if the author had explicitly selected it. This saves time and errors because the output assumes the way that most writers have authored their tables.

These changes allow you to always make `<simpletable>` into HTML layout tables.

For data tables:

- Use only `<table>`
- Make `<desc>` a required child element of table
- Output `<desc>` as an HTML summary attribute (usually preferred) or a caption element
- Customize `<desc>` to disallow elements such as ``, ``, `<pre>`, `<image>`, etc. that cannot be expressed as an HTML attribute
- Customize `<table>` to allow only one `tgroup` child element to avoided nesting or “table units”
- Customize `<table>` to make `<thead>` a required child element

Too Many Table Elements Choices

DITA provides numerous table elements that could, if used properly and carefully, create accessible tables but there are no requirements or enforcements. Having too many choices and combinations makes it more difficult to achieve and enforce accessible output.

When it comes to accessible tables, simple is best. The DITA DTD permits nesting tables, spanning rows, spanning columns, merging cells, leaving cells blank, “table units”, and more. The problem is that screen reader users cannot navigate such complex tables and get the same meaning as a visually able reader. They become disoriented within the table. Therefore, you must make DTD customizations that prevent such complex tables.

Customize the DTD and the rendering tools as discussed in the “Graphics” and the “Tables” sections to streamline your process and prevent inaccessible outputs.

Understand That DITA 1.3 Provides More Table Accessibility Markup

The DITA 1.2 DTD and older DTDs are missing key elements and attributes that allow you to explicitly identify a table’s rowheaders and associates specific data elements within the table to those headers. DITA 1.3 adds these elements. Without these elements, authors are confined to only tables that read top down from the column headings or to tables where only the first cell in each row may be the heading for that row - such that all tables must be constructed in that way. With the new elements and attributes in DITA 1.3, writers can make any cell in a row the rowheader and still ensure a proper “read” of the table by screen readers.

DITA 1.3 adds a “headers” value to the table rowheader attribute that “tells” the rendering tool you are going to associate rows and columns with particular cells using the two new attributes on <entry> elements:

- scope = row, col, rowgroup, colgroup
- headers = uses the name of a header for an individual cell (<entry element)

With these new elements and attributes you should be able to make a somewhat complex table easily accessible. If the table is more complex than this, consider simplifying the table. For any table that seems overly complex, try to imagine whether you could navigate the table with your eyes closed having only the information in the table spoken to you.

Many Paths to Inaccessible HTML

Customizing your DTD and changing your rendering tools will go a long way to help you create accessible output efficiently. However, there are still many ways your content may be inaccessible. Developing (relatively easy) and using customized XML correctness checking tools will catch many errors during authoring where they can be fixed easily. The time spent to develop and run such tools is far better than focusing all of your accessibility checking on the HTML output.

For example, you are probably going to change your default table rendering to make the first cell in each row the rowheader. In that case, you must ensure that the first cell contains content. You will also want to make sure that the content is “real.” Some writers leave the first cell in a row blank by inserting a non-breaking space. If there is nothing in first cell and the table uses that cell as the rowheader, that table is NOT accessible to screen readers and would need a VPAT bug against the publication. Use the “XML Checking Tools” section to help your organization develop checking tools for your content.

Add Features and Pre-Test Template Output

Ensure that you are adding required/desired accessibility features into your outputs automatically. For example, many software programs state that their product contains a skip-headers mechanism that allows users to jump to the main page content without reading the page headers, tables of contents, or other repeated content every time. Ensure that you have built these features into the output and that you have carefully tested the template output for accessibility violations. If the template’s output fails accessibility tests, the authors work may be for naught.

Review the following standard areas of testing and more:

- Template-supplied tables in headers and footers
- Forms in the output headers and footers
- Template-supplied graphics
- Template-supplied links in the headers and footers
- Template color and contrast in text and images
- HTML hierarchical structure and id duplicates

rendering tool changes, and XML checking tools (before required HTML checking). The DTD customizations encourage correct use of elements and attributes to create the desired accessible HTML output, and XML “correctness” checker tools provide error-type fixes that authors must make before creating their HTML. This is not a validation tool - presumably your content is valid against the DTD and any customizations you have made. Instead, an XML “correctness” checker tool helps to ensure that you have content where it is required and that the result will be accessible.

XML “Correctness” Checker Tools

There are numerous ways to ensure and encourage accessible outputs during the authoring process. At Oracle, it is a combination of DITA DTD customizations,

Cause/Action

Use this Cause/Action section to help you design your own DTD customizations and/or XML correctness checker tools.

<image> may only have <longdescref> or <alt> child element but not both

Cause: An <image> element includes both a <longdescref> and an <alt> element.

Action: Remove <alt> (most likely) or <longdescref> but not both. Including both disables the alert to blind users that there is a link to a longer description. Use <longdesc> for most images. Use <alt> for icon-type simple graphics.

Most graphics are complex enough that they cannot be described briefly (usually 60 characters or less) using the <alt> element and they require a <longdescref>. <longdescref>, upon rendering, inserts an HTML alt attribute value (not to be confused with the DITA XML <alt> element) that alerts blind readers that a link to a description follows. If you also include a writer-supplied <alt> element, that alert is disabled and readers assume the alt attribute is complete and sufficient.

<image> must have <alt> or <longdescref> as a child element

Cause: An <image> is missing an <alt> or a <longdescref> element. To be accessible, an <image> must have one or the other.

Action: Insert <longdescref> or <alt>. Use <longdescref> for most images. Use <alt> for icon-type simple graphics. <alt> may only contain text and no other elements.

Note: <alt> may contain only the following inline elements in addition to text: keyword, apiname, option, parmname, cmdname, msgnum, varname, wintitle, term, ph, b, i, u, codeph, synph, filepath, msgph, uicontrol. <alt> may not contain the elements data, data-about, foreign, unknown, abbreviated-form, sup, sub, tt, systemoutput, userinput, and/or or menucascade because the intention of these elements cannot be conveyed in an HTML attribute. User these elements generate errors in this context.

<chhead> may not be used

Cause: <chhead> element is not allowed because it outputs a heading row in a layout table which is not accessible.

Action: Remove <chhead> or replace the parent <choicetable> element with <table> and <thead> or consider using the <steps-unordered> element.

<sthead> may not be used

Cause: <sthead> element is used but not allowed because it outputs a heading row in a layout table which is not accessible.

Action: Remove <sthead> or replace the parent <simpletable> element with <table> and <thead> or consider using the <steps-unordered> element.

<prophead> may not be used

Cause: <prophead> element is used but not allowed because it outputs a heading row in a layout table which is not accessible.

Action: Remove the <prophead> element or replace the parent <properties> element with <table> and <thead> or consider using the <steps-unordered> element.

<table> missing <desc> as child element

Cause: <table> element is missing a <desc> element that becomes an HTML table summary attribute value, which is required.

Action: Insert a <desc> child element and provide a meaningful description of the table content.

Note: <desc> may contain only the following inline elements in addition to text: keyword, apiname, option, parmname, cmdname, msgnum, varname, wintitle, term, ph, b, i, u, codeph, synph, filepath, msgph, uicontrol, q, term, state.

<desc> may not contain the elements data, data-about, foreign, unknown, abbreviated-form, sup, sub, tt, systemoutput, userinput, dl, parml, image, lines, lq, note, hazardstatement, ol, p, pre, codeblock, msgblock, screen, sl, ul, boolean, systemoutput, userinput, menucascade, tm, xref, state because the intention of these elements cannot be conveyed in an HTML attribute.

<table> must contain only one <tgroup> as a child element

Cause: <table> element contains more than one <tgroup> element

Action: Remove additional <tgroup> element or elements or move them into separate <table> elements. <table> elements with more than one <tgroup> cannot be made accessible.

<tgroup> missing <thead> element

Cause: <tgroup> element in <table> is missing a <thead> element.

Action: In <tgroup> in <table>, insert a <thead> element and required child <entry> elements to create a heading row. <table> element output data tables and data tables cannot be made accessible without a heading row.

Notes: The content of the cells in the heading row or rows (<thead>) is used by screen readers to orient users in the table.

More than one <row> is allowed in <thead> but keep tables simple.

No <entry> element may be blank or the equivalent of blank in <row> in <thead> because the screen reader will have nothing to read for cell orientation.

First <entry> element in <row> in <tbody> is blank or contain only a space or non-breaking space

Cause: The first <entry> element in a <row> in a <tbody> in a <table> is blank or the equivalent of blank.

The following shows the possible XML markup that causes this error.

```
<entry><p> </p></entry>
```

```
<entry><p></p></entry>
```

```
<entry><p>&#160;</p></entry>
```

```
<entry><uicontrol>&#160;</uicontrol></entry>
```

```
<entry></entry>
```

```
<entry><p ><uicontrol> </uicontrol></p></entry>
```

Action: Supply proper elements (<p>, etc. and text) in the first <entry> element of the <row>. Leaving the first <entry> in a <row> blank to imply ‘same value as the previous row’ is not permitted. Instead, repeat the value of the first <entry> in the previous <row> if this is the case. The first cell (<entry> element) in each row is key for screen readers to orient users in the table.

Figure 2 shows an example of this error in a table where the second body row is about “Assets” and the fourth body row is about “Enterprise Planning, etc. Visually, non-blind readers understand this but screen reader users have no way to know this because the first cell in the row, which is the “row heading” is missing.

Figure 3 shows the proper way to code the previous table so that the first cell (<entry> element) in every row has content that will be read by screen readers. It is OK and desirable to repeat information in cells.

<entry> element in a <row> that is in <thead> is blank or the equivalent of blank

Cause: <entry> elements in a <row> in a <thead> in a <table> are blank or the equivalent of blank. The resulting output is a table heading row with blank cells which is not accessible.

Action: Supply proper elements (<p>, etc.) in all <entry> elements in <row> elements in a <thead>. Leaving the <entry> element blank or the equivalent of blank means the screen reader has nothing to read when it attempts to navigate the table.

<entry> elements must not use the attribute morerows="some number"

Cause: Rows in a table have been spanned (vertically) causing this markup

Action: Remove (vertical) row spans which causes the morerows attribute. Use of the morerows attribute indicates the table is complex and will not render or read correctly by screen readers. Instead, simplify the table and remove row spans (morerows attribute). Horizontal spans are allowed in single rows.

Note: The morerows attribute usage will be restored late December 2016 approximately and the ACC check will be different.

<alt> contains disallowed elements

Cause: <alt> element contains disallowed elements

Action: Remove disallowed elements: data, data-about, foreign, unknown, abbreviated-form, sup, sub, tt, systemoutput, userinput, and/or or menucascade. Use only text plus inline elements: keyword, apiname, option, parmname, cmdname, msgnum, varname, wintitle, term, ph, b, i, u, codeph, synph, filepath, msgph, and/or uicontrol

Note: In HTML output, the <alt> element becomes an HTML attribute value - inline markup is allowed but is not expressed.

<desc> contains disallowed elements

Cause: <desc> element contains disallowed elements dl, parml, image, lines, lq, note, hazardstatement, ol, p, pre, boolean (deprecated), sup, sub, tt, systemoutput, userinput, menucascade, q, abbreviated-form, tm, xref, state, data, data-about, foreign, unknown

Action: Remove and/or unwrap disallowed child elements. <desc> may contain only text and inline elements: cite, keyword, apiname, option, parmname, cmdname, msgnum, varname, wintitle, ph, b, i, u, codeph, synph, filepath, msgph, uicontrol, term

Note: In HTML output, the <desc> element becomes an HTML attribute value - inline markup is allowed but is not expressed.

<row> element with only one <entry> element

Cause: All of the <entry> child elements in a <row> element are spanned horizontally.

Action: Remove a row span that spans an entire row horizontally causing only one cell in that row. This almost always is a method to divide the table into visual table units which is not accessible. Redesign your table or create multiple tables to avoid table units.

Figure 4 shows an example of all of the entry elements spanned in the first and fifth row. This cannot be made accessible using DITA 1.2 or previous.

Figure 5 shows an accessible alternative way to handle the previous table by adding a row that provided the labels for the Command Types: Element Commands, Format Commands.

Product	Job or Abstract Role	Security Profile
Assets	Asset Accountant	View All Workers
Assets	← Accounting Manager	View All Workers
Enterprise Planning	Budget Analyst	View All Workers
Enterprise Planning	← Budget Manager	View All Workers
Financial Consolidation	Consolidation Accountant	View All Workers
Financial Consolidation	← Consolidation Manager	View All Workers
Financials Common Module	Intercompany Accountant	View All Workers
General Ledger	Financial Analyst	View All Workers
General Ledger	← General Accountant	View All Workers

Figure 2. Example of an error in a table where the second body row is about “Assets” and the fourth body row is about “Enterprise Planning, etc.

Product	Job or Abstract Role	Security Profile
Assets	Asset Accountant	View All Workers
	Accounting Manager	View All Workers
Enterprise Planning	Budget Analyst	View All Workers
	Budget Manager	View All Workers
Financial Consolidation	Consolidation Accountant	View All Workers
	Consolidation Manager	View All Workers
Financials Common Module	Intercompany Accountant	View All Workers
General Ledger	Financial Analyst	View All Workers
	General Accountant	View All Workers

Figure 3. The proper way to code the previous table so that the first cell (<entry> element) in every row has content that will be read by screen readers.

<alt> is blank or the equivalent of blank

Cause: <alt> element has no content, has a non-breaking space, or contains inline elements that then have no content.

Command Name	Shortcut	Behavior
Element Commands		
Split	Esc E + s	Splits an element, creating a second element of the same type
Merge	Esc E + m	Merges two elements of the same type.
Wrap	Esc E + w	Wraps the current selected elements or text.
Format Commands		
Font	Control f + t	Change the typeface
Style	Control s + t	Changes the format. Bold or italic, for example.
Size	Control s + z	Changes the typeface size.

Figure 4. An example of all of the entry elements spanned in the first and fifth row. This cannot be made accessible using DITA 1.2 or previous.

Command Type	Command Name	Shortcut	Behavior
Element	Split	Esc E + s	Splits an element, creating a second element of the same type
Element	Merge	Esc E + m	Merges two elements of the same type.
Element	Wrap	Esc E + w	Wraps the current selected elements or text.
Format	Font	Control f + t	Change the typeface
Format	Style	Control s + t	Changes the format. Bold or italic, for example.
Format	Size	Control s + z	Changes the typeface size.

Figure 5. An accessible alternative way to handle the previous table by adding a row that provided the labels for the Command Types: Element Commands, Format Commands.

Action: Supply a very brief description as text only or using inline elements: keyword, apiname, option, parmname, cmdname, msgnum, varname, wintitle, term, ph, b, i, u, codeph, synph, filepath, msgph, and/or uicontrol where the inline elements include content.

<desc> is blank or the equivalent of blank

Cause: <desc> element has no content, has a non-breaking space, or contains inline elements that then have no content.

Action: Supply a brief table description as text only or using inline elements: cite, keyword, apiname, option, parmname, cmdname, msgnum, varname, wintitle, ph, b, i, u, codeph, synph, filepath, msgph, uicontrol, term where the inline elements include content.

Conclusion

You can automate the process of making your DITA output accessible by:

- Customizing the DTD
- Customizing your rendering tools
- Developing and using XML correctness checker tools
- Moving to DITA 1.3

Principal Program Manager, Edna was the Director of the Oracle DocTools group. Edna has worked on a variety of authoring and related tools including Arbortext Epic, FrameMaker, SDL, Dreamweaver, and proprietary authoring and rendering tools. She has designed the architecture for numerous conversions that include XML-to-XML, HTML-to-XML, and XML-to-MIF. Since 2007, Edna has handled the many large-scale conversions for the Oracle acquisitions of Sun Microsystems, BEA, Siebel, Stellent, and others. Edna holds a patent for accessible definition links in online documentation (US7493560).

Resources

2017 WebAIM: Center for Persons with Disabilities - Utah State University,

“Creating Accessible Tables,” <http://webaim.org/techniques/tables/#uses>

Nielsen Norman Group, “Keyboard-Only Navigation for Improved Accessibility,” April 6, 2014,

<https://www.nngroup.com/articles/keyboard-accessibility/>

Schengili-Roberts, Keith, “DITA 1.3 and Improved Table Accessibility for Screen Readers.”

<http://www.ditawriter.com/dita-1-3-and-improved-table-accessibility-for-screen-readers/>

University of Minnesota Disability Resource Center, “Data vs. Layout Tables,”

<http://accessibility.umn.edu/tutorials/data-vs-layout-tables>

Author Contact Information

Edna Elle
Senior Principal Program Manager
Oracle
One Oracle Drive
Nashua, NH 03062
edna.elle@oracle.com
603.897.3060

Author Biography

Edna Elle is a Senior Principal Program Manager for documentation tools at Oracle with 20 years experience in authoring tools, HTML accessibility features for blind and low-vision readers, and keyboard users. Edna also works on XML translation and translation workflow. Before she became a Senior



Gardner, Michelle M.

Walking in Your Users' Shoes: Create Journey Maps for an Improved User Experience

Journey maps help us clarify the needs of users and identify their pain points. Development teams use this information to ensure the success of new features and make changes to existing ones. Writers can learn to use journey maps to create targeted content. With this knowledge, writers can also push for improvements to the user interface to reduce the need for documentation. In a perfect world, a product manager or user experience expert might run the journey mapping process. However, most of us work for organizations that do not have UX or journey mapping experts. That's an opportunity for writers to take the lead in improving the user experience. You can become an advocate for the user. When you create a journey map, you're actually walking in a user's shoes.

A journey map helps us understand the user experience for a single process or a very complex one comprising multiple processes. At each stage of the user's journey, we must understand the user's needs, goals, and emotions. We also identify the key moments where the user interacts with our product or company, and makes decisions based on those interactions.

You can group user activities by the solutions that they provide. You can then organize the overall process, plan for future solutions or enhancements, and set priorities for competing deliverables. You can also use the journey map to organize content within the user interface because it helps you identify the information and content that the user needs at each stage of the journey. The journey map can identify the user's pain points. It also helps you discover what you might have missed or where you have added too much to the process or user interface.

Some UX experts have likened a journey map to a good story because they both have rising and falling action, a climax, and a denouement (Lichlaw, 2016). For example, Figure 1 shows the process for purchasing a cup of coffee (Journey, 2013). Conflict occurs at different points in the journey, such as deciding whether to walk or drive and which option to choose from the menu. But it all leads to the user's receiving

the coffee and taking that first, satisfying sip. In the denouement, the user drinks the rest of the cup in a good frame of mind.

The journey, like a good story, has an end goal: The user wants to consume a cup of coffee. We hope that the first sip is satisfying because the purpose of our product development teams is to ensure that the experience satisfies the user. (You can replace satisfies with delights, encourages, or the adverb that your product brand promises.) Journey maps give us an opportunity to directly affect that adverb by identifying the areas where we can improve the user's experience.

Understanding Journey Maps

Each journey map is a story told from a particular persona's point of view about the activities that persona takes to accomplish a specific goal.

A Story Told from a Specific POV

You cannot create an effective journey map without a user persona. The user persona represents a virtual user, complete with name, picture, and background story. The persona serves as an archetype, based on

the profile data. It describes that user’s skills, needs, behaviors, goals, motives, and interactions with other user personas. It explains why the user does what he does and why she chooses how to do it a certain way.

Because the user persona serves as the “voice of the user” (Ilama, 2015), you can treat that persona as a human when interacting with your development team. Tell stories about what the persona does and why. The more you tell stories and talk about the persona, the more your team can empathize with the users that persona represents. And the only way to create a good journey map is to put yourself in that user’s shoes.

For example, at the coffee shop you might see a group of friends chatting in the line; they don’t notice that the line is progressing slowly. They’re invested in the coffee shop experience. However, the individual behind them just wants to get a cup of coffee to consume on the way to work. These are different personas with the same apparent goal: a cup of coffee. The very impatient, hurried customer has a different need compared to the one who enjoys chatting with friends while in line and while sitting in at the tables. If we fail to tell their individual stories as individual journeys, then we fail to design a good user experience for at least one of them.

About the Activities the Persona Takes

To create a journey map, select the feature, task, or solution that your product provides for a particular persona. Then walk through the activity stage by stage.

The example about buying a cup of coffee also identifies several stages of that persona’s journey: standing in line, choosing method of payment, ordering. When you create a journey map, you need to understand each of the stages and the choices that the user makes during that stage. It’s also possible that a stage might cause the user to take different courses of action. The user’s thoughts and feelings affect the choices that the user makes at each stage. These moments, or touch points, affect the user experience during the journey. For example, an impatient person might not want to stay in a slow line. Instead, that customer will choose an alternative way to get coffee. And your coffee shop might have lost a customer because of that stage in the journey.

The journey map would help you identify this pain point for that persona. Perhaps your coffee shop would create a self-service bar that allows users to pour a cup, scan their credit card, and then be on their way.

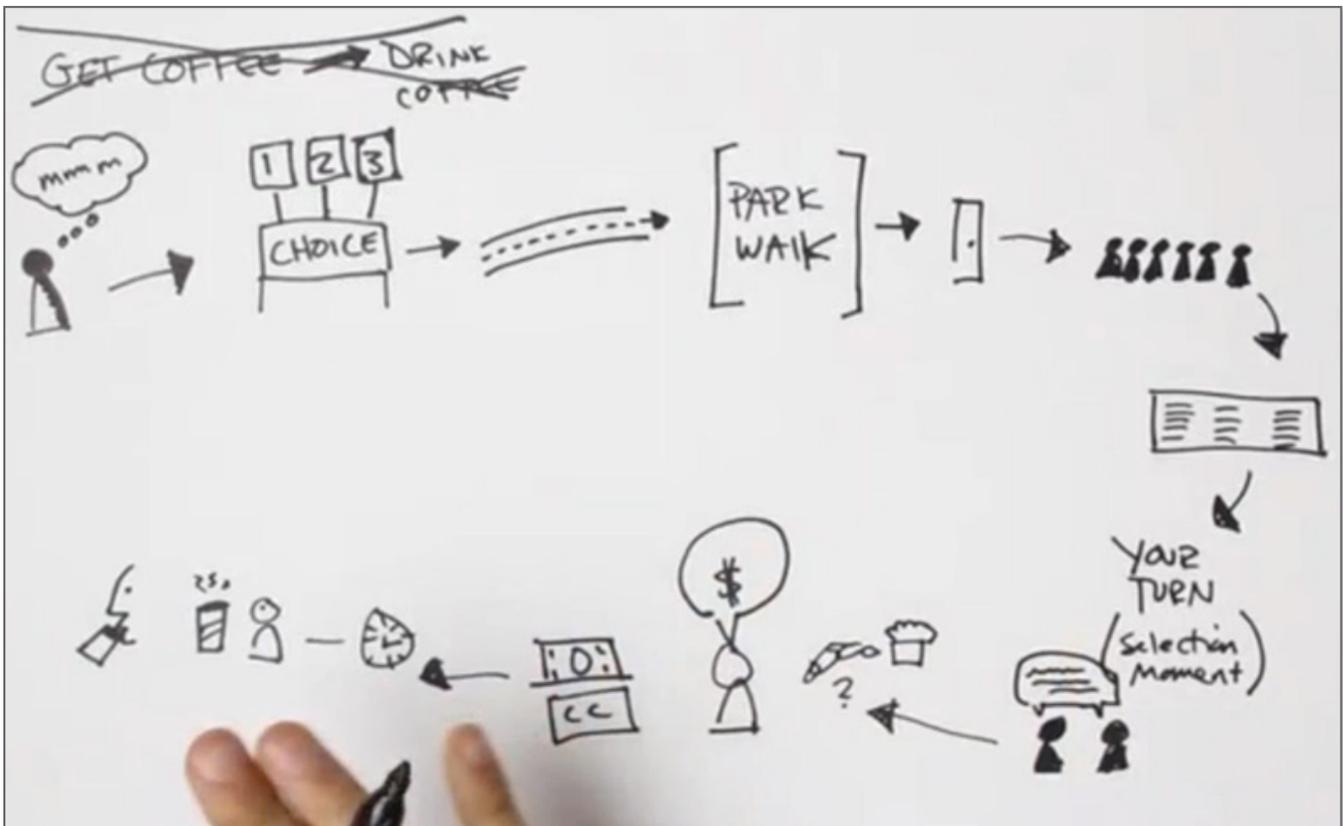


Figure 1. Sample Journey Map

It's a no-frills solution, but one that might suit the hurried persona.

To Accomplish a Specific Goal

When you design the user persona you identify that persona's needs and goals. Products should help users meet those needs and goals or solve problems related to them. As you step into your user's shoes and follow him through the stages of the journey map, you must keep that end goal in mind. At each stage, you should know the user's expectations for the outcome of that state. Then you should ask whether that stage could be avoided, improved, or moved to a different point in the journey. Would the change make it faster or simpler for the user to reach her goal in a satisfying way?

For example, for the hurried persona, you remove the need to stand in the line designed for leisurely customers.

Creating a Journey Map

Once you have the persona, the stages in the journey, and the goal or solution that your product provides, you can create the journey map. There are professional software applications designed to help you build the map. However, you can use also Post-it notes, a white board, Microsoft Excel, or PowerPoint. Basically, use any method that enables you to describe and share the user's journey to others. The method isn't as important as the content.

Elements of a Journey Map

Recommendations from UX experts vary regarding the terminology and content in their journey maps.

For example, you can apply Megan Grocki's (2014) must-have and like-to-have elements:

- Personas representing the main characters that illustrate the needs, goals, thoughts, feelings, opinions, expectations, pain points
- Timeline that identifies a set amount of time or time for each phase of the process
- Emotion that identifies the peaks and valleys of frustration, anxiety, happiness

- Touchpoints that identify customer interactions with the organization; this represents the WHAT the user is doing
- Channels where customer interactions take place; this represents the WHERE the customer is interacting
- Moments of Truth that identify positive interactions that leave a lasting impression; perhaps add at a touchpoint known to generate anxiety or frustration (like-to-have)
- Supporting characters who might contribute to the experience (like-to-have)

Kate Kaplan with the Nielsen Norman Group (2016) describes the journey map in terms of zones instead of stages. However, the content in her maps still address persona, emotion, actions, and the scenario you want to examine. She also points out that you need more than the right elements in your journey map.

You need to be aware of the activities that make the journey mapping process better:

- Establish the "why" and the "what" that represent the user's reasoning and end goal
- Base the journey map on truth to avoid assumptions and inaccurate stories; include research results and real narratives from users
- Collaborate with others because the process of creating the journey map causes questions and discussions among your team that help you clarify user needs
- Don't jump to visualization before you have a well-understood set of data for the journey map
- Engage others with the end product to ensure buy-in from your product team, stakeholders, and management

Applying the Elements to a Journey Map

Select the persona and timeline for the journey map, then start adding the stages. Most journey maps resemble a table where the stages are laid out as a series of columns progressing from left to right. For the coffee example, you might insert columns like "Stage 1: Decide to get coffee" through "Stage 12: Take a sip of coffee."

The table rows represent the persona's timeline, emotions, touchpoints, and channels at each stage.

You should include a row for identifying the outcome or result that the user expects for each stage. You might also want to include a row where you note whether a stage can be skipped, when and why. For example, skip the long line in favor of self-service. Finally, you could have a row to list opportunities for improvement, such as creating the self-service coffee bar within your full-service coffee shop.

When to Create Journey Maps

You can create a journey map at any time. In a best case scenario, your entire development team should participate in the process. Doing so allows the team to not only improve the design through questions and clarifications, but also have the same view of the mapped process, which can reduce issues when coding begins. If the whole team is involved, you should create the map at the planning stage of the project.

However, you can also create the map during the design phase to ensure you have the right elements in the user interface at the right time. If your development team is not ready for journey mapping, you can create a journey map on your own at any point during the release. After a release, you can use the map to track whether the features the team created actually match what the user needs to do.

From Journey Map to Targeted Content

The journey map clearly outlines the process a person takes to resolve a need or goal by using your product. It might also tell you what the user does before and after using your product to provide a coherent process. You can transform this information into targeted content. By knowing what the user is thinking, needs, and expects at each stage in the journey, you can identify the content to provide for that procedure or concept. You can also apply that data to improving the user interface.

For example, if your coffee shop knows that you have impatient customers wanting a quick cup of coffee, you could create a sign at the door that directs these customer to the self-service coffee bar. In your user guide or online help, you might have separate sections of text for the different personas. Each section outlines the steps associated with the journey for that persona.

“Come in and pour your own coffee” versus “Have a great time chatting with friends while you wait for coffee created just for you.”

Regardless of the content you develop, you can use the journey map as your guide to understanding exactly what you should say to that user at the right time to improve the journey.

Walk in Your User’s Shoes

If you can walk in your users’ shoes, then you can follow their steps as they use your product. Those steps infuse the journey map. Which in turn helps you identify the touchpoints in the process that need improvement. If the journey map can be likened to a good story with action, climax, and resolution, then you can become the author of that story – both in building the journey map and creating the user content. By doing so you can improve that user’s experience.

Resources

Anderson, Stephen P. *Seductive Interaction Design: Creating Playful, Fun, and Effective User Experiences (Voices That Matter)*. (New Riders) 2011. <https://www.youtube.com/watch?v=U-SNVLGBh5o>

Garret, Jesse James. *The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition) (Voices That Matter)*. (New Riders) 2010. <http://www.jjg.net/elements/>

Introduction to UX (October 2015). EffectiveUI. Retrieved from SlideShare: http://www.slideshare.net/effectiveui/introduction-to-ux?qid=4b66b853-9a9e-48fc-9740-a4ec76c8a971&v=&b=&from_search=9

O’Conner, Kevin. “Personas: The Foundation of a Great User Experience.” *UX Magazine* 640 (March 2011). <https://uxmag.com/articles/personas-the-foundation-of-a-great-user-experience>

Toporek, Adam. “7 Tips to Get Started with Customer Journey Mapping.” *Convince&Convert* (28 April 2015). <http://www.convinceandconvert.com/community-management/customer-journey-mapping/>

Schlomo Golz. “A Closer Look At Personas: A Guide to Developing The Right Ones (Part 2).” *Smashing Magazine* (13 August

2014). <https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-2>

User Persona Creator. (n.d.) Extensio: <https://xtensio.com/user-persona/>

References

- Grocki, Megan. "How to Create a Customer Journey Map." UX Mastery (16 September 2014). <http://uxmastery.com/how-to-create-a-customer-journey-map/>
- Ilama, Eeva. "Creating Personas." UX Booth (9 June 2015). <http://www.uxbooth.com/articles/creating-personas/>
- Journey Map (4 November 2013). Stanford d.school: <https://vimeo.com/78554759>
- Kaplan, Kate. "When and How to Create Customer Journey Maps." NN/g Nielsen Norman Group (31 July 2016) <https://www.nngroup.com/articles/customer-journey-mapping/>
- Lichaw, Donna. *The User's Journey: Storymapping Products That People Love*. (Rosenfeld Media) 2016.

Author Contact Information

Michelle M. Gardner
Sr. Information Developer
Micro Focus
515 Post Oak Blvd, Suite 1200
Houston, TX 77027
713.418.5180

Author Biography

Michelle M. Gardner is passionate about improving user experience and providing excellent user-focused content. She firmly believes that everyone on the development team should be a user advocate. In the last 25 years, she has written product documentation, designed or sold medical products, and managed tradeshows. She started as a technical writer at the University of Houston then moved to Sulzer Intermedics. At Intermedics, she segued into product management after becoming more involved in the product development process than a technical writer usually would. As a marketing and product manager for Intermedics, S&S Technology, and NeuroTherm, she solicited feedback from customers to develop new products. She even tried her hand at selling medication

dispensing systems for Artromick. Throughout her career, though, Michelle has most enjoyed talking to people about what they need and designing solutions to meet those needs.

In 2008, Michelle found her way back to technical writing, but with a twist: As a Sr. Information Developer at Micro Focus, she applies her marketing experience to develop a targeted content strategy that aligns with the product's user profiles. She works with development teams to apply UX principles to new product features. She also mentors colleagues for DIY usability testing. In 2016, she became the only non-software engineer on a corporate team that focuses on improving product innovation. Most recently, she joined a usability design council, which reviews UIs and makes UX recommendations for Micro Focus security products.

Michelle also is a founding member of the Houston chapter for User Experience Professionals Association.



Guren, Leah

Six Super Success Hacks

You can begin to take control of your career by making minor changes to your daily behavior. These six small changes, or “hacks”, are valuable whether you have been in TC for one year or twenty!

Why a Hack?

When most people thinking about making a positive change in their lives, they think in terms of goals:

- “I want to speak Spanish.”
- “I want to run a marathon.”
- “I want to lose 30 pounds.”
- “I want a PhD.”

The problem is that all of these goal-oriented thoughts, while nice for daydreams, don’t get you any closer to what you want. In fact, they can seem dauntingly impossible and even make you feel powerless.

Instead, changing daily behavior is an effective way to create change in our lives. For example, if you want to learn another language, study for ten minutes every day rather than focus on the seemingly impossible end-goal of fluency. Want to run that marathon? Work a training program into your daily routine.

Hacks are all about minor tweaks that are realistic. You don’t set yourself up for failure when you make a small, comfortable change that feels manageable. Through hacks, we can make subtle yet significant changes to our lives and our careers.

1. Work Your Calendar

Forgetting appointments and missing deadlines makes you appear flaky and unprofessional. Once clients consider you unreliable, it is hard to re-earn their good opinion. Your calendar is the most powerful tool for reminding yourself of commitments and

responsibilities. While there are many productivity tools out there for tracking tasks, nothing beats a simple built-in calendar, whether you use Google, Outlook, or something else.

In your calendar:

- Mark deadlines for client tasks on your calendar.
- Mark a buffer (that is, a “heads-up” date) for these tasks. After all, it doesn’t help very much if you don’t see that a paper is due until the actual date it is due!
- Mark meetings, even in-house sit-downs with SMEs (subject-matter experts), PMs (product managers), etc.
- If travel is involved to get to a meeting, mark a buffer (departure day or time).
- Every night before you go to bed, look at your calendar for the following day.
- The evening before the start of your work week, review your calendar for week.
- Never make a commitment without first checking your calendar.

2. Stop Lying to Yourself

We’ve all been guilty of telling ourselves lies. “I’ll write that article tonight after work,” or “I’ll just play one more game of Candy Crush.” Each time we lie to ourselves, we lose a bit more internal credibility. Our self confidence erodes further.

You must know yourself before you can accurately plan, schedule, and manage a workload. It isn’t a judgement

on your character; it is about organizing your work in a way that suits you. Not a morning person? Then it is not sensible to tell yourself that you will get up at 05:00 to finish proofreading something. Low energy at night? Then don't expect that you will be able to accomplish demanding exercise or chores.

Instead:

- Be honest about what you like and don't like.
- Look at your past track record; for example, how long did a similar task take last time?

3. Lobby for the User

A common reason that people are unhappy in their jobs is that they feel powerless. Despite their education, knowledge, and skills, they feel unable to accomplish anything worthwhile. When TCs get downgraded to a mere mouthpiece of the PM, they know that only a fraction of their skills are being used. It can suck all the joy out of work.

Becoming an advocate for the user is a great way to inject our knowledge and expertise into the equation. Often, clients have expectations because they do not understand what we can do and how helping the user actually benefits the company. By flipping things and looking at it from the user's perspective, we can reconnect with the positive human side of our work.

- Look at every new feature, new document, or new project from the standpoint of user value.
- Find ways to articulate how your work helps people.

4. Communicate Like a Pro

There is no excuse for a profession TC to make sloppy and careless mistakes in anything they write, whether an email, a social media post, or an in-house report. We live and die by our ability to communicate clearly and accurately. This must be reflected in our speech, as well.

- Before any meeting, understand the issue, prepare your own material, and be organized.
- Be able to articulate your opinion in clear, concise language. Don't waffle or dither.

- Not a strong public speaker? Take a class or volunteer to speak at your local chapter.

5. Find the Spotlight

Tech jobs tend to be male dominated and very much ego driven. But our profession is predominantly female and overwhelmingly introverted. It is common to see intelligent, capable TCs ignored or talked-over in meetings. While it may be uncomfortable for a shy person to draw attention to their achievements, it is necessary lest others take credit for the work. The very act of stating something can cause an association in a manager's mind between the idea and the person. This is sometimes referred to as the Echo Effect, in which a higher power individual repeats what a lower power person said, thus implicitly taking credit for the idea.

- Speak up in meetings. Don't let people talk over you.
- It isn't bragging or egotistical to "own" your work. We admit our mistakes; let's also admit our achievements.

6. Fake It till You Make It

There is real data behind this old adage. It turns out that that by taking action, even if you are unsure, you can build confidence. Amy Cuddy's TED talk, "Your body language shapes who you are," explores how changing posture not only changes how people perceive us, but creates chemical changes in our bodies. In other words, we even influence ourselves by our own nonverbal communication. So while many of us are dealing with Imposter Syndrome, we can overcome it!

- Remember about communicating like a pro and finding the spotlight? Even if it is difficult at first, by doing it, you get better and more confident at it.
- Try the Victory pose and the Superman pose. Try the forced smile. Two minutes can change your mood and your confidence!

Conclusion

These particular hacks seem to resonate with many TCs. I hope that you find them useful. I would also love

to hear about your own life hacks that have worked for you!

References

- Bonsignore, Alisa. "Introverted Entrepreneurship: Embracing Your Introvert Skills." Intercom, March 2017. <https://www.stc.org/intercom/2017/03/introverted-entrepreneurship-embracing-your-introvert-skills/>
- Cuddy, Amy. "Your body language shapes who you are." TEDGlobal 2012, June 2012. https://www.ted.com/talks/amy_cuddy_your_body_language_shapes_who_you_are.
- Warrell, Margie. "Afraid of Being 'Found Out?' How to Overcome Imposter Syndrome." Forbes, April 3, 2014. <https://www.forbes.com/sites/margiewarrell/2014/04/03/impostor-syndrome/#483d213c48a9>
- Whitlow, Stephanie. "How to Spot an Extroverted Technical Writer and Other Mythological Creatures." Intercom, March 2017. <https://www.stc.org/intercom/2017/03/how-to-spot-an-extroverted-technical-writer-and-other-mythological-creatures/>
- Woelke, Ben, and Morgan, Hannah. Empowering the Introvert Within; Becoming an Outstanding leader (Proceedings, STC Summit Conference, 2013).

Author Contact Information

Leah Guren, Fellow
Owner/Operator
Cow TC
leah@cowtc.com
+972.544.853473

Author Biography

Leah Guren is the owner/operator of Cow TC. She has been active in the field of technical communication since 1980. Now, as a trainer and consultant with some of the top high-tech companies internationally, she is known for her ability to bring humor and life to any topic. Leah is a published author in TC and a Fellow in STC.



Harvey, Michael

Documentation Support for an Internet of Things (IoT) Product: A Case Study

The Internet of Things (IoT) consists of devices embedded with sensors and transmitters that stream data over the Internet and the systems that process streaming data. I provide documentation support for SAS Event Stream Processing, my company's flagship IoT product. In this case study, I describe how I used common technical communication techniques to understand and support this bleeding edge technology.

Anyone who works in information technology has likely heard the buzz about the Internet of Things (IoT). What is it? How does it affect Internet users and technical communicators?

Kevin Ashton coined the term “Internet of Things” during a presentation in 1999. Ashton was a founder of the Auto-ID center, who performed pioneering research in the field of networked RFID and sensing technologies. RFID uses electromagnetic fields to tag objects. If you have a pet, you are likely familiar with the RFID chip that veterinarians embed under the skin for identification. Ashton and his colleagues were the first to link RFID-emitting objects through the Internet.

As the Internet became more pervasive and sensing technologies became more sophisticated, technology companies capitalized on Ashton's pioneering work. These days, the Internet of Things is exploding. See Table 1.

I learned about the Internet of Things providing documentation support for SAS Event Stream Processing. That product is a set of programming tools to build applications that process and perform real-time analytics on streaming data.

When I was first assigned SAS Event Stream Processing, I understood nothing about it. I inherited two technical documents: an Overview and a User's Guide. Immediately I was confronted with a variety of puzzling technical terms. The Overview presented

a “continuous query” with code. In the User's Guide, nearly every heading was a head 1, which implied that every topic was equally important. Chapters were unnumbered, which implied that their order did not matter. There were pages and pages of C++ code. How is someone supposed to understand anything by perusing C++ programs? The organization of the document was problematic and there was questionable repetition. I could barely understand the concepts covered, much less what someone actually did with the product.

I spent weeks reading, rereading, and annotating the User's Guide. I demystified those puzzling technical terms by redefining them with words that I understood. In doing so, I discovered that often a single term was used to mean multiple things. In one chapter “model” referred to a graph that represented the flow of events. In another, “model” referred to a nested set of containers, the bottommost of which contained the graph. So which was it? Sometimes the graph was called a “query” (not “continuous query” – was there a difference?) and other times it was called a “directed graph.” As I reviewed the material, I rewrote and edited it. I removed some terms and edited others for clarity. As I progressed, I met with developers to test my assumptions.

I looked for illustrations that could help me better conceptualize what was going on. See Figure 1.:

	Oil and Gas	Municipality	Truck Manufacturer
Data Source	Sensors on oil wells	Sensors on water meters	Sensors on truck engines
Activities monitored	Oil extraction rates, temperature, well pressure	Water usage, pressure	Engine diagnostic codes, fuel efficiency, braking patterns, routes, idle time
Purpose	Preventative maintenance	Billing, maintenance	Preventative maintenance, logistics
Frequency of readings	N x day x activity	N x hour x day	N x truck x day

Table 1. IoT use cases

The text of the User’s Guide spoke of the relationship between source windows and derived windows, but this diagram did not show “derived windows.” The text explained that engines contained projects, which contained continuous queries, but it also spoke of connectors and adapters. They did not appear in this diagram. I struggled to put the pieces together.

I also struggled to define the audience. Was it a programmer who builds event stream processing applications? Or someone who uses event stream processing applications to analyze event streams? It turns out that the answer was both.

I was able to apply rules that I had learned early in my career about how to write for programmers. Never elaborate or repeat what programmers already know.

Build on what programmers do know, and use terms consistently. Create a logical flow of topics and stick to it. Remember that programmers value easily accessible reference material.

Working on these documents proved to be an example of the cliché “building the plane during takeoff.” As I was rewriting the old documentation, the developers were giving me copious new source material. Every day, I would write, revise, and repeat. I found myself correcting technical errors in a C++ reference manual while handling new material about an expanded XML modeling layer. I had to learn how “connectors” and “adapters” worked while creating material about half a dozen new connectors and adapters. And I had to document a completely new user interface.

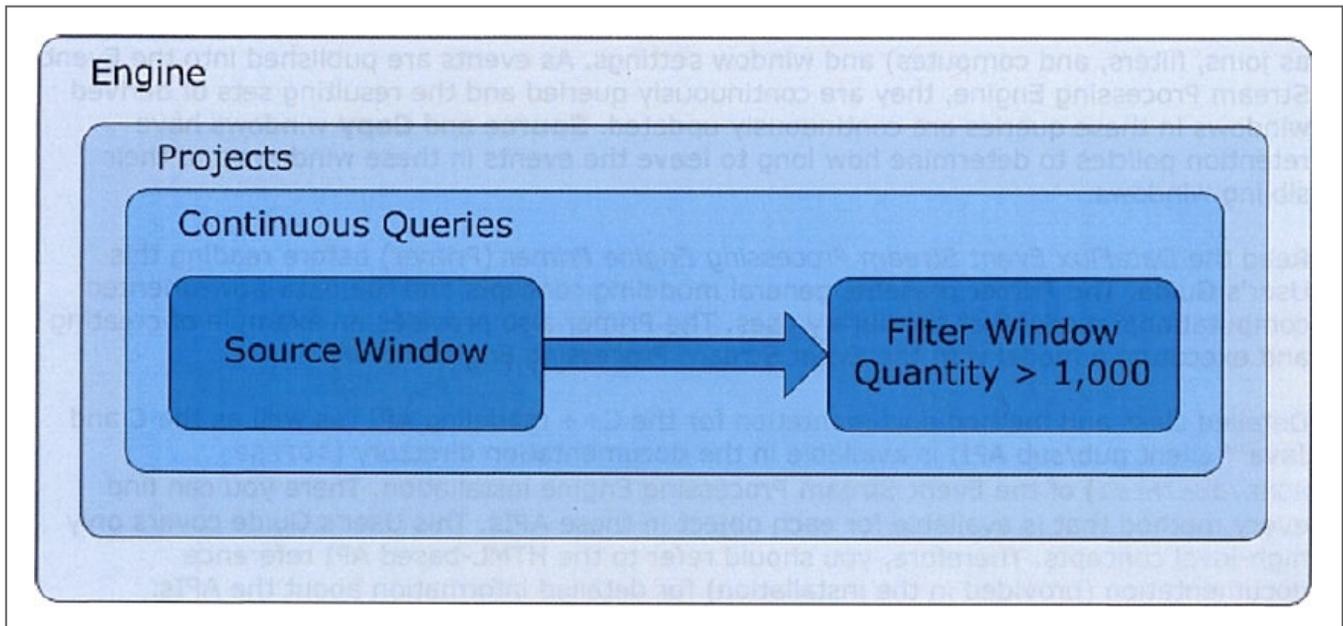


Figure 1. Example event stream processing application

Insert (i)					
Update(u)	Normal (n)				
Delete(d)	Partial update (p)	ID*	Symbol	Quantity	Price
	Retention				
Upsert(p)	generated (r)	int32	string	int32	double
[i,	n,	10,	IBM,	2000,	164.1]

To maintain this respect, I needed to learn a lot of technical detail. I learned how to read C++ code even though I could not write it. I became familiar enough with technologies such as YARN and Apache Camel that I could appropriately write about them in the context of the product. And every step of the way, I continued to ask questions, but only after I had done everything that I could to find the answers on my own.

Figure 2. Sample event that captures a stock transaction.

To do this job effectively, I needed to build trust with development. They were very competent software engineers, not technical communicators. The first thing that I did when I met with them was to communicate our different roles. They knew the product inside and out. My job was to overcome their “curse of knowledge.” That is, you unknowingly assume that the person that you are talking with has the background to understand what you are saying. This leads you to inadvertently leave stuff out, or take explanatory shortcuts. As technical communicators, we have to make sure there are no gaps in the explanatory or reference text that we create. I scheduled as much face time as I could with development, paraphrasing what I heard, continuing to test assumptions and ask questions. After they understood what I was doing to overcome the curse of knowledge, we developed very efficient lines of communication. And we developed mutual respect.

I knew that in order to write useful documentation for our customers, I needed to tell a coherent story. What were the important things to know before you started using SAS Event Stream Processing? And what were the important things that you did with this product?

After all of that work, a clear story emerged. Streaming data is ephemeral, unlike static data. Streaming data is structured in “events.” An “event” consists of metadata and field data. Metadata consists of an opcode and a flag. Field data conform to a specified schema. Events flow through SAS Event Stream Processing. See Figure 2.

The elements shaded in yellow are the metadata, and the elements shaded in green are the field data. The first two metadata elements denote the opcode (insert) and the flag (normal). The field data are an ID (10), which is a key element (denoted by *), a symbol

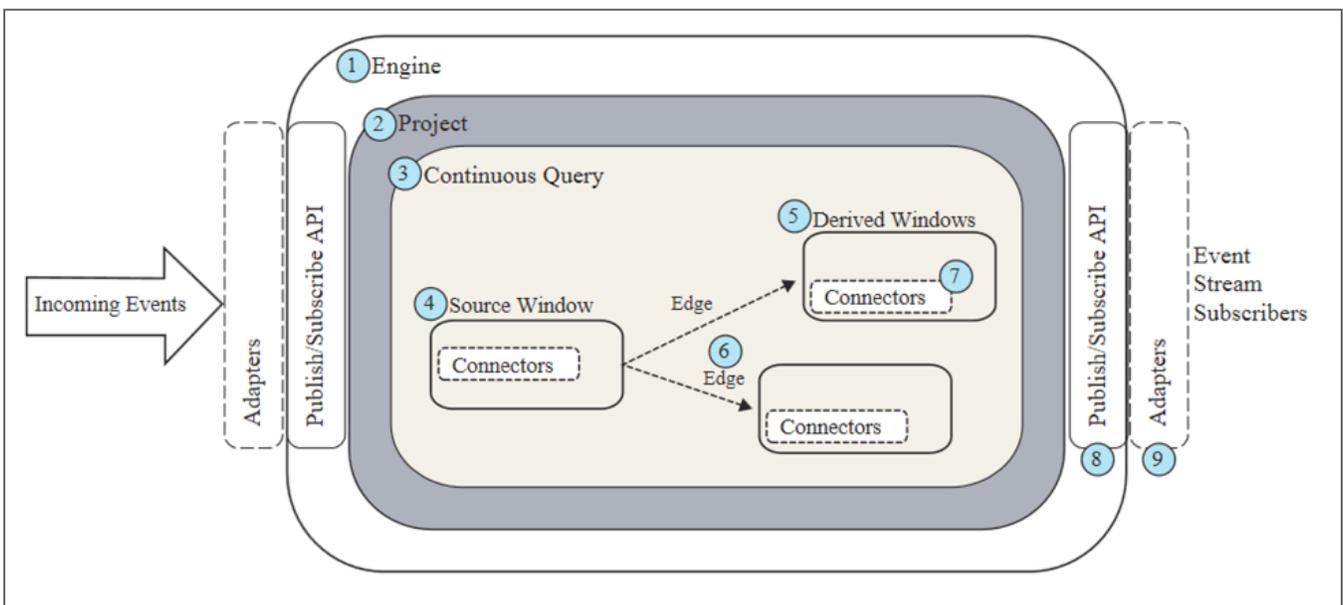


Figure 3. SAS Event Stream Processing Model Hierarchy

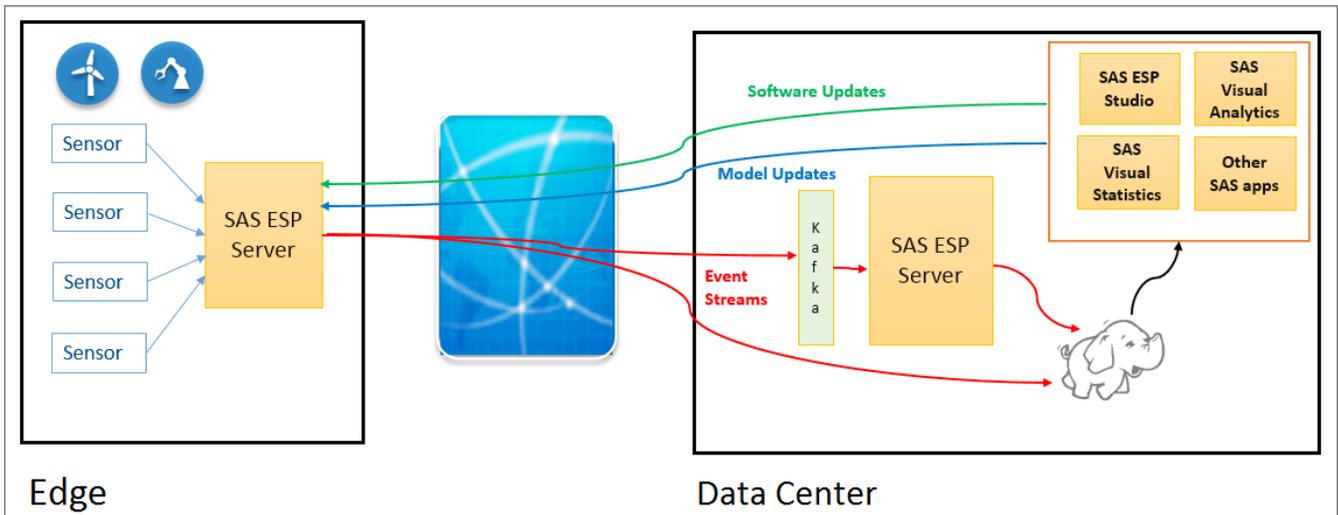


Figure 4. SAS Event Stream Processing used in an IoT context.

(IBM), a quantity (2000) that denotes the number of shares to trade, and a trade price (\$164.10) for those shares.

How do events flow through SAS Event Stream Processing? See Figure 3.

This direct descendant from that original illustration provides more detail about the relationship between product components, and shows how events flow through an event stream processing application. It includes callouts that are explained in text that I wrote for the revised User's Guide.

Recently, SAS Event Stream Processing has been offered as the keystone to our company's IoT offerings. See Figure 4.

An "edge" version of SAS Event Stream Processing runs on a system near the location of transmitting sensors. That system runs models customized to process incoming events from that specific device. Processed events are streamed through the Internet to another system in a data center, which runs a full version of the product. There, events are further evaluated and offloaded to Hadoop storage. The resulting static data is analyzed on a server that runs a variety of SAS products. Programmers use a model building program, SAS Event Stream Processing Studio, to refine models based on the analysis and push them back to the edge.

Recently, our CTO participated in a panel discussion about what is ahead for IoT. Presently, the verticals that use IoT include wind turbines, precision agriculture, trucking, construction rental, airlines, but that list is likely to grow dramatically in the next five

years. How? Anyone who builds a machine needs to start thinking of themselves as information vendors. If their revenue is 100% product, they are missing out on service opportunities through IoT. Another growth area for IoT is health care, because remote monitoring services are becoming more sophisticated.

IoT end users do not need or read documentation. They consume the analysis of event streams, but they do not need documentation to help them. What does this mean for us, as technical communicators? As my story shows, we are needed to overcome the curse of knowledge of IoT product developers. Clear and usable programming manuals, deployment guides, and architecture documents are going to be required as the IoT grows.

My personal challenges moving forward include connecting the dots for my company's IoT customers – offerings that bundle SAS Event Stream Processing with various SAS products. And I expect to continue to build more planes during takeoff.

Author Contact Information

Michael Harvey, Fellow
Principal Technical Writer
SAS
100 SAS Campus Drive
Cary, NC 27513-2414
919.531.1815

Author Biography

Michael Harvey serves as documentation project leader for risk management solutions and SAS Event Stream Processing at SAS. Previously, Michael worked as a manager and a writer for EMC. He has a BA in English and Psychology from the University of North Carolina at Chapel Hill and an MA in Experimental Psychology from Duke University. Michael has served in various leadership positions for the Carolina chapter of the STC and has presented at local and international STC conferences. He was honored to be named an STC Fellow in 2011. As an instructor for the Durham Technical Community College Technical Writing program in the late 1980s and early 1990s, Michael worked to overhaul the curriculum, emphasizing the importance of developing technical curiosity and acquiring technical expertise. See his professional profile at LinkedIn.



Jones, Adam

Avoiding the \$36 Comma: Clever Editing Strategies Can Reduce Translation Costs

Editing documents can often lead to unanticipated translation costs. Translation expenses are primarily based on word counts. Translation memory enables the reuse of translations across updates and in similar components, decreasing cost and increasing consistency. Reducing the number of words to be translated and edited, by writing and editing less, can drive down expenses. Using tag-based formats (XML and HTML) instead of formatted files (unstructured FrameMaker, InDesign, Word) can also reduce translation costs.

Secrets of Localization Pricing

In order to find ways to reduce translation costs, it is important to understand how they are calculated. There are usually several components, including translation, formatting, and project management.

Base Translation Costs

There are several approaches to professional human translation, including working with in-house employees, freelance translators, crowdsourced resources, or a multilanguage translation firm. Aside from internal staff members, who are paid a salary or hourly wages, translation is typically priced on a per-word basis.

The per-word rates for translation vary based on several factors:

- **Language** – Translation costs vary by language, largely due to the cost of living in the countries where translators work. For example, a translator for Simplified Chinese may live in China and be paid a third of what a translator who focuses on Swedish and lives in Sweden earns. Translation rates for common target languages tend to roughly follow the following order, from least expensive to most expensive:

- Chinese
- Latin American Languages
- Eastern European Languages
- South Asian Languages
- Western European Languages
- Japanese and Korean
- Nordic Languages

- **Qualifications and experience of translators** – Full-time professional translators tend to be more expensive than people with other jobs or students who do translation in their spare time (such as those who work in crowdsourced models). As translators gain more experience and higher degrees, they tend to charge higher rates; they may also be more efficient and produce higher quality translations.
- **Subject matter expertise** – Generalist translators who translate a variety of types of simpler content charge less than those who specialize in particular subject matter areas. For example, translators who focus on medical devices, complex engineering documents, and legal materials charge more. Creative marketing translators also garner higher fees.
- **File format** – Most file format processing is handled by translation workflow tools, so

Target Language	Crowdsourced Translation	Freelance Translator	Professional Firm (including review)
Simplified Chinese	\$0.06	\$0.08	\$0.12
Latin American Spanish	\$0.06	\$0.09	\$0.15
German	\$0.06	\$0.11	\$0.20
Japanese	\$0.06	\$0.13	\$0.25
Swedish	\$0.06	\$0.15	\$0.28

Table 1. Average base per-word human translation costs for a sampling of common target languages. Costs would be higher for more specialized subject areas and complex file formats.

translators rarely need to work with formatted files directly. Translation costs increase for files containing software strings and other brief text segments due to the additional time and effort required to translate content that lacks the context provided by sentences and paragraphs.

- **Translation management approach** – Working separately with crowdsourced or freelance translators for each target language usually results in lower translation costs than working with a multilingual translation firm. Of course contracting with individual translators for each language also creates additional management burden and risk.
- **Inclusion of review steps** – Generally professional translation firms include review and revision steps in their translation rates. These services are frequently referred to as “editing” and “proofreading.” Crowdsourced and freelance translators generally do not subcontract with separate linguists to undertake reviews so this service is not included in their base rates. Reviewers would need to be contracted individually and paid separately.

Impact of Translation Memory on Cost

Translation memory tools record the work of human translators. They store each segment of text in a database, linking the source to the translation. If the same source text appears in an update or is reused in another document or component each target-language translation may also be reused. A “segment” is typically a sentence, heading, software string, picture caption, table cell, or other stand-alone text element.

Translation memory offers the advantages of reducing costs, accelerating schedules, and improving consistency. Translation tools can be used by crowdsource resources, freelance translators, and professional firms. Creating and using translation memory actively should be a requirement of the translation process regardless of the approach or toolset selected. Most translation memory tools are able to output and read files in TMX format (an XML-based standard used for storing translation memories!).

When translation memory is used, translations are typically discounted accordingly as detailed in the table below.

Formatting Cost

Formatting of translated documents is usually charged on an hourly or per-page basis. This work is completed by professional translation firms, graphic designers, or in-house production staff members, but not typically by freelance or crowdsource translators.

Costs of formatting documents vary by the application used and target language, usually in a range of \$3.00 to \$12.00 per page. There are generally additional fees for editing of graphics, incorporation of screen captures, and creation of PDF files.

Documents in tag-based formats (XML or HTML) do not usually require individual page-by-page formatting. They need only a quick final-file integrity check of the tagged file or a review the PDF or webpage generated from it, taking a small fraction of the time of full formatting.

Tips for Reducing Expenses

Translation costs can be reduced by planning ahead early in the writing process and by keeping translation in mind during authoring and editing phases.

Limit the Volume of Text – Write Less!

Since translation is charged on a per-word basis, the easiest way to reduce cost is to limit or reduce the number of words. Assuming an average translation cost of \$0.20/word and density of 200 words per page, reducing a user guide from 250 pages to 200 pages would reduce translation costs by a total of \$10,000 for five languages.

The following strategies can help reduce the number of words in documents:

- Convert narrative explanations to bulleted lists
- Avoid repetitive warnings and steps
- Use direct language

Restrict Editing

If you have already had content translated, avoid changing it as much as possible. Every modification you make to the source text will make a segment no longer match the translation memory. That means your previous translations will all need to be edited.

For example, if you have a sentence that contains 14 words and has been translated into 20 languages, just adding or removing a comma will cost \$36. Probably nothing will need to be done to the translations themselves, since grammar differs in each language, but that segment will be flagged for editing, incurring a charge.

Edit only content that must absolutely be changed, where true errors exist. It may even be helpful to establish a policy that identifies what types of mistakes should be corrected (for example, perhaps only those related to safety or functionality, but not stylistic or minor grammatical issues).

Level of Translation Memory Match	Percentage of Base Translation Rate
Repeated text – same text appearing in multiple places in same component (e.g., a duplicated software string, a document footer used in multiple chapters, or a warning statement that applies to procedures in multiple chapters of a service manual)	10–33%
Exact match (100% match) – text that matches a segment stored in the translation memory, which could come from another file or component or from a previous revision of the same document	0–33%
In-context match (ICE match) – text that matches a segment stored in the translation memory and also appears in the same context it did when recorded in the translation memory (i.e., a sentence of documentation that is preceded and followed by the same sentences as it was when initially translated)	0–10%
Fuzzy match (75–99% match) – a segment of text that partially matches a segment in translation memory but has been changed somewhat (for example, a sentence with one or two modified words, perhaps to update a product name or correct a grammar or punctuation error)	50–75%

Table 2. Typical percentage charges for applying translation memory and reviewing text segments populated from memory. For example, a segment of fuzzy match text translated into Spanish by a professional translation firm would cost approximately \$0.10 per word ($\$0.15/\text{word} \times 66\% = \$0.10/\text{word}$). The exact percentage charged varies by content, translation approach, and supplier.

Reuse, Reuse, Reuse

Copying source text between components or from one product to another can be extremely beneficial. Since translations exist in translation memory, they can be reused even when appearing in another place. Instead of writing a new paragraph explaining how to install your new software, copy the one from your previous application and make some minor modifications.

Some companies buy tools to facilitate the reuse of text from a repository. Others just search old documents for keywords and copy relevant sections. If a new product or document is highly similar to one produced previously, it may be helpful to start with the old version and edit it for the new purpose.

Ensure Translation Memory is Built and Applied Aggressively

As explained above, translation memory is the key to saving money through reuse of translations. Ensure your translator builds a translation memory and provides it upon completion of the project. By owning and retaining translation memories in open formats (TMX), companies ensure the benefits from future releases.

It is important to ensure translation memory is applied aggressively to maximize leveraging. By negotiating no review of 100% match text during the translation of updates, you will reduce cost dramatically, particularly if there are few changes.

It is also useful to employ a strategy of cross-component leveraging. Requiring memory to be built for each component and leveraged between them will reduce cost through the reuse of segments of text that appear in multiple components. For example, an error message may appear in the software user interface and again in the related documentation. Leveraging this text reduces translation cost and ensures consistency.

Use a Tagged File Format

Documents written in FrameMaker, InDesign, Word, and other page layout and word processing applications require formatting in each target language. In particular, text expands in many languages, causing issues with pagination, line lengths, and formatting

style settings. Even if the majority of text is leveraged from translation memory, all pages must be formatted.

Choosing a tagged file format (such as HTML or XML) eliminates the need for this manual formatting. The file can be translated and target-language output generated automatically. Eliminating the formatting charge of \$3-\$12 per page is helpful for the initial publication and even more compelling for future updates with relatively minor changes.

Avoid Hidden Translation Cost in Changing Documentation Format

When changing documentation formats, particularly from a page layout application to a tagged format, as recommended above, there are risks of loss of leveraging from translation memory. The tags embedded to link documents to the page layout application differ from those used in tagged formats. For example, noting a FrameMaker style of "Subhead" is different than the HTML tag <H2> in the translation memory. Even if your user-facing text remains the same, the segments change relative to those in the translation memory.

Strategies to minimize the loss of leveraging when undergoing a format transition include the following:

Eliminate all tags from the translation memory so each segment contains only translatable words.

Search and replace old tags for new tags (e.g., the FrameMaker "Subhead" tag for HTML <H2>) in the translation memory database.

When completing the first translation in the new format have lower-cost resources review the segments to find and correct differences based on tags to isolate segments whose text for translation has not changed.

Resources

This paper is based on Adam Jones's 23 years of experience in the localization industry. Although no specific references were consulted in its creation, the following resources are helpful for people wishing to learn more about translation:

American Translators Association. <http://www.atanet.org>.

Faes, Florian, Managing Director. Slator Language Industry Intelligence. <http://www.slator.com>.

Kelly, Nataly, and Joel Zetzsche. *Found in Translation: How Language Shapes Our Lives and Transforms the World* (New York, NY: TarcherPedigree), 2012.

Parrish, Donna, Editor-in-Chief. *MultiLingual Magazine*. <http://multilingual.com>.

Sanders, Ella Frances. *Lost in Translation: An Illustrated Compendium of Untranslatable Words from Around the World* (Berkeley, CA: Ten Speed Press), 2014.

Various Authors. SimulTrans blog. <http://www.simultrans.com/blog>.

Author Contact Information

Adam Jones
Chief Operating Officer
SimulTrans
455 North Whisman Road, Suite 400
Mountain View, California 94043
650.605.1305
adam.jones@simultrans.com

Author Biography

Adam Jones oversees SimulTrans' worldwide operations, including project management, translation, engineering, testing, multilingual publishing, account management, sales, and marketing. Adam has spent over 20 years directing the company's customer outreach efforts, internal production groups, and other operations. Adam previously worked in Strategic Accounts at Oracle Corporation and as a high school English teacher. Adam graduated from Stanford University, where he studied Public Policy with an emphasis on Education. He remains connected to educational policy through active involvement in his sons' school district and related non-profit organizations.



Kerr, Debbie

Learning Styles and the Cancer Experience

Improving health literacy is critical to ensuring that everyone has the information they need to make informed decisions about their care. To do this, the information must be clear enough for patients and their families to understand and apply it to the situation at hand. While strides have been made to achieve this goal using plain language, there is more that can be done. Documentation will have a greater impact on its audience if it takes into consideration their preferences for learning, their situation, and the types of information that they have to process. For example, with a cancer diagnosis there are a multitude of terms and concepts to learn in a short amount of time in circumstances that are, to say the least, chaotic and very personal. When the stakes are so high, it is critical that, as technical communicators, we get it right.

Health Literacy Overview

The Public Health Agency of Canada says “60% of adults and 88% of seniors” do not have the required health literacy to understand and process medical information that is critical to achieve optimum healthcare (Public Health Agency of Canada, 2014). The United States has similar findings. The Centers for Disease Control and Prevention says, “Nearly nine out of 10 adults have difficulty following routine medical advice... Confused by scientific jargon, doctors’ instructions and complex medical phrases, patients are more likely to skip necessary medical tests or fail to properly take their medication” (Landro, 2010).

Both Canada and the United States have taken steps to try and resolve this issue.

In 2006, Canada took its first step to help resolve the health literacy problem by creating the Expert Panel of Health Literacy led by the Canadian Public Health Association. Their first report, published in 2008, focused on the correlation between people’s literacy and their health literacy; the lower the literacy level, the lower the health literacy (Rootman, 2008).

In 2010, the U.S. Department of Health and Human Services developed a Health Literacy Action Plan that

is based on the principles that people have the right to easy-to-understand health information that will help “them to make informed decisions” and services that will improve their “health, longevity, and quality of life.” The plan focuses on the use of plain language as the key to improving health literacy (U.S. Department of Health and Human Services, 2010).

While plain language is a huge step in the right direction, it is not enough. There are other factors like learning styles that, if considered, can further improve health literacy.

VAK versus VARK™

The concept of learning styles is not new. In fact, the concept of VAK (visual, aural/auditory, kinesthetic) learning may date back to ancient Greece (VARK Learn, 2017). This long-standing learning model remained unchanged until 1987 when Fleming determined that the visual component should be split. One component would be graphic-based and still be called Visual. The new component would be about learning through text and would be called Read/write. This resulted in a new model called VARK™ (visual, aural/auditory, read/write, and kinesthetic).

Although there are over 70 learning styles (Coffield et al, 2004), this paper will use the VARK™ model as its base because it is simple, easy to understand, and the most familiar to many people. Since this model does not take into consider other factors that can influence learning, this paper will identify some of those factors and provide examples based on my cancer experience.

Other Factors

There are many factors that affect learning such as personality, situation, and type of information to be learned. Each one plays a role in determining whether someone’s usual learning style may change.

Personality

I was surprised at the number of learning styles identified in Coffield’s 2004 study until I noticed that many of the learning styles listed had a personality component. For example, although most of us consider Myers-Briggs to be more of a personality test, it was one of the learning styles referenced in the study. In fact, the study states that approximately “2000 articles have been written about the Myers-Briggs Type Indicator (MBTI) between 1985 and 1995” (Coffield et al, 2004, p. 11). For me the large volume of articles and the inclusion of Myers-Briggs as a learning style highlights the importance of personality when it comes to how people learn.

For example, someone who is energetic and full of ideas may be more likely to jump in and experiment

than read a lot of material prior to taking action. Others are more methodical and like the opportunity to take their time learning by reading books or taking online courses. Some people are driven by facts and find it distracting and irritating if extra information is provided. Having a fact sheet available for this personality type would be useful.

Still others are interested in the feelings associated with an experience, like a cancer patient who is going through cancer treatments and is starting to lose her hair. She may also wonder what it feels like to have a biopsy or to have a port inserted for chemotherapy. An online forum would be useful for a person with this type of personality.

But not all cancer patients are the same. Some cancer patients are pragmatic. Their focus is on what has to be done. The emotional component and how a test procedure will feel is secondary to knowing that a specific procedure has to be done, even if it is going to be uncomfortable. In this case, a flowchart or a list of steps might be the best approach.

Situation

If you are not currently in school, think back to when you were. You had a lot of information to learn and, if you didn’t keep on top of things, you felt overwhelmed and lost. But still, you were learning over time. If you failed, you had the option of taking a course again even though you would rather not take it again at all.

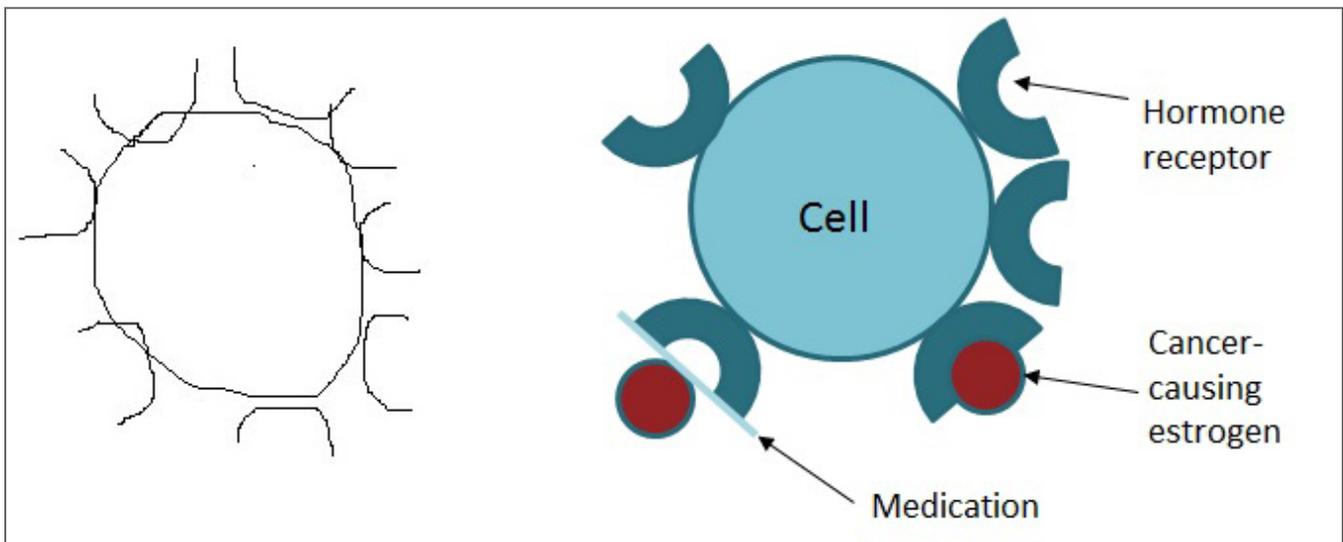


Figure 1. Doctor’s drawing of hormone receptors (left) and graphic of how medication works (right)

Outside the classroom, the impact of not completing a task correctly can have major repercussions. For example, if someone is doing their own wiring as part of a do-it-yourself project, the impact of not getting it right the first time could be disastrous, even deadly. Since the potential impact is so large, someone who usually learns by doing (kinesthetic learner) may decide to read instructions, look at diagrams, watch videos, or have someone walk them through the process rather than experimenting.

With a cancer diagnosis, I needed time to absorb and think about the information I received. I wanted information, but I did not want to see everything that was available to me on the Internet. I felt overwhelmed by the volume of data so I could only handle chunks of information at a time and only when I needed it. I did not want to find out about radiation when I was still having chemotherapy. I preferred just-in-time delivery of information. For me, the printed booklets that were available to me worked well because I could choose when I moved from topic to topic. I did not stumble upon information like I did on the Internet, because I could never stay on the page where I started. Some women (including me) stopped using the Internet until well after the treatments ended. I did not find forums where I could have talked to women who were going through the same thing. I would have benefited from a list of cancer-specific sites so that I would not have to do a search of the Internet. This list would have ensured that the sites I went to had accurate information. While I did not have this, this practice is happening more frequently now. In

a different situation, I would have continued to use the Internet to get information.

Type of Information to Learn

There is a hierarchy when it comes to explaining various types of information. The easiest information to convey is when someone can point at or act out something to get a message across. We use this method to teach children their first words or when there is a language barrier. If a child is young enough or someone does not speak or read English, the read/write learning style is not an option.

The next most difficult information to convey is a procedure unless it is simple enough to explain through actions. The complexity of the subject matter and the task to be completed will influence whether graphics, verbal explanations, written procedures, or videos are the best approach. In some cases, a combination of approaches may be the best approach.

Concepts and terms can be the most difficult to explain when you cannot see, touch, or do something to speed up the learning process. When conveying this type of information, an analogy can be useful because the concept is being tied to something familiar to the person. For example, explaining the digestive system can start with an analogy to a sink (possibly with a garbage disposal to simulate the initial stage of breaking down food). Attached to the sink can be a series of pipes that get progressively smaller until

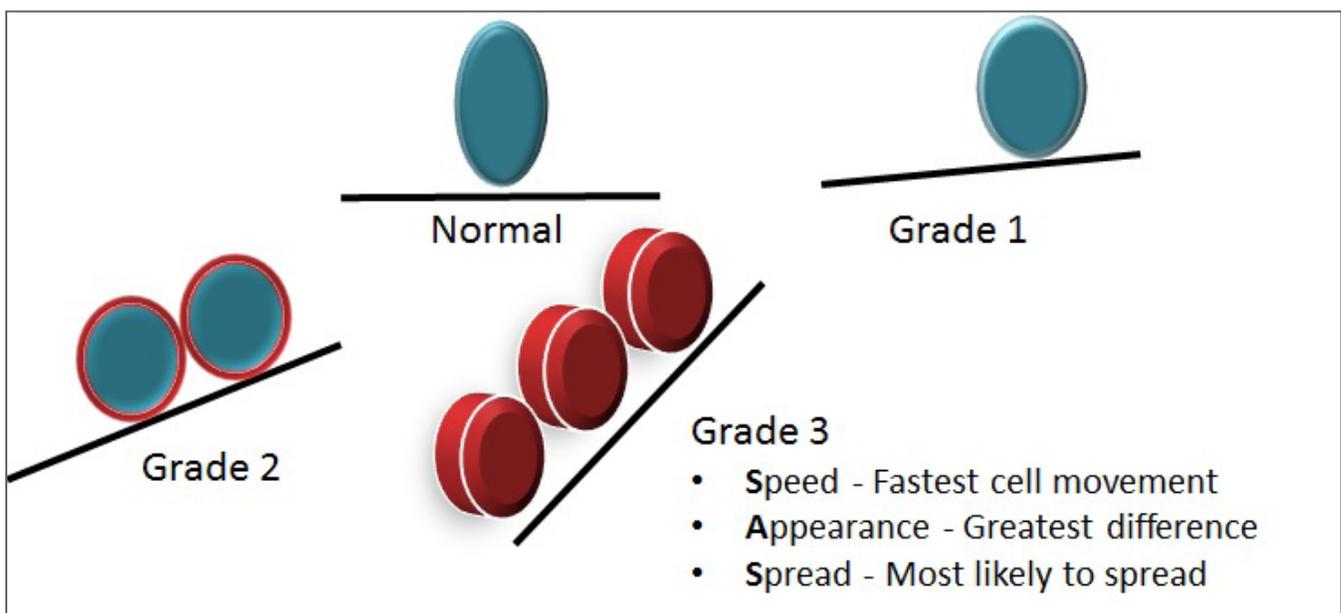


Figure 2. A graphical representation of the differences in cancer grade

water (in this case food) leaves the plumbing in the house. Once this basic concept is understood then additional details can be added and the proper terms for each of the parts can be introduced.

“Effective health literacy begins in early childhood and continually builds on knowledge and experience gained throughout the life span” (Canadian Public Health Agency, 2008). For this reason, using analogies to what is familiar to patients is a good method to communicate abstract information.

In my book, *When Cancer Takes Flight*, I use an analogy to taking a trip on a plane to explain and describe the cancer journey. The book goes from symptoms and tests to a diagnosis to treatments to the end of treatments and life after cancer. Like any flight, you have an airport where you start your flight to get to your destination. Not everyone’s journey is the same in between (Kerr, 2016).

With cancer, there is a trigger that either takes you for a mammogram (and then possibly to your doctor) or a symptom that takes you directly to your doctor. If you are diagnosed with cancer, your destination is to be cancer free (in remission).

For someone who has symptoms but no cancer diagnosis, it is a non-stop flight. For someone who receives a cancer diagnosis, there may be many stops and layovers along the way to becoming cancer free. There may be surgery, chemotherapy, and radiation (in any order).

Look at the Table of Contents of *When Cancer Takes Flight* to get a better sense of how the analogy works.

1. Introduction
2. Planning Your Journey
3. Your Baggage / Emotional Health
4. The Wait / Test Results
5. Takeoff / A Diagnosis
6. The First Stop / Surgery
7. A Layover / At Home after Surgery
8. Back to the Airport / Next Steps
9. The Flight Continues / Chemotherapy (Phase 1)
10. The Flight Continues / Chemotherapy (Phase 2)
11. Next Layover / Between Chemo and Radiation
12. Another Leg of the Journey / Radiation
13. The Landing and Jet Lag / Emotional Adjustments
14. Ongoing Maintenance / The New Normal
15. Closing the Book on Breast Cancer

Visual Learners

You do not have to be a visual learner to prefer graphics and images over reading large amounts of text.

In fact, our brains are pre-wired to process information more quickly when looking at an image. Here are some facts cited on the Visual Teaching Alliance website:

- Approximately 65 percent of the population are visual learners. – *Mind Tools, 1998*
- The brain processes visual information 60,000 faster than text. – *3M Corporation, 2001*
- 40 percent of all nerve fibers connected to the brain are linked to the retina. – Jensen, 1996
- Our eyes can register 36,000 visual messages per hour. – Jensen, 1996

A University of Saskatchewan study (Bateman, 2010) demonstrated that visuals also impact comprehension and long-term recall of information. In this study, two groups of charts were shown to test subjects. One group of charts consisted of traditional bar and

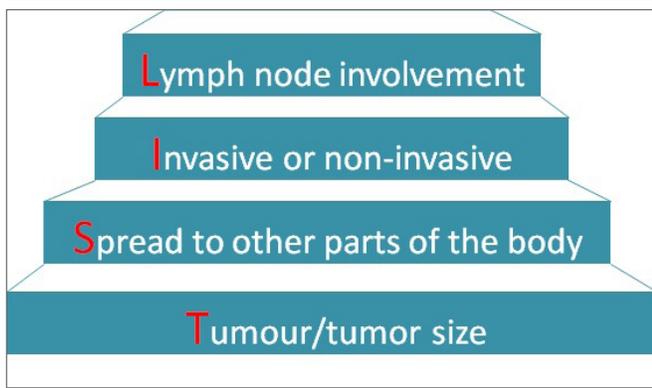


Figure 3. Factors to consider when determining a cancer stage

pie charts. The other group of charts had the data associated with a substantial amount of graphics. The initial belief was that the graphics would be distracting to viewers and result in a reduction in comprehension and recall. In the end, the opposite was true. During the time of testing and about five minutes after the initial test was completed, there was very little difference between the comprehension and recall of data between the two groups. The biggest difference became apparent when subjects were tested 12 or more days after the initial test. The recall of the charts with graphics was substantially higher than with traditional pie and bar charts.

For me, during my cancer experience, any picture that someone drew for me became memorable. It turns out that I'm not the only one. In talking to a friend who had the same oncologists, I discovered that she still had all the drawings that she had been given, even though she was going through cancer for a second time. Like me, it did not matter to her that the graphic would not make any sense to anyone else. All that mattered was that the drawing, plus the description the doctor gave verbally, pulled everything together.

Cancer Grade

With breast cancer, the lower the grade the better (for example, Grade 1 is better than Grade 2).

The grade is based on the following criteria:

- **Speed** – how fast the cancer cells are moving,
- **Appearance** – how different the appearance of the cells are from normal, and
- **Spread** – what the likelihood is that the cancer will spread. (National Cancer Institute, 2013)

To remember these criteria, I remember SAS (speed, appearance, spread). The greater the **SAS**, the more things go downhill and affect your grade.

Since grading cancer cells is a concept, I used the grade of a hill as a point of comparison because it is something that would be familiar to a cancer patient.

Aural or Auditory Learners

Auditory learners may like to hear themselves talk but that is only because it is a good approach for their learning style. Not only is it good for them to hear themselves read information out loud, they are a small group of people (10%) who do well in lectures where 80% of the information is delivered orally. (University of Illinois Extension, 2009, cited on Visual Teaching Alliance website, 2017). In addition to the classroom, these types of learners may do well when their doctor tells them their test results but only if they can understand medical jargon and are in the correct emotional state to remember what they have been told.

Based on my own cancer experience, I think the percentage of information delivered verbally is much higher than 80%. Only one doctor of many, over the course of almost a year, drew a picture for me.

I also believe the number of auditory learners may be increasing but only out of need. Since health literacy is so low, especially with seniors, people try to understand medical information by talking to others who may have been through the same thing. The problem is, since most people have poor health literacy, the odds are high that the information being shared will be incorrect, and potentially dangerous.

The good news is that there has been some great success using music to teach. One such example is the "Pump Your Blood" song (Dunne and Williams, 1979) that was performed on the TV sitcom Happy Days.

In Season 6, Episode 27 of Happy Days (a TV show from the 1970s), Potsie fears that he will fail his exam about the circulatory system. Luckily, Fonzie (the cool guy on the show) makes the connection between an auditory learning style and Potsie's love of music. Working with his classmates, Potsie writes a song, "Pump Your Blood," that explains how the circulatory system works.

Throughout the exam Potsie hums and taps his foot as he writes what turns out to be the perfect paper. Since Potsie's previous test results were poor, the teacher accuses Potsie of cheating. To demonstrate how he learned the material, Potsie sings the "Pump Your Blood" song as he points to a diagram (and later a model of the heart) while intermittently dancing around the room.

Today, a search of "Pump Your Blood" on the Internet returns videos of the classroom scene on Happy Days as well as newer versions, where creators have added graphics or displayed the lyrics as part of the video (Jkamcken, 2007). The search also returns classroom projects where teachers, 38 years later, continue to use this song as a teaching tool. The difference is that now teachers are asking students to create music videos. There's something for all learning styles.

Read/Write Learners

These learners are the ones who are the most likely to read the documentation that technical communicators create. I also suspect that many technical communicators have a read/write learning style, because we learn the subject matter that we read and write about.

When I had cancer, most of what I read was filled with medical jargon. Although there were websites that had plain language information, I stopped going online when I saw too many pieces of information that I was not ready to see. To learn the many concepts associated with cancer, I used analogies coupled with images and text.

For example, for most of us, the term stage refers to a place where someone stands to speak, sing, or act. With cancer, the term stage refers to something entirely different. Someone takes the necessary steps to provide your oncologist with a cancer stage. Use the word, "LIST" to remember the four factors that are evaluated (Breastcancer.org, 2017).

Kinesthetic Learners

Kinesthetic learners learn best through movement, which could be as simple as walking around a room while reading rather than sitting at a desk or using a keyboard and moving a mouse to complete an online tutorial. They like to experiment to figure out the right way to do something.

Even if our primary learning style is something other than kinesthetic, we have to become kinesthetic



Figure 4. Image used in the "Know Your Lemons" campaign

learners when the subject matter has to involve movement, like learning to play a sport. While you can learn these sports by reading a book or watching a video, it will never compare to actually doing it.

With cancer, patients and caregivers need to be able to see and review information multiple times. Hearing information once in the doctor's office is not enough. I was given a DVD to watch to learn about radiation treatments, but that did not allow me to actively take part in my learning. An online tutorial would have been more useful, especially for a kinesthetic learner.

Reneguette, an academic researcher and technical communicator at Indiana University-Purdue University Indianapolis, worked with a medical animation company and a surgery clinic to develop educational software to explain gastric by-pass surgery to patients (Renguette, 2016). Subjects completed a test both before and after using the software. The results of the second test showed improvement in how well the test subjects understood the surgery and how much better the test subjects used the correct terminology.

Know Your Lemons Campaign

"It's every communicator's dream to reach viral status for their campaign. In January, that happened to Worldwide Breast Cancer's #knowyourlemons breast cancer awareness campaign. In just 3 weeks, we reached over 166 MILLION people around the world." One of the reasons they give for their success is that lemons can go places on social media that images of actual breasts cannot (Beaumont, 2017).

In addition to the social media savvy, there are many other reasons why this campaign garnered so much attention and spread so quickly:

- The image is unusual and makes an immediate impact.
- The use of an egg carton makes it easy to see that there are 12 symptoms of breast cancer and not just the lump that people usually associate with cancer.
- It is a lot easier to recognize a symptom as a picture than as a description.
- Lemons are less intimidating to look at than looking at pictures/drawings of breasts.

There are also other graphics on the Worldwide Breast Cancer website. In fact, the entire site is structured like an infographic.

Conclusion

"Improved health literacy has been shown to make a difference in both patient satisfaction and in patient health outcomes. So, health literacy is not only about receiving and comprehending health information but also about being able to engage with the material and use the information to make better health decisions" (Renguette, 2016).

By using plain language and taking into consideration factors such as learning styles, personality, situation, and type of information being learned, technical communicators can improve clarity, comprehension, and recall of medical information. Since the time spent with healthcare providers can be brief, it is not always possible for them to assess someone's learning style, personality, and situation (other than the fact that the person may be ill). For this reason, the best way to present information is to use an approach that works for the greatest number of learning styles. Doctors verbally explain information (auditory learner). If, in addition to hearing a doctor's explanation, a picture and text are added for visual and read/write learners, three of four learning styles are addressed. If the kinesthetic learner takes notes during doctor visits, the last learning style is covered.

Given the emotional aspect of a cancer diagnosis, having an interactive tool, like an online course or tutorial about surgery or chemotherapy would allow patients to learn at their own speed, see real-life examples, and review information multiple times. The best method to teach anyone about cancer, or any other disease, is to use teaching tools that address most of the learning styles associated with the VARK learning model. Once doctors get to know their patients, adjustments can be made to accommodate a patient's personality and situation.

Resources

Lincoln Land Community College. "Characteristics of Learning Styles" Lincoln Land Community College (2016) <http://www.llcc.edu/student-services/cas/helpful-handouts/characteristics-of-learning-styles-2/>

Gutierrez, Karla. "Studies Confirm the Power of Visuals in eLearning" SH1FT DISRUPTIVE EARNING (8 July 2014) <http://info.shiftelearning.com/blog/bid/350326/Studies-Confirm-the-Power-of-Visuals-in-eLearning>

References

Public Health Agency of Canada. "Health Literacy" Public Health Agency of Canada (April 2014) <http://www.phac-aspc.gc.ca/cd-mc/hl-ls/index-eng.php>

Landro, Laura. "Taking Medical Jargon out of Doctor's Visits" Wall Street Journal (July 2010). <https://www.wsj.com/articles/SB10001424052748703620604575349110536435630>

Rootman, Irving, and Deborah Gordon-El-Bihbety. "A Vision for a Health Literate Canada, Report of the Expert Panel on Health Literacy" (Ottawa, Ontario: Canadian Public Health Association), 2008

U.S. Department of Health and Human Services, Office of Disease Prevention and Health Promotion. "National Action Plan to Improve Health Literacy" (Washington, DC: Author), 2010

Coffield, Frank. David Mosely, Elaine Hall, and Kathryn Ecclestone. "Should we be using learning styles? What research has to say to practice" (United Kingdom: Learning Skills and Research Centre), 2004

VARK Learn. "Copyright Information" VARK Learn (2017) <http://vark-learn.com/copyright-information/>

Kerr, Debbie. *When Cancer Takes Flight* (Guelph, Ontario: Full Circle Publishing), 2016

Bateman, Scott. Regan L. Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher Brooks. "Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts" (2010): 1-10

Visual Teaching Alliance. "Why Visual Teaching?" Visual Teaching Alliance (2017) <http://visualteachingalliance.com/>

National Cancer Institute. "Tumor Grade" National Cancer Institute (2013) <https://www.cancer.gov/about-cancer/diagnosis-staging/prognosis/tumor-grade-fact-sheet>

Breastcancer.org. "Stages of Breast Cancer" Breastcancer.org (26 January 2017) <http://www.breastcancer.org/symptoms/diagnosis/staging>

Dunne, Jimmy, and Anson Williams. "Pump Your Blood" song. Youtube (written and performed 1979 on Happy Days, video uploaded December 2010). <https://www.youtube.com/watch?v=gIXcWEObTwY>

Jkamcken. "Happy Days' Pumps Your Blood" song. Youtube (18 March 2007). https://www.youtube.com/watch?v=P_d0YkzQgY

Renguette, Corinne. "Technical Communication, Academic Research, and Patient Education: A Multidisciplinary Collaboration" Technical Communication (November 2016): 365-374 <https://www.stc.org/techcomm/2016/11/01/technical-communication-academic-research-and-patient-education-a-multidisciplinary-collaboration/>

Beaumont, Corrine E. "Know Your Lemons reaches 166 million in January." Worldwide Breast Cancer (29 March 2017) <https://www.worldwidebreastcancer.org/blog/know-your-lemons-reaches-166-million-in-january>

Author Contact Information

Debbie Kerr
Author/Speaker/Business Analyst
Guelph, Ontario, Canada
www.laughterandcancer.com
debbieakerr@gmail.com

Author Biography

Debbie Kerr has been a technical communicator for over 30 years, a cancer survivor for over five years, and a council member for 13 years with the STC Southwestern Ontario Chapter (four years as president). In her role as a technical writer and now as a business analyst, she sees herself as a user advocate. This is the reason she is so passionate about improving patient's health literacy through her book, *When Cancer Takes Flight*, her volunteer work as a Patient and Family Advisor with Cancer Care Ontario, and for speaking at conferences to the people who can help make a difference.



Kleinsmith, Mark

Video Provides the Edge

Video provides technical documentation groups the edge in communicating complex concepts to customers. When given the choice, would most people watch their problem be solved or read about it? At MiTek, U.S.A, our experience strongly suggests that customers want to see how to fix an issue or learn a new software feature. In the past year, our online library of over 1,300 targeted training and help videos recorded more than 52,000 Plays and 32,000 Finishes. The MiTek, U.S.A Technical Documentation Team manages an ever-expanding online help library that hosts videos ranging in length from one to three minutes. The site's primary goal is to assist internal and external customers in answering the question, "How do I.....?"

How Does Video Provide the Edge?

Like it or not, we live in a YouTube world. The typical problem solving process in 2017, whether you're fixing a clothes dryer, replacing a headlight or looking for an answer to a technical question, typically includes Google and a YouTube video. And, as more millennials become a part of the workforce, this trend is likely to grow. According to Intercom magazine, "Millennials

have become a critical demographic, one that has grown up with computers and cell phones. They are projected by the U.S. Census Bureau to number 75.3 million this year, surpassing the projected 74.9 million Boomers (ages 51 to 69). And, when it comes to troubleshooting technology issues at work, Millennials "tend to try three to five different strategies to resolve their issues before calling for help." So, how can your technical documentation group meet the needs of this

All,

I want to make sure everyone is reading and watching the videos from the MiTek Productivity e-mail that comes every Thursday morning. Please take the following quiz and reply with your answers. (Do not use "REPLY ALL") Just reply to me.

1. What are the two dormer types that can be input?
2. Can you delete a job in MiTek Link if it is "In Processing"?
3. What is the maximum number of planes that can be linked together?
4. Is it possible to design "cantilevered" outlookers with the Input Outlookers tool?
5. List one thing that you think will be a useful tool that you would use daily that you learned from this week's Mitek Productivity.

Thank You,

*Truss Design Supervisor
Christensen Building Components*

Figure 1. Quiz Email from Design Manager at Christensen Building Components

Thanks.... We all sat for about two hours reviewing the videos, actively demoing SnV, and ultimately buying into SnV.

Thanks,
 , Tech Rep
MiTek USA, Inc.

Figure 2. Email from Regional Technical Representative

growing demographic? The answer is to “...provide up-to-date video tutorials that can be easily found online.”

Video is a personal form of technical communication that creates a bond with customers. At MiTek, the Technical Documentation Team is now referred to as “The Voice of MiTek.” After producing over 1,300 help and training videos, we are MiTek! When we meet customers face-to-face or talk to them on the phone, we often hear a similar reaction, “Hey, I know that voice!” One of our most successful communication campaigns has been the distribution of a weekly “Productivity E-Mail.” Every Thursday morning, this email is sent to approximately 7,000 recipients and includes links to four videos. On average, the email generates 350-400 Video Plays and 200 Video Finishes. Customers look forward to the email each week. In fact, one design manager at Christensen Building Components in Lincoln, NE views the content and creates a weekly quiz for his designers.

In addition to meeting the self-help requirement that Millennials are searching for and creating a bond with customers, help and training videos build credibility with technical support. Over the past five years, Mitek Technical Support has learned that help and training videos satisfy customer needs and allow them to quickly answer support calls. Technical Support

representatives are confident that sending a customer a video link thoroughly answers the question.

To this point, we have discussed how videos connect your documentation group to customers and provide a much needed self-help outlet for the growing Millennial segment of the workforce. In addition to these valuable benefits, the MiTek Technical Documentation Team has found that our outside sales force is a proponent of video-based documentation. In the past year, The MiTek Technical Documentation Team has been invited to present at two regional sales conferences in order to promote the support website directly to customers. In addition, our weekly Productivity email and video based training/help library are often mentioned as part of the sales cycle. Our competitors do NOT have 1,300 help videos, weekly emails aimed at increasing productivity or free monthly online training sessions! A unique benefit of working closely with the sales and business development teams is the opportunity to work on-site in order to produce custom documentation for new accounts. The Technical Documentation Team has produced custom landing pages for customers converting to our software suite and produced custom documentation that details individual customer processes. Regional Sales Managers also promote our monthly training sessions, aimed at creating an educated and informed customer base.

Subject: WebEx on using Search & View

Good Afternoon Everyone,

There is a great upcoming WebEx on using Search & View in the software. It is 30 minutes long and is being offered 2x in December. I would highly recommend that you attend it. It is being offered on

- Thursday, December 10th, 3:00PM Mountain Time
- Friday, December 11th, 9:00AM Mountain Time

To register for one of the sessions, please send an email to [**Mark Kleinsmith \(mkleinsmith@mit.com\)**](mailto:Mark Kleinsmith (mkleinsmith@mit.com)) and indicate which session you will be attending.

Please visit the Search and View Home Page ahead of the webex to familiarize yourself with this productivity-enhancing design tool (you must be logged in to access the link below). <http://www.mitek-us.com/Software/SAPPHIRE/Private/Help/Videos/Structure/General/Search-and-View-Home-Page/>

PLEASE ensure that ALL of your designers receive this email!

Have a great day,

Regional Sales Manager

Figure 3. Email from Regional Sales Manager Promoting WebEx Training

Finally, adopting video as our primary medium for documentation has opened several doors within the organization. We now have our own Twitter handle, @MiTekSoftware, which allows us to communicate directly with internal and external customers. The Team also produces podcasts that are distributed to an internal audience. Podcasts represent another form of direct customer communication and increase the team's visibility. Because we are a part of a multi-faceted organization that is constantly expanding through acquisitions, we are often called on to produce documentation prototypes and consult on the benefits of merging documentation and communication groups. Now, when products require documentation, the MiTek Technical Community turns to us.

How Did We Achieve Success?

How did the MiTek Technical Documentation Team successfully convert an outdated, underused suite of help systems to a dynamic video library that records an average of over 2,500 Video Finishes each month? First, we selected the appropriate software and we continue to evaluate video creation and editing tools. Initially, the team used Captivate to record both screen captures and audio. In reviewing the videos that were produced five years ago, it's obvious that we've come a long way! However, we committed to starting the journey and we didn't wait for perfection, we work towards it. Now, the team has adopted a trio of tools, Blue Snowball microphones, the Adobe Creative Cloud which includes Audition for recording and cleaning audio tracks and TechSmith Camtasia for full screen, high definition screen captures. In addition, the team employees a shared office/work-from-home philosophy that ensures project collaboration and the ability to complete ad hoc recordings. With our flexible work environment, we are never searching for a conference room or office to record a video!

When the decision was made to create an online video training and help library, the Technical Documentation Team performed field research and consulted with internal customers to assess the marketplace's readiness for this shift in philosophy. We traveled to customer sites in order to interview designers and demonstrate prototypes. There was initial pushback based on the notion that customers did not have the bandwidth to view help and training videos. However, the Team worked through the objections and reached

consensus from external customers, sales and the internal technical support team.

Once the decision was made to proceed with video as our primary documentation medium, we were trained on an existing Content Management System (CMS) that was used by marketing and technical support to update the corporate website. While the CMS may have been, and continues to be "clunky," it meets our most immediate need of having complete control over the content displayed on the support site. We post videos today, the site syncs and the content is available to "the world" tomorrow. In addition, we use Vimeo to host all of our video content. For the most part, the video hosting site is "invisible." Videos are embedded on support.mii.com and most visitors don't know that the video is hosted on a third party site. The use of Vimeo as our hosting site prevents us from dealing with bandwidth and storage space issues. Instead, we concentrate on generating content.

After selecting the appropriate tools, gaining consensus from internal and external customers, and developing a process for posting content, MiTek's Technical Documentation Team marketed our online library of help and training videos.

While there were several marketing initiatives in the first year, the most successful were:

- Business cards that promote the site.
 - Thousands of business cards were printed and distributed to sales and technical representatives. Since we had buy-in from these groups, they agreed to leave the cards with customers as they made their rounds on a weekly basis.
- Customer WebEx sessions.
 - The Productivity Email was launched at approximately the same time as the support site. We advertised sessions that promoted the use of the site and showed customers how the site would be beneficial. At the same time we were marketing the site, we were generating content.
- Presentations at the annual Technical Symposium.
 - MiTek's Technical Community meets in St. Louis every summer for a week of continuing education. The Technical Documentation Team presented a "general session" where we reminded the technical community about the site and demonstrated new features.

- Attend industry tradeshows and speak directly to customers.
 - For the past five years, the Technical Documentation Team has been represented at our industry's largest tradeshow. We have a presence in the MiTek booth and spend the week talking to customers and promoting our online help and training resources.
 - In 2017 there will be "Help & Training" educational sessions off the show floor!

Throughout the entire shift to a video-based help and training library, MiTek's Technical Documentation Team has maintained a single core goal, "Create meaningful content." Within the department, we strive to never create a video on how to input a phone number. Instead, we work closely with Product Managers and other Subject Matter Experts in order to fully understand why features are developed. Understanding the "why" allows us to create meaningful documentation.

The video library, or support website, is currently being used as one of the core training tools for two major software launches. The Technical Documentation Team works closely with Deployment Managers to post video training series and over 150 training and help videos for products that are not yet in general release. The breadth of technical information that is available for products still in their infancy speaks volumes about MiTek's commitment to making video its primary medium for distributing technical information.

References

Deen McCrady, Victoria. "Millenials: A New Breed of Novice." Intercom 10.15:11 - 13. <https://www.stc.org/intercom/2015/10/millennials-a-new-breed-of-novice-user/>

Author Contact Information

Mark Kleinsmith
Technical Communications Manager
MiTek, U.S.A
8450 East Crescent Parkway
Suite 300
Greenwood Village, CO 80111
303.723.4965

Author Biography

Mark Kleinsmith has over 20 years of Technical Communications Experience in the Software Industry, from installing and supporting software systems to writing traditional help systems, training manuals and now an online video help library. For the past five years, Mark has focused on distributing technical information via targeted training and help videos. At MiTek, U.S.A, the technical communications team Mark manages has amassed a technical library of over 1,300 1-3 minute training videos. In the past 12 months, the website that hosts the technical library recorded over 52,000 Video Plays and over 32,000 Video Finishes. Mark's team delivered 54 days of video training and help content to internal and external customers. The team of four delivers targeted help content to the field every day. Mark looks forward to his presentation at the 2017 STC Summit and will share his experiences in order to give you tips on how to use video to provide an edge that separates you from the competition!



Kramer, Meredith

To Manage or Not to Manage: That is the Career Path Question

When looking at their career path most people are eventually faced with the question: Do I want to be in management? For some this is a no brainer and they can easily say, “Yes, sign me up!” or “No, I’d rather just do my job as an individual contributor.” Then there are others who just aren’t sure. For years I was in the “individual contributor” camp and I thought I always would be. Then, as different opportunities came my way, I started to waver in my decision. As our lives change, so may our career aspirations. So, whether you have never been a manager before or you have managed a team in what now feels like a past life, there are questions to ask yourself to see if management would be a good fit for you. This article guides you through one technical writer’s journey from individual contributor to manager and all of the introspective questions asked along the way.

How Do I Know Which Path to Take? Do You Have the Ability?

Haven’t we all heard that “knowledge is power”? That “knowing is half the battle”? How can we apply that same logic to guiding our careers? How might you know if you want to be a manager? Over the past 18 years as a technical writer I had long stretches of time where I was happy doing my daily grind and others where I wanted to expand my responsibilities. The biggest leap by far was into management. But I didn’t leap without looking, analyzing, evaluating, and learning first. After all, you need to know what you want before you can get it.

Before I could change my stance on the question “Do I want to be in management?” I needed to explore whether I had the ability to be a good manager. After all, if I wasn’t going to do it well I didn’t want to do it at all. However, it wasn’t enough to know that I could do it. I also needed to explore whether I wanted to do it day in and day out. Did I have the drive?

I’m pretty sure that most people think that almost anyone can be a manager, and while this may be true, not everyone is cut out to be a *good* manager. There are a plethora of books, articles, and blogs on how to be a good manager but there are some things you can ask yourself first to see if the job suits your personality.

Can Your Schedule Be Flexible?

There are many days that I have my calendar perfectly planned but little do I know that the moment I step into the office my plans will go out the window. You may arrive to a slew of emails that need immediate responses, team members that need a few minutes of your time, and a staffing crisis. When these things happen you have to be able to address issues as needed, make time for your team, and reprioritize tasks on the fly. It isn’t all fire drills and chaos, but it is important to know yourself well enough to know if that is the type of environment in which you can thrive.

Do You like Working with People?

Enjoying the company of others is nice, but this is more than just having lunch with your co-workers. There will be days when you have to have difficult conversations, such as when someone isn't performing up to department standards, one team member refuses to work with another, or someone keeps wearing perfume that is triggering another team member's asthma. Having a genuine interest in people and wanting to help them succeed is what will get you through the difficult conversations and will help you approach them in a constructive manner.

What Makes You Feel Successful?

There are days when I realize I spent eight hours on calls and answering emails but didn't actually produce anything – and I'm okay with that. I remember the moment I finally started thinking about moving into management. I had joined a project where the writers were drowning in work. The processes and expectations needed adjustments so I was brought on the team to take on the daily project management so the writers with the product knowledge could actually write. After another long day full of meetings I realized that by attending these meetings I was aiding in the productivity of three other people. That was a bigger impact than I could have made being assigned a few features to cover. Managers can make a big impact on the team and their company but may not have a tangible measure of their successes. The question you need to answer is whether you can find that work rewarding.

What Do You Hope to Gain?

Many people move into management because they feel that it is the next logical progression in their career or they want the title and a larger salary. As our industry evolves there are many avenues that weren't previously available, such as content architects and usability experts. These roles are other options for advancement and higher salaries that are sometimes a better personality or interest fit than management. By the way, there is a secret about salaries that most managers know. At one time or another you may have a member of your team who makes more money than you do. Being a manager isn't the only way to expand your salary range. Knowing your motivation for the job can help you determine if it is what you are really looking for.

What Do You Want to Accomplish?

I was at a point in my career where I was starting to think more strategically than tactically. I thought about what I could do as a manager that I couldn't currently accomplish as a lead writer. I wanted to take my motivation from aiding three other writers in being more productive and multiply it by helping more than one project team. For me, knowing what I wanted to accomplish, and could accomplish, as a manager helped me decide whether I wanted to transition to a management role on our team.

Does Your Career Need to Be in a Straight Line?

As our lives change so does our career. There are times when you have certain constraints and acknowledging them can help guide your career and set you up for success. As people have more obligations in their lives, whether it be young children, aging parents, or volunteer work, you may have less flexibility for work commitments such as long hours or travel. There was fellow writer who went from Lead, to Individual Contributor, to Manager over the course of many years. Our careers don't have to follow a straight path and there is a sense of relief knowing that what you choose today doesn't have to be forever.

Do You Have the Drive?

Andrew Carnegie said, "Show me a man of average ability but extraordinary desire and I'll show you a winner every time." The drive to achieve something can win out over raw ability. Knowing what type of work you want to do is a huge part of our happiness and success at the office.

You Can Do the Job but Do You Want To?

It isn't enough to know if you *can* do the job. You also need to decide whether you *want* to do the job day in and day out. Knowing what you want, and sometimes more importantly what you don't want, can help you make your own path instead of following one set by others. My dad is a CPA and my brother was at the Texas A&M School of Business with a 4.0 in upper level Accounting – of course I was going into Accounting. It was in my genes. Or so I thought. Let's just say Accounting and I weren't meant to be and it wasn't

until I followed my own path in Journalism that I was successful.

What Are Your Pros and Cons of Managing?

There is no shortage of blogs or articles about the pros and cons of management positions. A quick Google search can leave your head spinning. These articles can be a good place to start but it is important that you define your personal pros and cons. What one person thinks of as a pro another may see as a con. Each of us will determine our pros and cons by both the constraints in our lives (such as our ability to travel for work) and our comfort level (such as whether I get anxious at the thought of difficult conversations).

Is There a Way to Test Drive the Role?

The best way to set yourself up for success is to know you can do the job, and want to, before you get it. Before I became a manager I was the lead writer for all of our security line products. This gave me insight into managing multiple projects, working with writers on projects for which I wasn't writing, and the responsibility to ensure all of these projects were delivered on time with high quality. It was an excellent experience and it helped me to know that this was a path I wanted to take.

Seek New Opportunities

There may not be a ready-made opportunity to test drive a management position, but there is always the option to seek new opportunities. Whether it be offering to work with a team to update your company style guide or working on a grass roots effort within your team to create videos to supplement your current documentation you can put your leadership skills to the test.

Walking Your Own Path

With these questions answered I was able to determine that the management track was for me. After deciding what I wanted I went to our job ladder to determine how many of the tasks I felt I was already performing and which tasks I needed to gain experience with. I then talked with my manager about

my career aspirations and listened to her thoughts on what I could do to earn a promotion. This work allowed the team to see me as a leader before my promotion and helped with my confidence when making the transition to management.

For you there may be different questions to answer but as with all of those blog posts and articles on being a good manager, these questions are a good place to start in your introspective journey to determine if you can and want to be a manager.

Author Contact Information

Meredith Kramer
Information Development Manager
Micro Focus
515 Post Oak Blvd, Suite 1200
Houston, TX 77027
713.418.5400

Author Biography

Meredith Kramer, an Information Development Manager at Micro Focus, has worked in the technical communication industry for 18 years and has held several positions before moving to a management role.

She leads a team of 14 people in multiple geographic locations covering 15 different products. Meredith strives to help her team members find their particular areas of interest in our ever-changing field while encouraging them to guide their own careers.



Kreger, Jessica

Proving Return on Investment (ROI): Agilely Cutting Costs in Help

ROI isn't just a buzzword in the C-suite – it's a must-have for technical communicators and managers striving for the resources they need to succeed. Boost your bottom line through the tools of agile project management, single-sourcing, built-in context sensitive help, and information architecture for design. Learn how you can cut costs, improve efficiency, and achieve the metrics that you need to prove your business case.

Agile Project Management

Coordinating your efforts in an agile development environment will smoothly manage your projects, providing visibility, increasing productivity, and cutting costs. Using Atlassian Repositories such as Confluence and JIRA, you can plan two week iterative sprints on scrum boards, such as the one shown in Figure 1, below.

Single-Sourcing Help Content

Using single-sourcing technology, you can generate multiple help systems for several different markets from one repository. Centralizing your help files, conditioning the text, videos, and images for specific audiences, and customizing your targeted output will improve efficiency, reduce redundancy, and boost your bottom line. Popular software programs for single-sourcing content include MadCap Flare and Adobe FrameMaker.

Built-in Context-Sensitive Help Content

Meet your users where they are and enable them to request help on a specific task for your product through a help button, menu, browser, or pop up that appears

directly on their screen. For example, my company, TradeStation, enables users to right-click on a chart in TradeStation's desktop platform to display a help topic on chart analysis. Context-sensitive help saves the user time by empowering them to find what they need where they need it, improving the user experience and lessening the learning curve.

Information Architecture for Design

Content strategy expert Ginny Redish encourages writers to let their site visitors “grab and go” and “skim and scan” in her book, **Letting Go of the Words, Second Edition: Writing Web Content that Works**. We can also use the tools of information architecture to chunk content into digestible topics. Beyond the words, in his book **Beautiful Evidence**, information design pioneer Edward Tufte illustrates how we can effectively present visual information by turning “seeing into showing” and “evidence into explanation.”

Proving Your Business Case

You can prove the value of your work by delivering analytics on decreased costs for support calls, emails, and training. **Return on Investment (ROI) for Usability, 4th Edition**, a report by user experience consultants from the Nielsen Norman Group, describes other categories for metrics and Key Performance

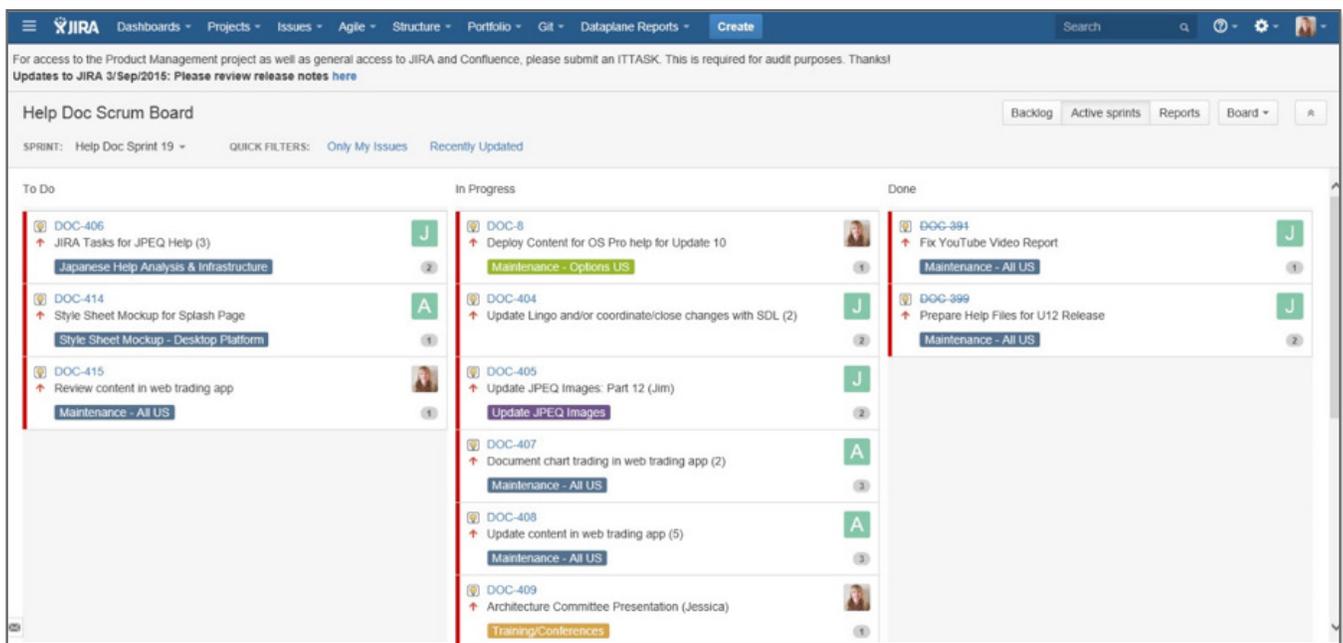


Figure 1. Plan your team's sprints in two week iterations on a scrum board to track progress and maximize resources. To capitalize on team synergies, sync your sprints with developers and engineers.

Indicators (KPI) that may be central to your case, including sales and conversion rates, traffic and visitor numbers, frequency of feature use, user performance and productivity, and development time.

Redish, Janice (Ginny). *Letting Go of the Words, Second Edition: Writing Web Content that Works* (Waltham, MA: Elsevier), 2012.

Tufte, Edward R. *Beautiful Evidence* (Cheshire, CT: Graphics Press), 2006.

Resources

Adobe FrameMaker (2017 release)-For XML/DITA Authoring & Publishing (14 April 2017). <http://www.adobe.com/products/framemaker.html>.

Confluence - Team Collaboration Software | Atlassian (14 April 2017). <https://www.atlassian.com/software/jira>.

JIRA Software - Issue & Project Tracking for Software Teams | Atlassian (14 April 2017). <https://www.atlassian.com/software/confluence>.

MadCap Flare for Online Help and More | Software for Technical Documentation (14 April 2017). <http://www.madcapsoftware.com/products/flare/>.

References

Nielsen, J., Berger, J.M., Gilutz, S., and Whitenton, K. *Return on Investment (ROI) for Usability*, 4th Edition (Fremont, CA; Nielsen Norman Group), 2012.

Author Contact Information

Jessica Kreger
Senior Manager, Client Training and Education
TradeStation Securities
8050 SW 10th Street, Suite 2000
Plantation, FL 33324

Author Biography

Jessica Kreger leads the technical documentation team at TradeStation to publish information that empowers active traders around the world. With over fifteen years of experience in communications, she has worked at Dell, the University of Miami, Carnegie Mellon University, and Alcoa. Jessica is passionate about creating optimal user experiences. She earned a M.A. in Professional Writing from Carnegie Mellon and a B.A. in English from Penn State. She is a Senior Member of the STC and belongs to the Technical Editing SIG.



Maddox, Sara

A Tech Writer, a Map, and an App

It's hard to create an app. It's even harder to get the app published and make it available to other people. That's true whether you're a developer or a technical writer. You need to put yourself on the edge and take the jump. You need courage, strength of conviction, and knowledge. Above all, you need documentation and examples. They give you the edge. By taking the jump into app development, Sarah has gained first-hand knowledge of what developers go through. She applies this knowledge to the documentation she writes. It's also a lot of fun!

This session teaches the technical details:

- The nuts and bolts of a web-based application like *Tech Comm on a Map*: where it's hosted, where the data is stored, the JavaScript code and the APIs that create the map and the app's functionality.
- How the app's data is crowd-sourced.
- What open sourcing your code means, and why you may want to do it.
- The difference between a web-based application and a mobile app, from a developer's as well as a user's point of view. *Tech Comm on a Map* is available as a native Android app as well as a webapp.
- The information sources that Sarah used when developing the app.

The presentation also shows how such a project can help develop your soft skills:

- Sarah's engineering colleagues helped her kick off the development of the app, and made ongoing suggestions for refinement. The resulting interactions increased mutual understanding and respect.
- Fellow technical writers all over the world help compile the data. A project like this is a good way of connecting with your peers.

- Developing an app can help you better understand your subject and your audience of software engineers and other specialists.
- Such a project gives you confidence in your own abilities, even if you're just skimming the surface of code complexity.

See *Tech Comm on a Map* in action at <https://sarahmaddox.github.io/techcomm-map>.

Why Develop an App?

Why would a technical writer want to develop an app - isn't that carrying audience research a little far?

To Better Understand the Audience

I wanted to experience their problems and needs first hand. As an API technical writer, it's hard to put yourself in the shoes of your readers. They're application developers. They'd rather read code than prose.

To Learn the Technology

There's only so much you can learn by following tutorials.

To Gain the Edge, and Help Developers Gain the Edge

It's hard to create an app. It's even harder to get the app published and make it available to other people. That's true whether you're a developer or a technical writer. By taking the jump into app development, I've gained first-hand knowledge of what developers go through, and I can apply this knowledge to the documentation I write.

To Create Something Useful and Interesting for the Tech Comm Community

I wanted to create a useful map for technical writers and all in the technical communication community. One that we can all contribute to.

To Do Something Different from the Daily Slog

Developing an app is a lot of fun, and very satisfying.

What is *Tech Comm on a Map*?

Tech Comm on a Map is an interactive web-based map that shows events of interest to technical communicators. It's an app for technical communicators, and technical communicators contribute to the data.

The web-based app uses HTML, JavaScript, and the web browser. *Tech Comm on a Map* is also available as a native mobile app, using the Android SDK and Java.

About Maps, Data, and Apps

Maps are made of images and data. The image may be a conceptual outline of continents, seas, and other geographic features. Or it may be an actual picture, such as a satellite image.

The data is something we add to the map, to convey a particular meaning. Perhaps we add labels to describe geographical features. Or perhaps we draw a polyline to indicate a route from one place to another.

In *Tech Comm on a Map*, the data is a collection of events and other items relating to technical communication. The app adds a marker, in the form



Figure 1. Tech Comm on a Map, running in a web browser

of a circle, for each item. When you click or tap on a marker, the app shows details of the event in a popup box, called an info window.

The app gets its data from a Google Spreadsheet. The data is crowd-sourced: a number of technical communicators contribute information to the map. Both the web app and the Android app draw their data from the same source.

There are also some controls on top of the map:

- The app name
- A search box
- A set of options to select event types
- The standard Google Maps controls

This presentation shows you how to get the image of the map into your app, and how to superimpose your own set of data and controls on the map. That's where code and APIs come in.

Components of the Web App

- These are the primary components of the web-based app for *Tech Comm on a Map*:
- To use the app, people need a web browser like Chrome, Internet Explorer, Safari, and so on.
- The code is HTML, JavaScript, and CSS. I'm using a JavaScript library called jQuery, which helps make the code more compact and ensures cross-browser compatibility.
- I'm using the Google Maps JavaScript API to get the interactive map.
- The search box comes from the Place Autocomplete widget, part of the Google Places API.
- The code is stored in GitHub, an online code repository.
- The web page is hosted on GitHub Pages, a static site hosting service provided by GitHub.
- The event data is in Google Sheets.

The presentation looks at these components in detail.

The Mobile App

There's also an Android app for *Tech Comm on a Map*. These are the APIs and SDKs used in the Android app:

- For the map: Google Maps Android API
- For the search box and place picker: Google Places API for Android
- To get the location of the device: Fused Location Provider in Google Location Services
- Data source: The Android app uses the same data store as the web app - the spreadsheet, with some Apps Script to push the data to the app.
- On-device database: Realm.io
- Draggable sliding-up panel: <https://github.com/umano/AndroidSlidingUpPanel>
- A validation library: Android Saripaar: <https://github.com/ragunathjawahar/android-saripaar>
- The code is on GitHub.

Open Sourcing a Project

An open source project is a project that is open to updates from the general public. The project content is usually code, but it could be something else, such as documentation.

The project owner decides who can contribute, and how. In many cases, anyone can suggest a contribution, by creating what's called a "pull request". In other cases, people have to be allowed into the project before they can make updates.

Tech Comm on a Map is an open source project on GitHub. Reasons for open sourcing a project include:

- Harnessing the knowledge and skill of the community.
- Helping other people give back to the community.

To open source a project:

- Choose a repository - somewhere that makes it easy for you to manage contributions from a large number of people. There are a few possibilities. Two of the best known are GitHub and BitBucket.
- Choose a license, and publish it in your repository. The license describes the terms under which your software or documentation is made available.

- Configure the permissions in your repository, giving people the rights to update the content, create pull requests, add issues, and so on, based on your requirements.
- Advertise your project - on social media, by word of mouth, amongst your special interest group.
- Keep an eye out for contributions, which will come in the form of pull requests.
- Check the changes in the pull request, and merge them into your project if you want them.
- Release the updated version of the app.

Lessons Learned

Here's a summary of what I've learned from this project. The presentation goes into more detail about each point:

- This project has given me confidence in my own abilities, in a highly technical field.
- I've gained understanding of my audience's character and needs.
- My colleagues and I have gained a mutual understanding of, and respect for, each other's abilities.
- The tech writing community is awesome.

Author Contact Information

Sarah Maddox
Technical Writer
Google
Sydney, Australia
Blog: <https://ffathers.wordpress.com>
Twitter: <https://twitter.com/sarahmaddox>



McCue, Shannon A.

Case Study: Finding Your Documentation Utopia in an Agile Environment

For writers who work in Agile companies, it can be tough to carve out a place in the overall process. How do companies share writers as specialists across several development teams? How do technical writers function as active, embedded members of an Agile Scrum team? Does it work? This case study shows how a technology company with an advanced Agile implementation embedded their writers on multiple development teams and how that same company later removed the writers from the teams to experiment with a different approach to shared team members.

My software engineering department successfully works within the Agile framework using Scrum. If you are familiar with the methodology, you know this means the core Scrum team members are typically developers, QA testers, Product Owners (POs), and Scrum Masters. How do the technical writers fit in?

Case Study 1

When I began working with the company 4 years ago, I was the only writer for a couple months. At my boss' request, I inserted myself into the process by embedding myself on Scrum teams. Being embedded meant I was an actual team member on not 1, but 3 or more development teams at any given time. I was the only person shared across multiple teams whose work affected the teams' sprint goals. My work life became a whirlwind of meetings (i.e., Scrum ceremonies such as planning, standups, etc.) and sprint deadlines. However, I worked alongside the developers in a document-as-you-go approach that I had not experienced in previous companies. The benefits of documenting iteratively with each sprint story easily outweighed the hectic schedule. I thought I had found my documentation utopia.

We hired another writer, and the company embedded her on several teams, too. Together, we were attending Scrum meetings for 6 or more teams, and we were

struggling to keep up with development output. The teams couldn't complete their stories until our part of the work was complete, which meant we sometimes delayed story completion and affected the sprint's success. Most times, it felt like everyone else had a sustainable work pace, but the writers were in overdrive.

The business monitored the impact of this embedded team member approach for over a year.

Pros

Pros of embedding writers on scrum teams"

- The writers collaborated with developers/QA daily to meet 2-week sprint goals.
- The development team had an active stake in completing documentation tasks with the writer so they could pass their sprint. Because documentation was part of a story's definition of done, a PO did not accept a story as complete until the team finished the documentation.
- We stayed informed during every aspect of the software development life cycle (SDLC), and we had high visibility into every change made to the software system.

- We had as much background information about a feature as the developers/QAs. Being embedded on the team meant asking our SMEs fewer questions during the document development life cycle (DDLDC).
- We had early input into development decisions, such as usability issues and GUI errors, because Scrum teams foster a highly collaborative environment. We were part of the think tank that designed solutions.
- Documentation kept pace with development. When development was complete, little-to-no lag time existed to compile the finalized documentation set.

Cons

Cons of embedding writers on scrum teams

- Any steps we took to create cohesion across documents were grass-roots efforts.
- We spent much of our time in Scrum ceremonies, and had very small chunks of time to do actual writing.
- We were sometimes bottlenecks at the end of the sprint.
- Many other teams needed writers, and we did not have the bandwidth to help them.
- We had no metrics to show the need for more writers. We tracked our work on various teams' virtual task boards, but had no way to show the drag from context switching.

Hiring more people seemed like the simplest solution to giving the writers a sustainable work pace, meeting all sprint deadlines, and reducing the likelihood of delaying a sprint. Reducing the number of teams per writer would take care of most, if not all, of our issues.

We hired a contractor. She was quickly in the same predicament as us. Embedded on 3 or more additional teams, she did not take away any of our workload. Instead, we expanded the number of teams/projects the writers could serve.

Our wish list for additional writers was met with the realities of staffing. The business needed a new approach to shared team members.

Case Study 2

In August 2016, we formed a non-traditional documentation Agile team acting as a service organization to the other teams. The goal of this experiment was to ensure the business used the writers efficiently. We needed to cover more ground with the same number of writers.

Embedded vs. Assigned

The business removed us from the development Scrum teams. Well, not really. Our status on the teams changed from embedded members to assigned members. Each writer still documented for 3+ teams across multiple projects, but we were no longer required to attend the mandatory Scrum ceremonies for all our development Scrum teams.

What about the definition of done? Thankfully, the business decided to keep documentation as part of the requirements for completing a development story. However, we changed what it meant to complete documentation as a developer. With the new definition of done, the developer simply logged a documentation request to our writing team. The request included several items enabling us to start work on it.

In the meantime, we gained remote writers to help manage our workload. Together, we worked toward being a more cohesive team that services 11+ development teams. In the previous process, we were not able to fine tune a process or document our standards.

Pros

Pros of assigning writers to scrum teams:

- We began producing more documentation.
- We created our own virtual Kanban story board using VersionOne®. We gained all the metrics we need to show how many stories we close, how many requests other teams make to us, bottlenecks in the process, cycle times, and more.
- We created a virtual backlog and started tracking technical-writing related work, such as updating a standards guide or writing this paper. It was visible to everyone what we are working on at all times.

- We gained a Product Owner and a Scrum Master. The PO prioritized our work. We started estimating each story's size and imposing work-in-progress limits on our Kanban story board.
- We began to follow a process for SME reviews, peer reviews, and Product Owner acceptance of each story. We couldn't implement peer reviews with any sort of regularity in the old process because of tight sprint deadlines.
- We learned more about other projects and started working toward being completely cross-functional.

Cons

Cons of assigning writers to scrum teams:

- We lost presence on development backlogs. We are behind development instead of documenting iteratively in the same sprint as the developers.
- We required additional people to support our Scrum team. Existing POs and Scrum Masters added our team to their current responsibilities.
- We had considerably more drag time from context switching than the traditional development Scrum teams at our company.
- We continued to spend much of our time in Scrum ceremonies and had only small chunks of time to do actual writing.
- We attended meetings for our assigned teams and the meetings for our documentation team because we didn't want to lose all the background information we gained from being active team members. The meetings (except standups) were optional, so we often opted out of development meetings. Skipping meetings had the potential for us to lose visibility into the product and some of the lower-level development details.
- When opting out of meetings, we relied more heavily on the development Scrum teams to know whether their stories required documentation.
- We still did not have enough writers to cover all the projects/teams needing writers.

Is this case study our new documentation utopia? Yes and no.

So much of the new process works well for us. We love having our own technical writing team, but until we get additional staff or completely remove ourselves from the development teams' Scrum process (or both), we will always struggle to keep pace with development. On the other hand, removing ourselves from the meetings may affect the documentation quality and certainly affects our product knowledge.

Conclusion

As technical writers, we constantly strive to produce quality products while streamlining our processes. My company commits to lean practices and a scaled Agile framework, which means we constantly measure, inspect, and adapt. To me, this approach means it is only a matter of time before we identify ways to turn the cons of our current process into pros.

If your company has reasonable limits on the number of teams/projects per writer, you may find that embedding yourself as a full-fledged team member works best. If shared team members are in limited supply, you may want to form a service organization similar to ours. Other approaches exist, too. An excellent blog post, *Patterns for Engaging Specialists in Agile Development*, by Ken Rubin, outlines several ways specialized professionals can work with Scrum teams.

The point of Agile is to be agile. Try new approaches, know what success means to you, commit to the experiment, and find your documentation utopia.

Author Contact Information

Shannon A. McCue
Technical Writing Manager
Video Gaming Technologies, Inc.
Franklin, TN 37067
Email: shannonalisha@charter.net

Author Biography

A technical writing veteran of more than 20 years, Shannon McCue is a hands-on writer, editor, and leader with a strong track record in creating thorough, usable documentation for end users, administrators, and business analysts. At her current company, Video Gaming Technologies, Inc., she manages the technical writing team and writes casino video

gaming documentation while immersed in the Agile framework. Over the last 4 years, she has become more focused on process improvement, especially the Scrum process and how it can work for or against technical writers.



Mincey, Chrystal R.

Knowing Your Client's Language

As you move through the contracting world, it is a given you will work with numerous agencies. If you are proficient, you may work at the same agency but in various divisions. In any case you will encounter different clients who have different desires, needs, and expectations. As a consultant, it is your job to know your client's 'language.'

Define Your Client

Depending on the task, the actual client may vary. You may report to particular government lead or be a resource to other teams within your Division.

Some of the things you should ask are:

- What is the need of that branch or unit?
- Who is the audience?

It is incumbent upon you to learn the language of the additional leadership and your target audience. Become multilingual.

What Is It That Is Expected?

Once you both speak the same language, know what your client expects of you. Generally this may be discussed upon coming aboard the project. What you say you can offer is probably how you were chosen. Make sure to follow through on your word. If the project changes make sure you adapt to those changes quickly. However, when in doubt, ask questions so that you have a full understanding of what is needed.

How Should It Be Done?

Different projects need different things. You may have done the same thing on a different project but every project is unique. This project may require agile methodology, whereas past projects used system or

software development lifecycle methodology. Know or learn the language of the project. Research the topic. Ask subject matter experts that have been on the project much longer. What is the correct way the methodology should be introduced or tailored to the current work environment? Make sure that your client agrees with your approach or that you have agreed on a compromise to the approach.

Why?

You may not agree with how a client agrees to approach the project, and that is okay. The customer is always right...on their project. Different agencies operate different ways. Do not just learn the language of the client, learn the language of the project as a whole. Be able to speak to your client on the same level so that you are working towards the same goal.

Author Contact Information

Chrystal Mincey, Member
Senior Technical Communication Professional
Bart and Associates
202-618-2119
cmincey@thealphaomegavirtualprofessional.com

Author Biography

Chrystal has been member of the Society of Technical Communication for a few years. She is a faculty member at Prince George's Community College and

the University of Maryland University College. She has been a government contractor since 2006. Chrystal is a Senior Technical Writer with Bart and Associates in Mclean, Virginia.



Navarro, Elizabeth

Everyone's an Editor

Having a full-time dedicated editor for a technical writing team is a luxury that few organizations can afford these days. Documentation quality suffers if no one's work is being edited, but how do you implement a peer editing approach and does it work? Is it possible for writers to be editors too? This paper maintains that peer editing is a viable alternative to hiring a full-time editor. It offers several peer editing approaches we have tried at NetIQ, and provides perspectives on the pros and cons of each approach.

Assumptions

This paper does not really address lone writer environments. It's intended for companies that have several writers, whether in one location or multiple locations.

There are lots of other sessions at the Summit that are tailored to lone writers and offer much more in-depth help with the special issues that lone writers face, but here are a few recommendations if you are a lone writer:

- Create your own style guide (or follow an existing one such as Microsoft Manual of Style) and/or project style sheets and follow them consistently
- Use tools like spell check without fail
- Set up your own production checklists to catch common mistakes

The Problem

With budget cuts and inadequate staffing, many companies are left with a staff of technical writers, but no editors. Documentation managers might acknowledge that editors have important skills, but given a choice, they often cut editorial positions rather than losing writers. As a result, documentation quality can suffer. Even if your documentation is reviewed for technical accuracy, how do you hold your team of

writers to the same writing standards? How do you ensure that your documentation is written with one voice? How do you get documents edited without a trained, dedicated editor (either in-house or contract)? Can writers be editors too?

I believe it is possible to successfully implement a peer editing process in a team writing environment, and in this presentation I will describe some approaches we have tried at my company (NetIQ, now a Micro Focus company), along with the lessons learned from each approach.

Historical Approach at NetIQ

We have always done peer editing at NetIQ. (Even when I first started working for the company over 10 years ago, we had a full-time usability specialist and a graphics specialist, but not a full-time editor.)

We had a company style guide that everyone was expected to follow. New writers read the style guide as part of their training. Depending on a new writer's past experience, we usually eased them into peer editing a little at a time with some mentorship. More experienced writers were expected to refer to the style guide whenever they had questions about our current standards.

Approach 1: Assigned Editors

At first we each had an assigned editor. Those of us who had more editing experience sometimes edited documentation for more than one writer. All writers were expected to follow the company style guide and make the changes that the editor marked, unless there was a good reason not to.

Less experienced writers who joined the team were given easier editing tasks to begin with, to develop their editing skills. Docs that they edited were reviewed by a more senior writer, who would discuss their edits with them and give them pointers.

About five years ago, we merged with Novell. We went through a process of combining our corporate style guides – which, fortunately, were pretty similar. At the time, they had a full-time editor at Novell, but expectations were different: Writers were not really required to incorporate the editor's changes. Edits were regarded more as suggestions or recommendations.

Pros and Cons of Assigned Editors

The main drawback of this approach was that some writers worked on products that had a lot more documentation, or documentation that was in bad shape already. That meant that their assigned editors had a lot more work than editors of other writers. Sometimes writers who had more product documentation work also had more editing work because they had more experience in both areas. It was really hard to find a good balance.

Some writers – usually less experienced ones – got the benefit of having a much more experienced editor reviewing their work. (This could be both a benefit and a drawback, depending on your perspective.)

Even though all editors were supposed to follow the style guide, we all have pet peeves and things that we notice or don't notice in others' writing. Having assigned editors meant that sometimes there was too much focus on certain types of grammatical or style issues and not enough on others. Writers would sometimes just start ignoring edits if they were tired of seeing them from their editor, and wouldn't always communicate with their editor.

<u>Upcoming Peer Reviews</u>									
Document	Author	Reviewer	Start Date	Due Date	Type of Review	# of pages	Notes to Reviewer	Status	
AppManager WinQS 7.9 Release Notes  AppManag...	Judy	Lee	8/17	8/20	Updated sections	3	What's New section and last 2 items in Known issues on page 6		
Sentinel - Sequence Timeout Rule (new section)	Amudha	Alyssa	8/17	8/19	Full	2			
CloudAccess Connectors Guide  connectors-commenta...	Elizabeth	Judy	8/19	8/24	See Notes	114 total	Little of the content has actually changed since the last release, but I have reorganized the guide, so it needs an overall/high-level review.	Done	
AppManager for JAVA UNIX Management Guide	Dean	Will	8/27	9/4	Full	32	This is a new document and has never had a peer review before.		
<u>Done Peer Reviews</u>									
Document	Author	Reviewer	Start Date	Due Date	Type of Review	# of pages	Notes to Reviewer	Status	
CloudAccess 2.2.1 HF2 Release Notes  releasenot... commenta...	Elizabeth	Judy	7/7	7/9		1.5	Very short, quick review.		

Figure 1. Example Peer Review Table in Microsoft OneNote

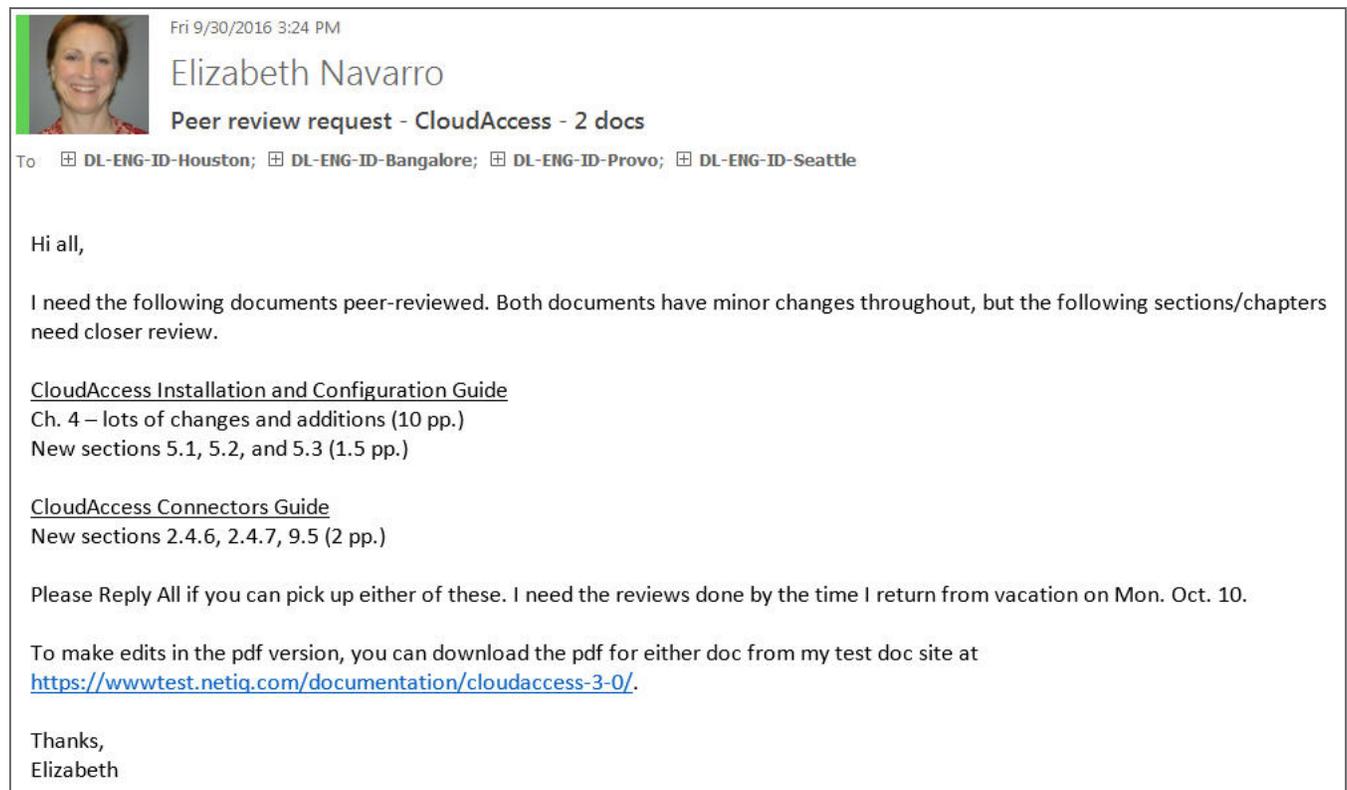


Figure 2. Example Peer Review Email Request

These issues were solved to some extent by rotating editors every year or so. However, unless you have a very large group of writers, it's still hard to maintain the right balance and not overload people.

Approach 2: Peer Review Table

A few years ago, we decided to try a different approach to peer editing at NetIQ:

We set up a peer review table in Microsoft OneNote and writers added documents that needed editing, along with dates and other details.

Pros and Cons of Peer Review Table

The main disadvantage of this approach was that people were busy with their own product docs and would forget to check the table for docs that needed to be edited. At first, no one remembered to check the table, so we started checking the table as a group at our ID team meetings. This worked pretty well when we had weekly team meetings, but as our workloads increased, we had meetings less often. Writers often had to email the team because their documents didn't get picked up quickly enough.

Another problem with this approach was that some writers didn't plan ahead and schedule editing time, or schedules changed, so they didn't always add their documents to the peer review table and things slipped through the cracks. (This seemed to happen more often than it had when we had assigned editors, probably because there was less visible accountability.) In some cases, writers might have deliberately not added their documents to the table, either because they secretly did not think they needed their work edited or they want to avoid the additional work of applying edits. In theory, managers knew what their writers were working on, but they did not always know if a guide had been properly edited.

Another issue was that sometimes writers in our India office had trouble accessing the OneNote notebook, or it didn't sync properly, so it didn't always work well for distributed teams.

The main benefit of this approach was that writers were able to balance their own workloads and didn't have to take on editing work when they were already overloaded. Another benefit was that a given set of documentation was not always being edited by the same person, so writers received input from peer editors with different perspectives.

Approach 3 (Current): Email Requests

About a year ago, we decided to stop using the peer review table and try a different approach. We now email the entire NetIQ writing team (writers in Houston, Provo, and Bangalore) when we need a document edited.

We specify the following in the email:

- Name of the document
- Whether the whole document needs to be edited, and if not, which pages/sections need editing
- How thorough an edit is needed
- The editing deadline
- Any other pertinent details

Note: To keep the size of our inboxes down, we typically do not attach the document to the initial email. However, if we are sending the email at the end of the day, we do sometimes attach the file so a writer in a different time zone who is willing to do the editing can pick it up immediately.

A writer responds (by doing a Reply All to the team) offering to do the peer edit.

The original writer sends the document to the peer editor in commentable pdf format and communicates directly with the editor if any clarification is needed after that point.

Pros and Cons of Email Requests

With this approach, we have had very similar issues to the peer review table (documentation not always being edited because of inadequate planning, etc.). However, in general I think this approach has been the most successful, as well as the easiest to implement.

Because we all get emails when someone needs a document edited, we don't have to remember to check the peer review table. We still have more control over our own workloads than we did when we had assigned editors.

Critical Success Factors

For peer editing to be successful in any organization, there are some critical success factors:

- All writers on your team must follow the same style guide.

If you haven't already developed an in-house style guide, I would suggest starting with the Chicago Manual of Style and the Microsoft Manual of Style and go from there. Take what you need, add to it, and customize it for your organization. One person should be responsible for updating the style guide and you must have a defined process (a style guide committee or something similar) for determining what your standards are.

- Writers must understand the importance of being edited.

In our interview process at NetIQ we ask for original writing samples, but we don't just look for candidates with good writing and technical skills. We've always included a grammar/editing section where we evaluate the candidate's ability to assess their own writing and willingness to accept constructive criticism from others on the hiring team. If they get argumentative or defensive during this portion of the interview, that's a big red flag.

- You must have strong enough leadership to enforce the rules.

Writers sometimes need to be reminded that they are expected to apply edits unless there is a good reason not to. Writers also periodically need to be reminded to go back to the corporate style guide to verify your standards. I often get style questions from writers that suggest to me that they are falling back on their own writing style when they edit another writer's work. They aren't referring to the style guide as often as they should, and they have apparently forgotten some of the new standards that we rolled out to all the teams when we combined our style guides. We have to do refresher training to make sure everyone is still on the same page.

- Accept the fact that some writers are better editors than others, but peer editing is still far better than no editing.

Writers who have no previous formal editing experience might need some help with editing. You must recognize that inexperienced writers (or just writers who are new to your organization) might be intimidated by the prospect of editing other writers' work – and having their own work edited as well. Besides pointing them to our style guide and training them on our standards, we've also put together some editing "tips and tricks" training. You can find lots of online resources and training too.

Author Contact Information

Elizabeth Navarro
Information Developer II
NetIQ (a Micro Focus company)
515 Post Oak Blvd., Suite 1200
Houston, TX 77027 USA
713.548.1700

Author Biography

Before Elizabeth gradually transitioned into technical writing, she was a full-time editor for several years (in Canadian government legislation, then in software documentation). She has worked as an information developer in various industries for over 20 years, but has continued editing and training others as editors. About half of her years as an information developer have been spent as a lone writer, figuring out her own standards. The other half have been in a team writing environment at NetIQ with a formalized peer editing process. She led a style guide committee for several years to determine and implement appropriate updates to the style guide. When NetIQ merged with Novell several years ago, she led a team to combine both companies' style guides, and has since provided training to NetIQ writers on the evolving standards. She recently developed some Editing Tips and Tricks training and is starting to work on some grammar refresher training for the writing team. Now that NetIQ has merged with Micro Focus and Serena, she is a member of two committees tasked with standardizing templates and addressing style guide differences among the merged companies.



Parkinson, Mike

Perfect: How to Turn Words and Data Into Powerful Graphics

Learn to turn your words, data, and ideas into clear, compelling graphics. Visuals, when done right, improve understanding, recollection and adoption.

What we see has a profound effect on what we do, how we feel, and who we are. Through experience and experimentation, we continually increase our understanding of the visual world and how we are influenced by it. Psychologist Albert Mehrabian demonstrated that 93% of communication is nonverbal. (Dr. Mehrabian notes that the actual percentage varies situationally but nonverbal communication carries great weight.) Studies find that the human brain deciphers image elements simultaneously, while language is decoded in a linear, sequential manner taking more time to process. Our minds react differently to visual stimuli.

Relatively speaking, in terms of communication, textual ubiquity is brand new. Thanks to millions of years of evolution, we are genetically wired to respond differently to visuals than text. For example, humans have an innate fondness for images of wide, open landscapes, which evoke an instant sense of well-being and contentment. Psychologists hypothesize that this

almost universal response stems from the years our ancestors spent on the savannas in Africa.[1]

People think using pictures. John Berger, media theorist, writes in his book *Ways of Seeing* (Penguin Books, 1972), “Seeing comes before words ... The child looks and recognizes before it can speak.” Dr. Lynell Burmark, Ph.D. Associate at the Thornburg Center for Professional Development and writer of several books and papers on visual literacy, said, “... unless our words, concepts, ideas are hooked onto an image, they will go in one ear, sail through the brain, and go out the other ear. Words are processed by our short-term memory where we can only retain about 7 bits of information (plus or minus 2). This is why, by the way, that we have 7-digit phone numbers. Images, on the other hand, go directly into long-term memory where they are indelibly etched.” Therefore, it is not surprising that it is much easier to show a circle than describe it (Figure 1).

When it comes to quick, clear communication, use a combination of the two. Include visuals and brief textual descriptions to increase the likelihood of success.

Making graphics is 50% concepting and 50% rendering (Figure 2). Concepting is the discovery process and rendering refers to the creation of your graphic for final distribution. Follow these six steps to develop effective graphics.

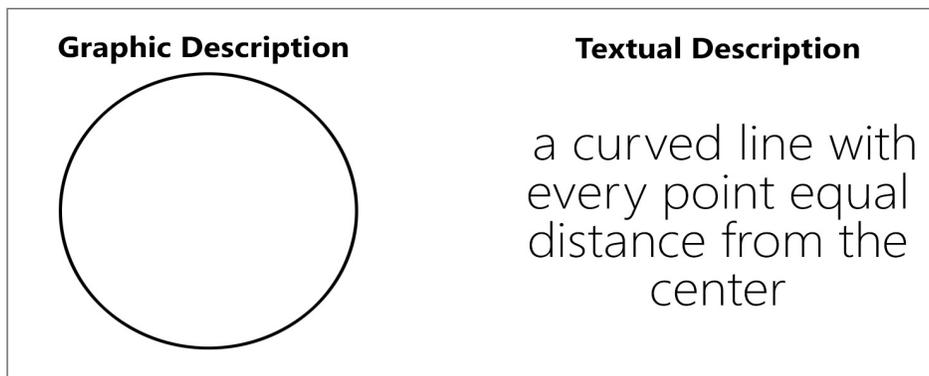


Figure 1. A graphic vs. textual description for a circle.

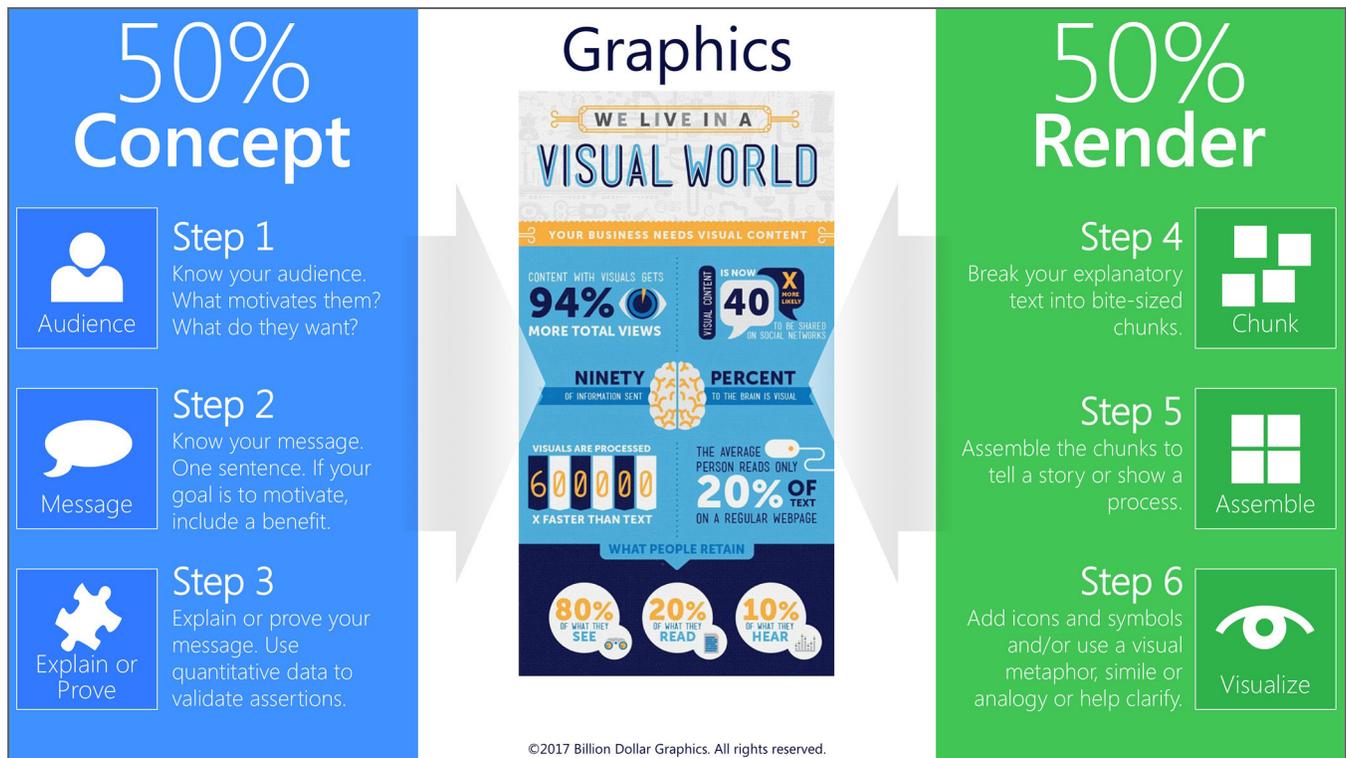


Figure 2. Make successful visuals using Mike’s visualization and rendering process.

Step 1: Know Your Audience

Know who they are, what they want to see, and why they should care. Learn what your audience truly desires. Your target audience is the sole reason why you are creating your visual. Tailor your graphic to your target audience. Make sure your audience can see themselves in your graphic. Connect with their world.

Use the message to develop the title of your graphic. For example, the title for this visual might be “3 Steps to Reduce Cost.” Your benefit can be explicit or implicit as long as your audience knows what they will get *after* they understand your graphic.

Step 2: Know Your Message

Your message is your takeaway. Your message is a one sentence summary of your graphic. I recommend including a benefit followed by how the benefit will be achieved. The benefit gives your audience a reason to care about your content. Figure 3 is an example of an appropriately structured message.

Step 3: Explain or Prove (Your Message)

The final conceptualization step is to explain or prove your message. For example, using the previous message (i.e., “Reduce cost in three steps.”), explain the three steps and prove how they reduce cost.

Step 4: Chunk (Your Content)

Once you have your message and explanatory text, chunk the information into small, bite-sized, digestible pieces. To chunk the information, highlight the most important elements and delete all other information. The goal is to



Figure 3. Takeaway structure to help ensure audience engagement.

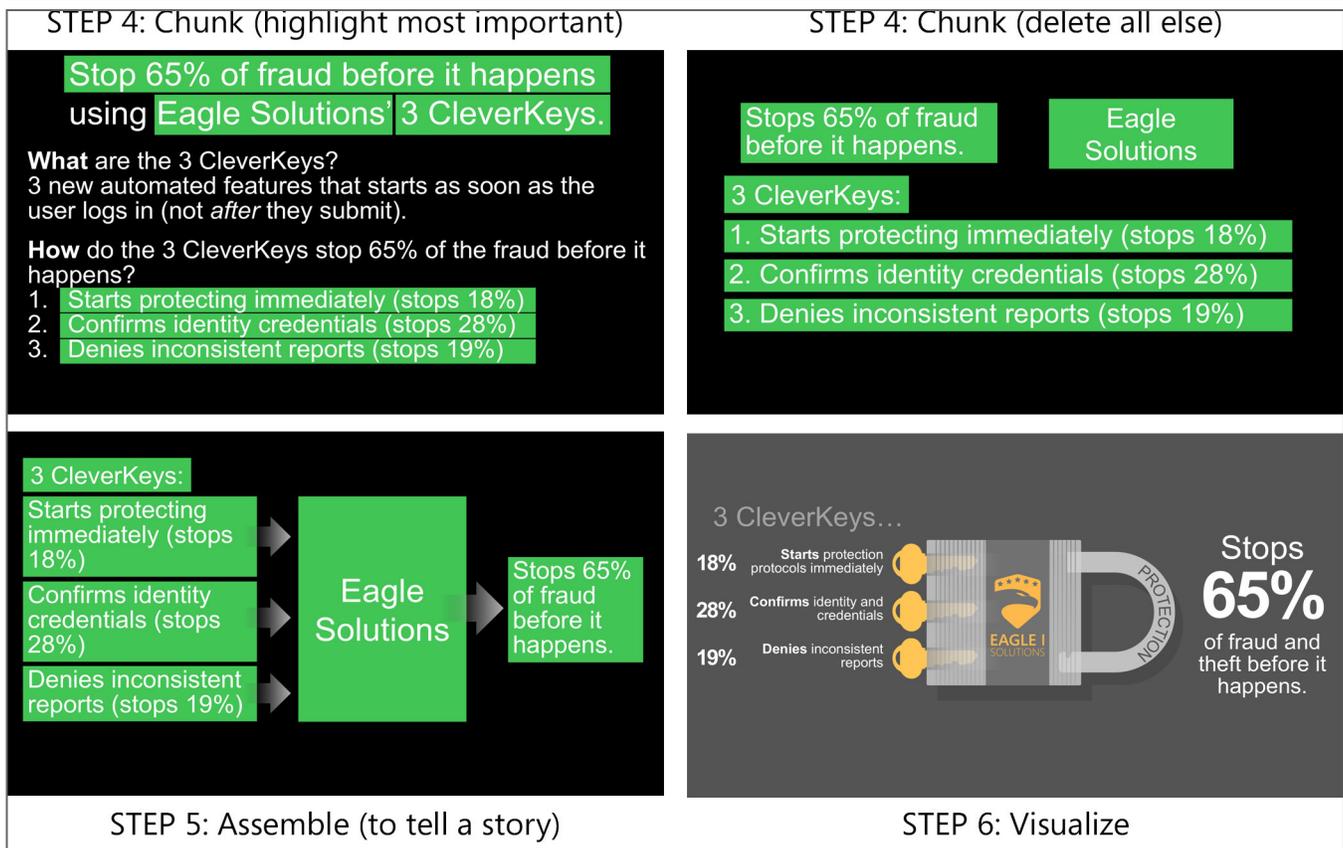


Figure 4. Steps four, five, and six examples.

eliminate words that are unnecessary to communicate your message and achieve your goals.

Step 5: Assemble (Your Chunks)

Next, assemble the chunks of information to tell a story. Your graphic should have a starting point and a conclusion. You want your content to have a logical flow that helps your audience understand (and agree with) your message.

Step 6: Visualize

Finally, use one of the following three visualization techniques to augment or replace your text.

- Literal Method: Literally show that which is being described.
- Quantitative Method: Use a quantitative chart when communicating amount, value or time.
- Substitution Method: Use a visual metaphor, simile or analogy to communicate content that is too complex for your target audience.

Figure 4 is an example of steps four through six. (This example was used for a PowerPoint presentation. Subsequent slides would further explain the three clever keys. In the final version, an alternative approach was used for the final rendered slide to illustrate that the keys secure the lock.)

Tip: Download a copy of my Graphic Cheat Sheet (www.BillionDollarGraphics.com/GCS.pdf) to quickly pick the best graphic type to communicate your content.

Follow these six steps to turn your words, data and ideas into powerful graphics that help you achieve your goals. Practice. The more you do, the better you will be. If you get stuck, contact me at mike@billiondollargraphics.com. I'm always happy to help.

Resources

Parkinson, Mike. *Do It Yourself Billion Dollar Graphics* (Annandale, Virginia: PepperLip), 2006 – 2016

Parkinson, Mike. *Graphic Cheat Sheet*. <http://www.billiondollargraphics.com/GCS.pdf>

References

- [1] Stevenson Johnson, “Beauty and the Beastly PC, The Graphics on Your Screen Can Affect the Way You Feel—and Think,” *Discover* Volume 25: Number 5 (May 2004): 20-21.

Author Contact Information

Mike Parkinson, Microsoft MVP, CPP APMP Fellow
Owner, Billion Dollar Graphics
7308 Ivycrest Place
Annandale, VA 22003
703.608.9568

Author Biography

Mike Parkinson (Microsoft MVP, CPP APMP Fellow) is an internationally recognized visual communication expert, trainer and award-winning author. Mike regularly works with and trains at large companies (e.g., Microsoft, Dell, Lockheed Martin), government agencies (e.g., NSA, CMS) and at international conferences (e.g., ATD, APMP, IVMG). Mike owns Billion Dollar Graphics (www.BillionDollarGraphics.com) and is a partner at 24 Hour Company (www.24hrco.com), a premier creative services firm and an owner. His training, creative services, *Billion Dollar Graphics* book, and *Graphic Cheat Sheet* help organizations and agencies achieve their goals using clear, compelling graphics. Contact Mike now at mike@billiondollargraphics.com for more information.



Perlin, Neil

We're Going Mobile! Great! But What Does That Mean?

This presentation discusses two facets of taking technical communication mobile – how to physically do it and the effect on content design and development.

Three Approaches to Go Mobile

The presentation discusses three approaches to going mobile. In order of increasing “mobileness”, they are:

- Apply responsive output to traditional help outputs – Responsive output uses media queries, fluid grids, and relative font size units to let technical communicators create one output that can automatically change its format, style, and even content depending on the device on which it's displayed. For example, if your content must be viewed on desktop PC monitors, tablets, and mobile phones, you can create one output rather than three.
- “Appify” traditional help – This approach converts the output of a tool like MadCap Flare or Adobe RoboHelp to a hybrid mobile app that should run on any mainstream mobile device, such as an iPhone or Android phone. This approach uses a conversion tool from Adobe called PhoneGap (<http://phonegap.com/>) that has to be downloaded and configured locally on the author's PC. A simpler option is the cloud-based variant PhoneGap Build (<https://build.phonegap.com/>). To use it, you upload the output of a project. PhoneGap Build converts the output to an app and sends you a URL and QR code for users to download the app. There are also a variety of APIs that can add app features like a camera and GPS.
- Create “true” mobile apps – This is usually viewed as engineering's job, but new GUI app development tools let business users create

powerful apps for business use, including for documentation. I use one such tool called ViziApps (www.viziapps.com). You can find a list of such tools at <http://tinyurl.com/hzz4j5c>.

Which Approach Should You Use?

The responsive output approach is the simplest. You may even be able to do it using your current authoring tools. You *will* get involved in new concepts like breakpoints, media queries, fluid grids, and dynamic content but today's authoring tools let you add most of these features without getting into code. The drawback is that responsive output may not meet your goals for going mobile in the first place. For example, if your mobile plans call for a camera, or geolocation using the GPS chip inside the phone, responsive output won't do it. If so...

Appifying your online output – converting it to an app via PhoneGap or PhoneGap Build – may do the trick. The output will be a hybrid app and PhoneGap offers a range of APIs that can add a camera, geolocation, a compass, and other features. The drawbacks to this approach include a) the fact that adding the APIs may take you into unfamiliar code or call for a programmer, b) the inability to dynamically change the content or let users add to or modify it, and c) the fact that while an app created using PhoneGap is indeed an app, it doesn't look like what people think of when they hear the term “mobile app.” In other words, there's a failure to meet users' expectations. If so ...

Creating a “true” mobile app eliminates all the problems noted above. The GUI tools let you add API-driven features with little or no coding, the data can be modified because it comes from a database rather than being static text, and it looks like what people expect an app to look like. (It *won't* look like Fruit Ninja™ but business apps rarely do.) The drawback – there’s always a drawback – is that creating and maintaining a true app will change how you work by getting you involved in using databases to hold the content.

Effects on Content Design and Development

How might going mobile affect your content design and development?

Here are some likely examples

- “Hacks” in the project. (A reviewer of a “Beyond the Bleeding Edge” presentation once said that “A hack is a one-off; good coding is forever”.) Hacks may be harmless in projects that stand alone but can cause a lot of trouble when the project has to be converted to a different format or its code has to be modified. The conversion tool may not know what to do with the hack. The result can range from a conversion with corrupted features to a conversion that crashes entirely. In either case, someone has to go into the project and fix the problems under deadline pressures. Getting rid of the hack before conversion can save time and money.
 - Local formatting in the project. Local formatting in place of styles isn’t a mobile issue, although it does bulk up the files, possibly requiring extra calls to the server which slows downloading, and is just bad coding practice in general. Any files to be made mobile should be checked for local formatting and cleaned up. This may also require creating a CSS or cleaning up a legacy CSS.
 - Write for mobile first. We tend to write for the most common device size, usually a desktop monitor or laptop. This means we write a lot of content because we have the screen real estate for it. The “Mobile First!” idea by Luke Wroblewski (<http://www.lukew.com/ff/entry.asp?933>), suggests writing for mobile first. One effect is that it forces us to focus on the really important content and leave out what isn’t. And if some content isn’t important on a mobile phone, it may not be on a tablet or desktop either. Our overall content shrinks.
- Making our writing shorter and more focused isn’t that hard to do for new content but can be very difficult for existing content, where extensive editing can change the tone as well as the content. (The first line of Alfred Noyes’ poem “The Highwayman” starts “The wind was a torrent of darkness among the gusty trees.” Cutting this to a length more appropriate for mobile gives us “It was windy.” It works, but there’s something missing...)
- Rethink tables, which are often just containers for rarely used information and don’t translate well to the small screens of mobile devices. For example, consider a table that lists the fifty US states alphabetically and, for each state, lists the capital and the state flower. Typically, users only want information about one state so forty-nine lines of the table go unused and waste space. Consider whether it would be possible to replace the table with a one-column list of the states, each linked to a topic that presents the information. Or perhaps go further and replace the whole table with a search box?
 - Consider how changing navigation is affecting interfaces. For example, indexes have been slowly but steadily disappearing from online content but the typical tri-pane window still offers an Index tab. But MadCap’s new top-navigation, or topnav skin has no provision for an index. Might this be an issue? Do you know if your users actually use your indexes?
 - Keep your authoring tools up to date. I often encounter companies that are using a very dated version and then want to upgrade to the latest version, skipping multiple intermediate versions. The problem is that there might be features in the old version that were slowly deprecated over successive releases but cause trouble when you leapfrog multiple versions because the feature conversion mechanisms are no longer available, or assume that the feature has been converted. One example of this is the multilevel lists in RoboHelp. They work well in RoboHelp but literally do not convert to MadCap Flare because the multilevel list code is not standard HTML.

Summary

This paper has glossed over the details of the three conversion options and skimmed the issues related to how our content design and development will change. Suffice to say that the shift from traditional online to mobile, or “Mobile First!”, may be as wrenching as was the change from print to online in the early 1990s. Which of course means new and challenging work for the new generation of technical communicators.

Neil is the author of eight books on computing, most recently Essentials of MadCap Mimic 6, Advanced Features in MadCap Flare 10, and Creating Mobile Apps without Coding.

In what passes for spare time, Neil builds telescopes, uses simulators to recreate a lost career as a Navy pilot, and cooks southern barbecue. You can follow him on LinkedIn, Facebook and Twitter (at NeilEric).

Author Contact Information

Neil Perlin, STC Fellow
President
Hyper/Word Services
Tewksbury, MA
www.hyperword.com
nperlin@nperlin.cnc.net
978.657.5464

Author Biography

Neil has 38 years of experience in technical communication, 31 developing, training, and consulting for various online formats and tools, some modern and cutting edge, others now known only in legend. He was a partner in a documentation consulting firm in the late 1980s, where he created the world's second and third commercial hypertext products – online user guides for MS-DOS 2.1 and Lotus 1-2-3 2.01.

Neil founded Hyper/Word Services in 1990. The company provides training, consulting, and development for online help and documentation, Flare, RoboHelp, Mimic, XML, single sourcing, structured authoring, and mobile apps. Neil is Madcap Certified for Flare and Mimic, Adobe-Certified for RoboHelp, and ViziApps-certified for the ViziApps Studio mobile app development tool.

Neil spent years as a columnist for STC and IEEE PCS. He is a frequent speaker at professional conference and created and ran the Beyond the Bleeding Edge session at the STC summit from 1999 to 2006 and again from 2008 to 2014. Neil also spent four years as STC's lead representative to the WorldWide Web Consortium from 2002 to 2005. He is a Fellow of the STC.



Peters, Ear

Section 508: Are You Ready for Americans with Disabilities Act (ADA) Compliance Standards?

Ensuring that websites and the content posted on them are accessible to users with disabilities has been a requirement for federal government offices and government suppliers for over a decade. Section 508 of the Americans with Disabilities Act has provided strict guidelines that outline the specific needs and how they can be met. However, there is an increasing push for nongovernment entities such as retailers and manufacturers of consumer products to require a comparable level of accessibility, and numerous lawsuits have added to the pressure on companies. While no official mandate yet exists, and numerous proposed deadlines for one have been shifted or extended, the current target is 2018. This presentation helps demonstrate what ADA compliance means for your content and website, and why it matters to your organization.

What Is Section 508?

When the Americans with Disabilities Act was passed in 1990, the focus was preventing discrimination in employment, public services, public accommodations, and telecommunications. Things like wheelchair accessible entrances and restrooms in stores and restaurants, the addition of braille to elevators, and closed captioning on TV are just a few of the more visible manifestations of this landmark legislation. At that time, the Internet was in its infancy.

Fast-forward 27 years, to the era of Amazon, Netflix, and Facebook. The Americans with Disability Act was amended in 1998 with Section 508 for federal government websites, and updated further in 2001, with the addition of Electronic and Information Technology (EIT). This addition set clear guidelines and standards for federal government agencies, as well as major suppliers and government-funded nonprofits and educational institutions, to ensure accessibility to websites and the content housed on them. Since then, this category has continued to grow and evolve as digital technology and our reliance on it has increased.

What about the Private Sector?

Having strict standards for federal government agencies and taxpayer-funded organizations is all well and good, but as technical communicators, many of us work solely in the private sector, typically writing manuals or developing web content or supporting software solutions and apps for consumer-facing products. So why should we worry about ensuring our content is ADA-compliant?

Here are a few simple reasons:

- You are improving accessibility and ease-of-use, regardless of your target consumer, meaning happier users.
- It reduces the risk of discrimination lawsuits.
- It's good business, and shows you recognize and value a large and often underserved part of the population.

In addition, in a 2015 statement, the Department of Justice clarified that public and private websites should fall under the jurisdiction of the ADA, but have not officially set any regulations or a schedule for implementation and enforcement.

However, the issuance of a proposed guideline was originally expected in 2016, and is now expected in 2018. Depending on the scope, compliance/enforcement timelines, and any rules for existing content, this guideline could have a massive impact on U.S. companies, ranging from consumer goods, to entertainment services, to retailers, to social networks. The expectation is that private compliance requirements will be consistent with those for government sites, with allowances for company size, public footprint, and exempt content.

Building Compliant Content

Building ADA-compliant websites and content doesn't just make sure that you meet a government requirement; it is a helpful step toward improving usability in general. If you build your content to meet the standards of Web Content Accessibility Guidelines (WCAG) 2.0, you will meet ADA-compliance requirements for almost every situation. Providing detailed guidance on all of the specific requirements for ADA-compliant content and WCAG 2.0 standards would be a daunting and time-consuming task. However, I would like to provide an overview of the key areas, and provide some examples of some common and easily correctible issues, and provide resources to dig deeper.

What Types of Disabilities Need to Be Addressed?

The ADA is designed to address a wide scope of disabilities, ranging from the obvious to some that can be very subtle and are often overlooked by designers. According to the 2010 census, 56.7 million Americans have some sort of disability, representing nearly 20% of the population. As our society continues to age, these numbers will continue to increase, even as we become more reliant on digital communication pathways and devices.

Some common (and some often overlooked) disabilities addressed by the ADA and WCAG standards include:

- Visual impairment (blindness, macular degeneration, color blindness, etc.)
- Hearing impairment (limited or complete hearing loss)
- Mobility issues (paralysis, fine motor skills impairment, amputees, etc.)
- Cognitive
- Other disabilities, such as epilepsy

To address these, WCAG 2.0 breaks its guidelines into four key categories:

- Perceivable

How to Meet WCAG 2.0
A customizable quick reference to Web Content Accessibility Guidelines (WCAG) 2.0 requirements (success criteria) and techniques.

Selected Filters: all success criteria and all techniques.

1. Perceivable

- 1.1 Text Alternatives
 - 1.1.1 Non-text Content
 - 1.2 Time-based Media
 - 1.2.1 Audio-only and Video-only (Prerecorded)
 - 1.2.2 Captions (Prerecorded)
 - 1.2.3 Audio Description or Media Alternative (Prerecorded)
 - 1.2.4 Captions (Live)
 - 1.2.5 Audio Description (Prerecorded)
 - 1.2.6 Sign Language (Prerecorded)
 - 1.2.7 Extended Audio Description (Prerecorded)
 - 1.2.8 Media Alternative (Prerecorded)
 - 1.2.9 Audio-only (Live)
 - 1.3 Adaptable
 - 1.3.1 Info and Relationships
 - 1.3.2 Meaningful Sequence
 - 1.3.3 Sensory Characteristics
 - 1.4 Distinguishable
 - 1.4.1 Use of Color

Principle 1 – Perceivable
Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1 – Text Alternatives
Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 Non-text Content — Level A
All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.

1.2.1 Audio-only and Video-only (Prerecorded) — Level A

Figure 1. W3C's website provides helpful information and tools to provide guidance on WCAG 2.0 standards

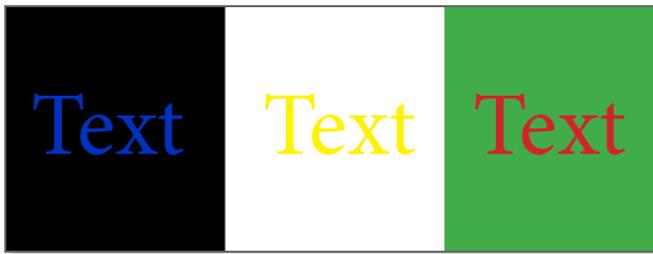


Figure X. Some examples of text with poor color combinations for contrast or color blindness

- Operable
- Understandable
- Robust

Each focuses on a different aspect of how information on a website is communicated, and each represents its own unique challenges and opportunities.

Perceivable

When creating content, it is important that it be presented in a way that doesn't exclude a particular category of users due to how it is displayed. For example, having a diagram explaining the critical assembly of something does little good if the user is relying on an electronic reader to read it aloud, and there is no form of alt-text to explain what is shown in the diagram.

Ensuring that content meets the definition of "perceivable" impacts how different content is handled, such as text, images, videos, audio, or graphics.

For example:

- Visual
 - All relevant content is tagged
 - Text is scalable, uses easy-to-read fonts, and meets contrast requirements to allow for improved visibility
 - Layout, charts, text, etc. avoids common color combinations problematic for those with color blindness, and does not rely on color to distinguish information
 - All images include alt-text that is readable by accessibility tools
 - Video include an audio descriptive track
 - Form fields include text descriptors

- "Decorative" text and elements, such as borders, are not tagged
- Audio
 - Video content should include subtitles or closed captioning
 - Audio content should have a readily available text transcript
 - Avoid using signal tones without an alternative form of communication, as well
- Mobility and Other Disabilities
 - Provide alternative options for time-based interactions
 - Avoid flashing lights/graphics that could trigger seizures
 - Ensure that content can be presented in alternative ways, such as simplified layouts, without losing information or structure
 - Minimize distractions

Operable

To ensure that content is readily accessible to users, the site must be easy to use. This impacts everything from the general layout of a website or document to how the way inputs can be provided.

Some of the most important aspects of operability include:

- Clear navigation that makes it easy for users to find content and determine where they are on a page or in a document
- Enabling the user to access all functionality through a keyboard only, without relying on a mouse or other secondary inputs
- Making sure that interaction points such as hyperlinks, check boxes, radio buttons, etc., are clearly identified and large enough for easy interaction
- Ensure that there is sufficient time for a user to interact with a page
- Content, including tables and alt-text for images, flows in a logical and sequential way when presented with accessibility tools

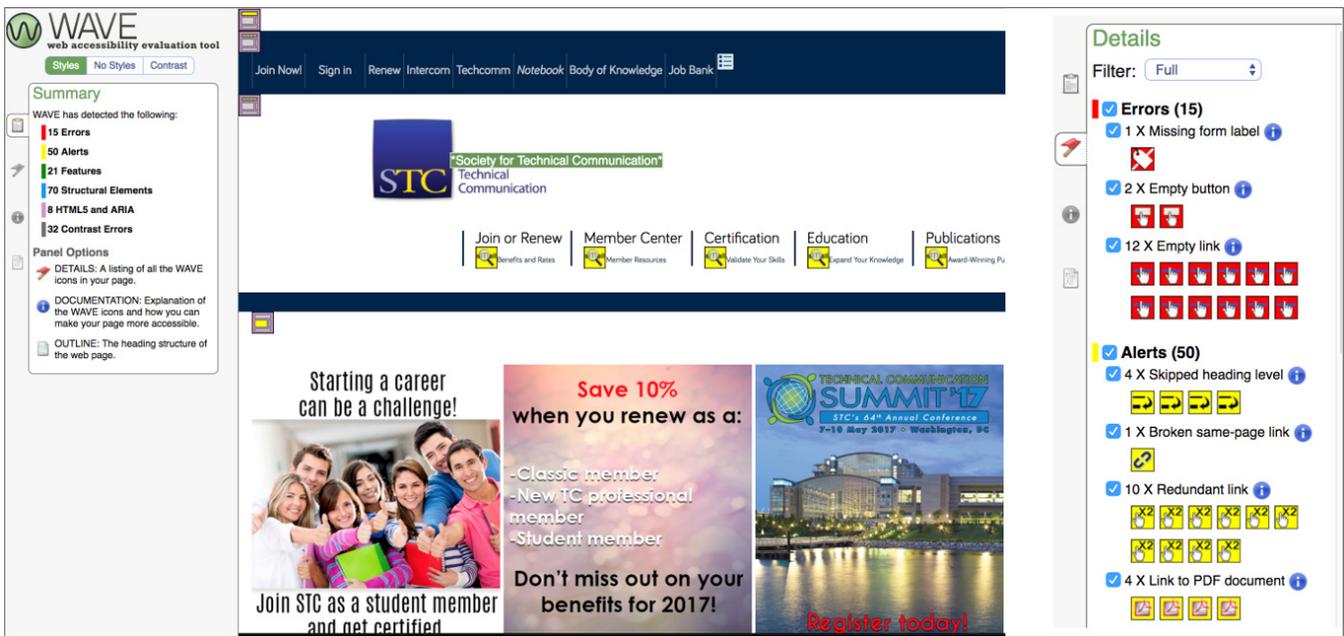


Figure 2. WAVE can help identify accessibility issues in websites

Understandable

While content must be easy to access and navigate, it must also be easy to understand.

This includes:

- Text is readable and easy to understand, avoiding jargon and unnecessary technical terminology
- Foreign languages are clearly identified as such

- Page appearance and operation should be predictable and logical
- There should be an easy way for users to identify, avoid, and correct mistakes, especially in forms

Robust

In some ways, this is one of the simplest aspects of ADA/WCAG 2.0 compliance, but can also one of the most challenging to implement. The entire purpose of robust design is to ensure compatibility and consistency across multiple platforms, and avoid potential errors or issues due to incomplete tags, or proprietary formatting tools that are not compatible with current or future assistive technologies.

Testing for ADA Compliance

WCAG 2.0 includes three different levels of conformance, A to AAA, most of which exceed the requirements for Section 508 of ADA compliance. As a general rule, if a site meets level A and some AA requirements in WCAG 2.0, it will meet Section 508 criteria, especially regarding text alternatives for images/graphics and adaptability.

To know whether a site or content meets these requirements, testing is critical. There are a number of options and tools for testing. A number of

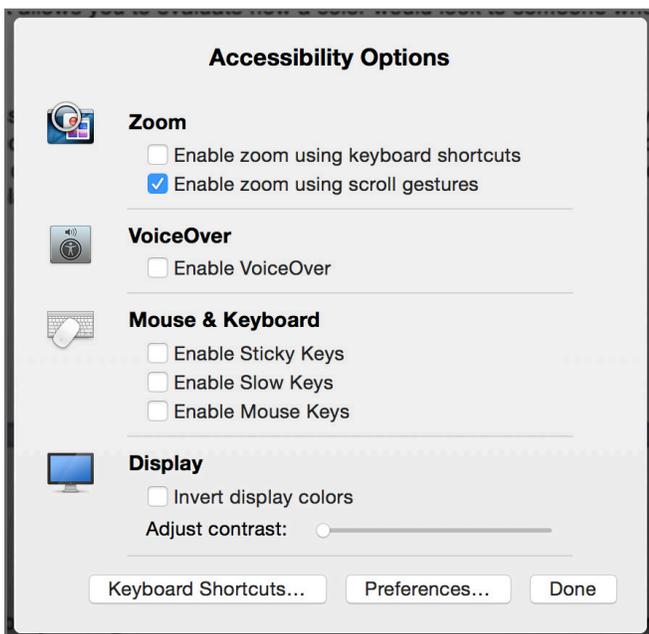


Figure 3. Basic Mac OSX accessibility options

companies provide testing services and even can assist in corrective actions to ensure that a website or PDF meets requirements. However, for most ADA compliance requirements, testing can also be done within an organization, using different readily available tools.

WebAIM.org

<http://WebAIM.org> (Web Accessibility In Mind) is a resource run by Utah State University's Center for Persons with Disabilities. It features a wide range of information, including testing tools, checklists, simulators, and more. WebAIM also provides a number of helpful services, including training, certification, consulting, and evaluation. Two of WebAIM's most helpful tools are listed below.

WAVE

WAVE (Web Accessibility Evaluation Tool) is a free web-based tool from WebAIM that automatically evaluates a webpage and provides feedback on the page, including alerts and errors regarding the style, structure, and potential issues with image contrast. It is available as an extension add-on for the Chrome and Firefox browsers, or by entering a URL on WebAIM's website: <http://wave.webaim.org/>

WAVE is a great way to check an existing website that you are adapting to meet ADA compliance standards, and determine potential problem areas.

Color Contrast Checker

The Color Contrast Checker is another tool from WebAIM that is designed to help you evaluate for WCAG 2.0 standards. You can enter two different colors and quickly determine whether they meet contrast requirements for both regular and large text on a website or in a document.

ColorHexa

ColorHexa <http://www.colorhexa.com> is a tool that allows you to evaluate colors for use on websites. It also provides you with the ability to test different shades, as well as transitions from one color to another. This can be helpful when placing two colors side-by-side to look for potential contrast issues. It also

includes a color-blindness simulator that allows you to evaluate how a color would look to someone who is colorblind.

Accessibility Tools

There are a number of accessibility tools integrated into both Windows and Mac operating systems, including tools to allow text to speech, large text, high contrast, and more. In addition, Adobe Acrobat features a number of tools for providing comparable accessibility in PDFs. By using these tools as part of your testing process, ideally on multiple platforms and browsers, you can get a good idea about potential issues.

Real-World Testing

Just like with any proper usability testing, there is no substitute for real-world users. To understand some of the challenges, have internal users test different aspects of functionality, such as navigating by keyboard only, zooming in close to test page navigation, or reviewing black and white printouts of color pages to simulate color blindness. Try navigating a page with your "off" hand, or with your hands full.

If at all possible, especially for large initiatives, it is recommended to have individuals with different disabilities test a website or content, not only to be sure that it meets the requirements as set forth by ADA, but also meets the challenges of the real world. There are several organizations that can assist with this type of testing, ranging from private companies to universities to nonprofits. You may even look within your own organization or customer base for volunteers to do testing.

Are You Ready?

Although a clear directive for ensuring that nongovernmental websites and content meet ADA-compliance standards does not yet exist, efforts are well underway at the Department of Justice. Perhaps more important, the number of lawsuits against companies such as Target, Netflix, and others, show that it's becoming increasingly expected that our online world meet the same accessibility standards as are seen in the brick-and-mortar world. Recognizing the large and growing number of people with disabilities, it also makes good business sense to

accommodate their needs as the market share they represent increases.

While ADA compliance standards does add some new challenges to the world of technical writing and web development, meeting those standards also means an improved focus on usability in general, not just for those with disabilities. There is a growing pool of resources available for developers, and it is easier and more important than ever to ensure that the content we develop meets the requirements for Section 508. With a push to bring Section 508 standards to a much broader market, and the potentially huge impact it could have on the world of communications, are you ready?

Resources

Caldwell, Ben; Cooper, Michael; Reid, Loretta Guarino; and Vanderheiden, Gregg eds. "Web Content Accessibility Guidelines (WCAG) 2.0", WC3 (11 December 2008). <https://www.w3.org/TR/WCAG20/#conformance-reqs>

None listed. "WebAIM We Have Web Accessibility in Mind." WebAIM (14 March 2017). <http://webaim.org/>

None listed. "How to Meet WCAG 2.0." WC3 Web Accessibility Initiative (12 March 2017). <https://www.w3.org/WAI/WCAG20/quickref/>

References

Dick, Wayne and Jewett, Tom. "Mapping Section 508 to WCAG 2.0." Tom Jewett Web Design, LLC (2009). <http://www.tomjewett.com/accessibility/508-WCAG2.html>.

Mitrano, Tracy. "Section 508 WCAG 2.0 Oh My!" Inside Higher Ed. (12 January 2016). <https://www.insidehighered.com/blogs/law-policy-and-it/section-508-wcag-20-oh-my>

Pan, Jennifer. "Does Your Website Need to be ADA Compliant?" Logic Solutions (28 September 2016). <http://www.logicsolutions.com/does-website-need-ada-compliance/>.

Pellicore, Sylvia. Usable for Everyone: Accessibility as UX (World Usability Day, Michigan State University, 2016).

None listed. "The Americans with Disabilities Act." Equal Employment Opportunities Commission (27 January 2017). <https://section508.gov>.

None listed. "The Americans with Disabilities Act." Equal Employment Opportunities Commission (11 February 2017). www.eeoc.gov/eeoc/history/35th/1990s/ada.html.

Author Contact Information

Earl Peters
VP User Experience
JohnsonRauhoff
2525 Lake Pines Drive
St. Joseph, MI 49085
269.930.1606

Author Biography

Effective technical communications is often more about solving problems that a client didn't even realize existed than it is about writing about a specific product. With more than 20 years of experience in the industry, Earl has worked with products ranging from small kitchen appliances to industrial manufacturing equipment. As the VP of User Experience at JohnsonRauhoff, he and his team have worked with a number of Fortune 100 companies to develop technical communication strategies that take them from traditional printed solutions to next-generation communication paths.

With a double Bachelors degree in Economics and German studies from Kalamazoo College, followed by graduate studies in Germany, Earl's love of learning and passion for foreign languages has also been a great asset in his role as a technical communicator. This, combined with several years in the retail electronics industry, has given him a strong appreciation for good technical communication. Earl currently splits his time between SW Michigan and southern California.



Pfeilsticker, Steve

Supporting Customer Advocacy with Intelligent Knowledge Management

Express Scripts is the nation's leading full-service pharmacy benefit management company, helping employers, health plans and other health-insurance payers to reduce costs while providing the best care for their members. The company saw an opportunity to implement process improvement solutions that would create synergies to enhance positive member outcomes; while improving employee experience and achieving business goals. We evaluated our existing knowledge management processes and were inspired to explore our documentation technology and strategy to achieve better business results. This led to a thoughtful and strategic evolutionary approach to aligning our Knowledge Management System (KMS) with our business goals of improved efficiency and enhanced customer experience.

The journey began with the creation of a knowledge management system infrastructure designed to enable our employees to effectively and efficiently provide industry leading customer care. As our tools and processes took shape,

Customer Advocacy and Enterprise benefits realized were:

- Increased efficiency metrics in resolving customer contacts
- Reduced Inappropriate and Unnecessary Transfers/Referrals
- Improved Quality Performance
- Expanded Business Adoption
- Reduced Operating Costs

In this proceedings paper, I will share our experience with using cutting edge tools, i.e. MadCap Flare, to develop a centralized one-stop shop Knowledge Management System, while also sharing our experiences in creating relevancy, gaining support, and expanding adoption across a large organization with a complex structure.

In addition to sharing the pathway we navigated, I will provide the “Top 10” tips and principles we used to successfully “Support Customer Advocacy with Intelligent Knowledge Management” at Express Scripts. Hopefully the lessons learned throughout our journey will help you plan yours.

Introduction

First, it would be helpful to explain the role of a pharmacy benefit manager (PBM). Express Scripts is privileged to serve approximately 3,000 clients which have more than 83 million people enrolled in their pharmacy benefits. At the most basic level, we provide point-of-sale adjudication of claims at retail pharmacies that include safety checks and payment information for individual patients who are filling prescriptions. Our home delivery pharmacy, which provides safe and convenient 90-day fills, is the nation's third largest pharmacy. Our specialty pharmacy, Accredo, offers home delivery as well as expert clinical care from pharmacists and nurses for patients with the most complex, chronic and costly conditions. The Patient Care Advocates (PCAs) who take member calls are a critical link in ensuring the

best service as well as routing calls as appropriate for the highest level of clinical advice and health care.

That said, a few years back, I was having a conversation with my manager about a new initiative at Express Scripts called “Enhancing the Member Experience”. Our company and organization was in a period of transition (with the acquisition of Medco being finalized), and the timing of this opportunity perfectly complimented the newly combined organization’s quest for building a solid foundation for our employees and members. While we knew opportunities existed to create a better “experience” for everyone, i.e. members, clients and agents, we didn’t fully understand the situation until we held a series of focus groups, conducted surveys, evaluated member satisfaction feedback, and solicited business partner feedback.

With the information in hand, we challenged ourselves to solve the problem from both ends by creating an environment of customer advocacy with better, top-of-the-line tools and documentation for our Patient Care Advocates (PCAs), while at the same time ensuring our members were receiving the level of service and quality they so rightfully deserved.

So with that said, I invite you to come along on this journey as I share with you the many tips and principles that came to light as we were able to successfully “Support Customer Advocacy with Intelligent Knowledge Management” at Express Scripts.

What is Customer Advocacy?

Customer Advocacy in itself is more of a concept, rather than something we DO, right? It’s more the by-product -- the end result -- of the things we do. It’s the way the customer is left feeling. You can provide any type of service you want, but in the end, advocacy is in the eyes and heart of the customer.

You can create advocacy by helping the customer feel:

- Acknowledged
- Valued
- Respected
- Supported
- Confident

- Represented
- Informed

All in all...it’s about how customers are left feeling as the result of their service experience.

The Conundrum

We’ve all been customers who’ve called a Contact Center at some point. So what would you say if I were to ask you, “What is the one thing you want when speaking with an agent?”

Would you say:

- “I want my question answered!”
- “I want the agent to care (show empathy) about my situation because I’m really upset or frustrated!”
- “I want an agent to go out of their way to educate me!”

I’m quite certain that if you were to think about it, it would be all of those things. Do you agree?

Unfortunately, it’s not that easy. It is a fact that there are forces in any contact center that can work against (or be a barrier to) an agent being able to spend this level of quality time with a caller. Sometimes, it can be the pressure to achieve certain metrics, whereas at other times, it could be deficiencies in their reference materials. Nevertheless, if at the end of the day the goal is to make sure that every caller is left feeling positive about their experience, it’s critical that front-line agents have the tools for success.

Our A-HA! Moment

In 2013, the idea of quality time got us thinking. On paper, it sounded great. But what would our PCAs think about it? Is it even possible with the pressures that they face? What are the barriers to achieving quality time? The only way to know was to get in front of them and ask. Therefore, we held a series of focus groups and surveys to better understand what our PCAs would need to make this happen.

It was through this direct engagement and the survey data that we discovered a recurring theme. PCAs didn’t feel successful using the Knowledge Management tools and resources available to them.

They said the existing tool was difficult to use and information was hard to locate, making it less likely to be used. In turn, this had the potential to create issues with quality and increase operating costs.

As a direct result of these focus groups, time studies, and our own observations, we had our proverbial A-HA!!! moment. We realized that we were not only supporting Customer Advocacy, but were supporting Employee Advocacy as well. Simply put, we couldn't support customer advocacy without supporting employee advocacy, and vice-versa.

The issue couldn't simply be rectified with continuous coaching. There was more to it than that. We ultimately determined that our opportunity was with the DOCUMENTS! While not glamorous or fun, documents are the key to ensuring PCAs have the latest information available. However, it's not just the documents themselves, i.e. the content. It was also about the technologies that supported the development and access to those documents.

The 10 Principles

It was at this realization about documents that we began the long journey towards "Supporting Customer Advocacy with Intelligent Knowledge Management." It led us to develop and follow 10 principles for success.

Principle 1: Get Your Bearings

In order to solve the problem with documents, first define where you are and determine where you want to go:

1. Understanding the Myth – It's important to understand the current culture. For example, is the belief that:
 - Users have everything memorized?
 - Procedures are static?
2. Define the Gaps – Look for any opportunities to streamline or centralize your process.
 - Are documents stored in multiple locations?
 - Is there a consistent style guide in place?
3. Know the Pain Points -- Know the specific factors that cause an individual person, a team, or a

business to not perform optimally. Also, think about it from multiple angles.

- Documents
 - Are the documents too lengthy?
 - Is the content written to the level of the user?
- System
 - How is it organized?
 - Is there effective of alignment in messaging between patient/customers, client, and agent?
- Process
 - Are there unnecessary referrals or transfers as a result of the documentation?

Principle 2: See Your Destination

Establish a mission and a vision for where you want to go and how you are going to get there. Brainstorm and trystorm potential ideas. Remember, no idea is a bad idea! It's all about achieving the goal at hand.

Our mission was to develop an enterprise-wide KMS that integrated context-specific knowledge management within the user's service system to improve efficiency and quality, and reduce operating costs.

We sought to achieve our mission, and dispel the aforementioned myths by:

- Building a single platform for knowledge management
- Publishing/authoring by business group
- Displaying information according to job responsibility
- Using best practices in formatting and display
- Providing just-in-time relevant search results
- Integrating within the user's system
- Using single-sourced, i.e. chunked, content in conjunction with metadata to build meaningful relationships between the content

Principle 3: Sell Your “Why”!

A solution is only as good as what you are going to get from it. What are the benefits? How will it enhance the customer experience? Will it increase efficiency? These all are great questions to ask as you go down the road of creating your KMS.

Principle 4: Understand the Technology

It’s important to have a handle on your current inventory of tools and properly assess if they will work sufficiently within the design of your new environment. If they do, that’s great because it’s one less cost and resource to incur. However, if it doesn’t fit within your new vision, and if the means are in place, look for tools that will help you achieve your vision.

For us, it was all about establishing our own footprint. Our goal was to implement a system using best-in-class tools, with the least amount of overhead. We accomplished this by creating a system with a foundation based on a mix of in-house and newly purchased systems and applications.

Principle 5: Catalog Your Content

If you don’t use metadata in your documents to influence search results, then at a minimum, catalog your content to increase efficiency and organization within your document library. Assigning metadata to your content makes a world of difference. It allows writers, archivists, etc., to more easily find specific pieces of content. More importantly, if you are using an ontology based scheme to organize content and influence search, you can use the metadata to create meaningful relationships between your documents.

Taxonomy

If you are considering going down the road of creating your own KMS, it’s important to understand the framework of what’s needed. For us, it was all about creating meaningful relationships between our content through the use of taxonomies. A taxonomy is used for classifying simple and general objects. Taxonomies are typically hierarchical in structure and are of a parent/child composition.

Ontology

Defining the taxonomy alone will not allow you to escape document silos. There is something missing. What we learned that is often overlooked in most knowledge management systems was the ontology of the content. If taxonomies are the parent/child relationship, the ontology is much more tightly knit. It has all of the parent-child relationships, plus all of your favorite cousins. A well designed ontology helps to reveal other relationships between topics. By exploring how your documents are related, you will identify meaningful aspects of the relationship and find new ways of cataloging the content.

Principle 6: Reuse is Good

Have you ever heard the phrase “Why reinvent the wheel?” Well, the same applies to content. Why rewrite content when you don’t have to? Reuse is a good thing. In fact, it’s not a good thing...it’s a great thing, it’s the right thing...I could go on and on.

When writing content, it’s very important to stay in tune with ways to repurpose (or single-source) content. Efficiently managing changes to content and ensuring the integrity of the information is key.

But on top of that, the concept of single sourcing is a core factor of efficiency. In this day and age organizations are constantly counting their pennies. As a result, processes are constantly being reviewed to ensure money is being spent wisely and resources are being properly allocated. As writers, you can be advocates of efficiency through your writing. You can take that SOP that once took 40 hours (as an example) to create/maintain and reduce the overall level of effort (LOE) by employing a single sourcing strategy.

Principle 7: Know Your Audience

“What does this mean?” Have you ever looked at a manual, procedure, or a job aid and asked that question? I’m pretty sure all of us have. It’s a proven fact that one sure way to lose an audience is not writing to their level of intelligence or aptitude. It seems very fundamental, but many times it gets forgotten. In fact, there are two big mistakes that a lot of writers make in regards to their intended audience:

The language and/or terminology in the document is too advanced or not advanced enough – e.g.

documentation intended for a non-technical audience, yet is loaded with technical terms that only an engineer would understand. The lesson is quite simple. Put yourself in the chair of the target audience and ask yourself, “Would I find this meaningful?” As a means to avoid this type of situation, consider employing a Quality Assurance (QA) strategy that ensures the content is reviewed by multiple individuals prior to release.

Documents are displaying in the KMS for people who have no reason to see them, e.g. agents are seeing documents in the KMS that are meant for supervisors only. When it comes to a KMS, there are several technical aspects to consider. For starters, is the metadata correct? If using a Help Authoring Tool (HAT), are the targets valid? Employing a QA strategy much like the one above will certainly provide the level of checks and balances necessary to ensure content is seen by the right people.

Principle 8: Avoid Overload

For those familiar with the principles of Information Mapping®, this will sound familiar. When it comes to developing your content, keeping the reader’s cognitive load in check is paramount to them being able to retain information. Too much information can cause overload, resulting in the reader missing important information, or losing interest all together.

One simply strategy we employed in our KMS to avoid cognitive overload was to make content visible or hidden based on the reader’s discretion. We accomplished this by employing a simple chunking strategy combined with features provided by the help authoring tool. Doing this empowered PCAs to view content that is relevant to them based on the call scenario.

Principle 9: Search Optimization

The search function of a KMS is what an engine is to car. If you don’t have it, it’s not going to go anywhere. In fact, our time study and focus group data showed that a good search function will mean the difference between low and high handle times, poor and great quality, etc.

So, what do you do if you recognize that your search function is less than stellar? For starters, if you have the means to do so, invest in a top-of-the-line search

appliance. If that’s not in the cards, then make optimization (with your current system) your mindset.

In other words, think of ways you can tune your search in order to address feedback like:

- Search results weren’t relevant, i.e. too many results displayed
- There was no rhyme or reason to the order of results
- Took too long to find things

Optimization is not an easy task. The change and effort to do so will be substantial. If you don’t have a metadata schema already in place, you will want to consider doing that since so many search engines can create dynamic functionality based off of it. In addition, consider ways you can promote important documents or links within your KMS. PCAs have said that having direct line of sight to these important documents has created a more pleasurable and less stressful experience.

Principle 10: Hone the Experience

The KMS is not all about the documents and the technology; it’s also about honing the experience of your users, authors, content owners, etc.

As you go about creating your own KMS, consider the following 5 things you can do:

1. Give your users a voice – Create a feedback process so users can have an active voice and share their experiences (whether good or bad) using the KMS.
2. Establish Document Ownership – Create a process so that the right people own the documentation. A technical writer may not own the material she works on. Consider linking user feedback processes directly to document owners so they also manage and respond to user comments.
3. Continuous Monitoring – Facilitate focus groups, or time studies, to ensure the system is still performing as expected. The data from these sessions will help drive any changes needed.
4. Focus on Continuous Improvement – Stay abreast of the current needs of the business to respond to any related changes. Whether it’s changes to process or even to the technology, it’s critical to

assure your users they have the most current and efficient processes, documentation, technology, etc., in place.

5. Launch a Change Control Group – Create a standing committee of Subject Matter Experts (SMEs) and Process Owners who ensure all changes requested are properly assessed and discussed prior to implementation.

Bonus >>> Principle 11: Don't be stagnant

Since our KMS launched in the fall of 2015, one thing that became very real to us an organization, and has in fact led to the creation of a brand new 11th principle, is the significant organic growth and evolution we have experienced. The KMS has truly become an integral part of our organization. In fact, the past 12 months has seen four new lines-of-business successfully integrated with the system, and we plan to integrate another ten over the next 12 months. In addition we have refined our processes to make them more lean and agile; thus fulfilling a goal to reduce overhead and provide greater speed to market. Lastly, with the integration of the KMS into the users' CRM system, users can now rely on the context-relevant results to assist when servicing callers; ultimately leading to reduced handle times and optimal quality.

So what is the lesson here? It's quite simple. Don't rest on your laurels and go with the flow. We live in a society where things constantly change. That is, what is cutting edge today is passé tomorrow. As writers, stay abreast on current writing trends and strategies and employ them whenever possible. As administrators or documentation managers, make continuous improvement a mindset. Remember, we want to give our front-end users the tools to be successful. Looking for ways to constantly evolve and grow your KMS will go a long way towards achieving that purpose.

Conclusion

An effective KMS strategy can contribute as a solution to many challenges that an organization may face. It takes courage to challenge existing processes. Looking beyond restructuring a KMS as the end product and truly connecting it to concrete business outcomes can lead to unintended positive results. Our original scope was to influence changes only for customers and call-

taking employees; however our thoughtful approach and demonstrated successes have expanded interest and implementation to several new areas across our enterprise. While building up a KMS to support our members, PCAs, and clients required a lot of time, effort, and resources, the end result certainly has paid dividends many times over for Express Scripts. As an organization, we are very excited for what next chapter in the evolution in our KMS will bring. We are very confident that whatever path we choose to go down, it will continue to serve our mission and provide a best-in-class KMS that supports both customer and employee advocacy. That said, I sincerely hope that you found the information in this paper to be interesting and potentially useful in your organization as you pursue your own KMS dreams.

References

Information Mapping Demo. (n.d.). Retrieved April 03, 2017, from http://www.informationmapping.com/demo/?page=did_u_know

Author Contact Information

Steve Pfeilsticker
Sr. Business Systems Analyst
Express Scripts, Inc.
6625 W. 78th Street
Bloomington, MN 55439
952.210.7778

Author Biography

Steve Pfeilsticker is a knowledge management and technical communications professional with 17 years of experience in technical writing, web/system development, and project management. He is currently a Sr. Business Systems Analyst at Express Scripts in Bloomington, Minnesota. For the last three years, his primary focus has been on developing and managing the enterprise-wide internal Knowledge Management System.



Prabhakar, Rahul

How Social Media Can Be Part of Your Set of Technical Communication Skills Tools

With the emergence of social media in the last decade or so, there has been a paradigm shift in consumer behavior, who no longer rely on conventional advertisements before making a purchase. Consumers these days would rather rely on a trusted source within the online or social community who can influence their purchasing decision by providing a true, holistic view of the product or its features. Brands realize the kind of reach and engagement social media influencers are known to have and thus want to work with them in order to champion content about their offering, drive engagement around their brand, and push traffic to their websites or landing pages. Technical communicators all over the world are known to have a range of skills or abilities that make them successful at the workplace. By being the best communicators within the organization, they can leverage third-party social media tools like Blogs, Facebook, Instagram, Twitter, etc. to create a communication channel with customers, potentially influencing their purchasing decision in the process. This paper focuses on how social media can be part of your set of technical communication skills.

The Impact of Social Media

According to a case study by the digital marketing agency ODM Group, 74% of consumers are more likely to buy when referred from a blog or social media.

A survey from Influence Central states that as many as 86% of women shoppers refer to social media before buying something and are more likely to buy a brand they've never purchased when they interact with that brand on social media.

Social media in today's context is:

- All about real-time connectedness, collaboration, and mobility
- Larger conversations within the community
- Free of cost; you only pay the service provider for using the Internet on your devices
- Global and ubiquitous in nature

The impact of social media on businesses today is unprecedented, and with influencers coming into the picture, marketers are constantly buzzing with talk about ways to engage them, mostly with the intention of driving value for brands.

Influencer Marketing

Nielsen Catalina Solutions in partnership with TapInfluence and WhiteWave Foods released a case study last year that proves Influencer Marketing delivers 11 times ROI over all other forms of Digital Media.

According to a Tomoson study, marketers rate influencer outreach as the fastest-growing online customer acquisition channel, beating organic search, paid search, and email marketing.

Influencer marketing is a type of marketing that focuses on using key leaders to drive a brand's

message to the larger market. Rather than marketing directly to consumers, you instead inspire/ hire/ pay influencers to get out the word for your brand.

An influencer is a person who is:

- well-connected
- regarded as influential and in-the-know
- looked to for advice, direction, knowledge, and opinions

Typical examples of influencers are people like CEOs, CMOs, PR professionals, creative people, bloggers, journalists and other experts.

Social media influencers usually are professionals that live and breath via their over-populated social media accounts, while focusing on very specific niches. They can coin terms, spread news like fire and influence other people's opinions.

From a brand's perspective, there are three main reasons for working with social media influencers:

- Brand Exposure
- Direct Sales
- Future Collaborations

The biggest difference between brand influencers and brand ambassadors is money. Brand ambassadors are hired to promote a brand; they get handsomely paid for what they say/do. For example, a celebrity with a huge audience that retweets everything they post is not an influencer until they start talking about a brand.

Brand influencers, on the other hand, show their love for the brand voluntarily; they may or may not get paid. Whether an influencer's audience is small or large, an influencer can reach consumers via their blogs and social networks that a brand may not be able to.

Using social media influencers in your marketing is the practice of building relationships with the people who can build relationships for you.

The most important question, however, is that what does an influencer mean for a brand? Typically from a marketer's perspective, it is crucial to know how many influencers are talking about their brand. They should be able to measure all influential posts that mention their brand each day.

To understand who qualifies as an influencer, you need to first learn the core concepts that will help you develop an influencer campaign with real results.

Defining Influence

In order to define influence, think of what aspects give an individual the power to influence others within their social sphere.

The potential to influence can be related to three attributes that are important for marketers to think about: Relevance, Reach, and Resonance:

- **Relevance:** Creating content that is relevant to your brand, or relevant to a topic that is important to your brand
- **Reach:** The ability to reach a targeted audience that is valuable to your brand
- **Resonance:** The proliferation or engagement with relevant content by an audience that is valuable to your brand

All three qualities are needed for a person to be considered an influencer for a brand.

To become an influencer, an individual doesn't always need to reach a large audience; sometimes reaching a small yet niche audience can be just as valuable. Another thing to keep in mind is that just because someone is influential on one topic doesn't mean they can be considered influential on another. For example, if someone is an influencer on all things technology, it doesn't mean that their posts about automobiles will carry the same weight.

Just like within our own personal networks, we identify individuals who are experts on a given topic, the same holds true for social media as well.

How to Engage Influencers

Most marketers believe that they're already engaging with social media influencers, and would probably seek to grow the engagement.

Typically, brands have a process in place to respond to influencers who mention them.

There are many ways to engage influencers such as:

- Encourage influencers to create original and valuable content for the brand
- Familiarize influencers on advances the brand is making in the market vis-à-vis competition.
- Entice influencers to follow social media accounts of the brand and regularly champion brand content

The degree to which an influencer is likely to advocate a brand can be determined by their social behavior (for example, what they post and how they interact with the brand).

The best and safest way to engage new influencers is by starting small; make them aware of product announcements and check their response, before calling on them to support the brand content and campaigns.

How to Identify Influencers

Before you start proactively engaging with influencers, it is necessary to identify the right ones first. This is where it can get tricky to expand the list of influencers beyond a group who are already talking about your brand.

The 'right' social media influencer must:

- Be an expert in a specific niche
- Be able to produce a variety of content, such as
 - Text
 - Videos
 - Photographs
 - Infographics
 - Slideshares
 - Rich Media
- Be able to communicate long-term in their chosen niche
- Be honest, even when the brand tries to dictate terms
- Have a good reach or engagement
- Learn when to back out if creative freedom is not given
- Share what they love

- Always add value for readers
- Not consider popularity at the cost of credibility
- Always be one step ahead in terms of knowing what the next big social media platform or product is going to be like
- Consider the transformed media landscape to produce content that is global and ubiquitous in nature
- Be aware that readers can now talk back using social media and are no longer disconnected with each other
- Make the best use of social media; the content should be original, not duplicate and be created to convene not control.

A good starting point to identify influencers is by diving directly into the data you have access to. For instance, start with Twitter - the primary channel for syndicating content from a wide variety of sources, especially blogs that contain more in-depth, thought leadership, or original long-winded content.

By querying a keyword or a collection of keywords, you will find influencers relevant to a given topic. You will also see how often individuals posts are pulled into that topic to determine their degree of relevance for the topic you're interested in.

With relevance established, you can determine how frequently those on-topic posts are engaged with to establish resonance. You can check whether that content actually gets shared.

By shortlisting potential influencers, you can also analyze the quality of their follower network to make sure they are a valuable audience for the brand. Audience quality for potential influencers plays into reach. If your topic is niche, you might be fine with accepting influencers who have a smaller number of followers. But if you're looking to drive brand awareness, you should be selective and only accept influencers with extremely large follower audiences.

If the devil is in the details and the details are in the data, that's where informed marketers should also be.

Can Technical Communicators become Influencers

With the improvement in hardware, the cost of creating content has come down drastically. Amateur content creators are ruling the roost on social media.

Here's what you as technical communicators can do.

- Find, follow and study social media influencers in your chosen niche/domain
- Start power networking through:
 - Events/Meetups/Conferences
 - WhatsApp/Facebook groups
 - Agencies (both Digital or traditional PR)
- Register with influencer marketing companies for paid campaigns
- Anytime you do a sponsored post, make sure you set aside a budget for Facebook ads and other social media promotion to give it a boost.
- Learn and master relevant social media platform(s)/ marketing strategies / content creation
- Build a loyal following over hours of writing/ editing/ sharing/ emailing/ chatting
- Intentionally design and plan out content ahead of time
- Stage the product in correct light, appropriate backdrop, and carefully designed flat lay
- Take multiple photos to get the right one (or pay to hire a photographer to take them). Edit and crop said photos.
- Craft the perfect caption and hashtags to maximize engagement
- Post at the perfect time to optimize reach/ influence/ interaction
- Follow up with readers and respond to their comments/ questions

Earning as Influencers

Even before you think of earning as influencers, design a blog media kit that contains details about all your social media handles with reach and engagement.

The quality of work and deliverables for a brand will be much higher if the influencer they're working with is receiving compensation.

Here are some tips to earn as influencers.

- Do paid campaigns only as a rule of thumb
- Not all campaigns are paid though; be open for barter/exchange
- Join an Affiliate Program that offers commission-based compensation to bloggers or influencers; get paid based on the success of the campaign i.e. per sale generated through you.
- Place advertisements on your blog

Here are some tips for brands as well.

- If you're over there throwing out offers for a product that no one's really heard of and offering no compensation in return for promoting it, there is a super fat chance word gets around and starts to damage relationships with bloggers that could really help you promote your brand all before they even start.
- If you want to build long-term relationships with the bloggers you use to promote your products (hint: you really, really do), then starting off on the right foot by offering reasonable compensation is your best plan.

Not only does this build a good working relationship, it means that as that blogger grows and continues to promote your product/ service, you grow with them.

Summary

User-generated content is slated to grow; however, quality and moderation will remain the two differentiators.

If you're thinking of becoming an influencer, stay exclusive, original, and honest.

Create a message that people actually want to hear and share. Whether it's in the form of research, infographics, surveys, fun facts, etc. make sure your message is something unique and interesting.

From there you will be able to steer your influencer strategy ship in an awesome direction.

Resources

Smitha, Nate, eds. "How to Define, Identify and Engage Social Media Influencers For

Your Brand." Simply Measured (2 April 2014). <http://simplymeasured.com/blog/how-to-define-identify-and-engage-social-media-influencers-for-your-brand/>.

References

DeBroff, Stacy, and Alexander, Lauren, eds. "Social Upends Traditional Media in Driving

Shopping Purchases." Influence Central. <http://influence-central.com/social-upends-traditional-media-in-driving-shopping-purchases/>.

Smith, Erin, eds. "TapInfluence and Nielsen Catalina Solutions Launch Collaboration for

CPG Brands to Measure Sales Impact of Influencer Marketing Campaigns." Tap Influence

(5 April 2016). <https://www.tapinfluence.com/blog-tapinfluence-and-nielsen-catalina-solutions-launch-collaboration-for-cpg-brands-to-measure-sales-impact-of-influencer-marketing-campaigns/>

"Influencer Marketing Study." Tomoson Blog. <http://blog.tomoson.com/influencer-marketing-study/>

He blogs at "When the Muse Strikes!" to help young people get info on new travel and culture stories from around the globe, including tidbits on technology, product launches, lifestyle, food, events, and much more (see <http://www.rahulprabhakar.com>).

Author Contact Information

Rahul, Prabhakar

Technical Communicator and Blogger

496 Housing Board, Saraswati Vihar,

Gurgaon, Haryana, India - 122002

91.9958750992

prabhakar.rahul@gmail.com

Author Biography

Rahul Prabhakar (37) is essentially a travel enthusiast from India whose love for the folklore took him to over ten countries, including South Korea where he stayed with his family from 2005 to 2012.

Rahul has over 15 years of technical communication experience and is a leading member of the global technical writing community. He has a Bachelors of Technology degree in Electrical Engineering and an Advanced Masters in Management of Global Enterprises.

Rahul has worked for some of the world's leading technology companies such as Samsung, Oracle, etc.



Prentice, Scott

Regular Expressions for Tech Writers

Regular expressions (regexes) may be one of the most powerful but least utilized technologies available to authors. Most authoring tools provide a Regex option in the search dialog, but when did you use it last? A regex is a special syntax that matches on a pattern of characters. Regexes can be quite complex, but a simple regex can do amazing things. What's nice is that you can learn this language slowly on an as-needed basis, and it's [mostly] not tool-specific. This paper shows real regexes that you can use in your daily work to potentially save considerable time from tedious search and replace operations. You'll see how to use regexes in various authoring tools as well as in scripting such as JavaScript, ExtendScript, or Perl, when some really heavy lifting is required.

Disclaimer: This information is not intended to be exhaustive or complete. It discusses regular expression features that may be most useful to technical writers and is designed for beginners or infrequent users. (However, some advanced topics are discussed.)

A regular expression, also known as a “regex,” is a text string that describes a search pattern. This is way beyond wildcards, but conceptually can be thought of as wild cards on steroids. Wild card search/replace is fine for simple patterns, but for more complex matching, you need a regex. In addition to a search pattern, a regex may also define a replacement string. The replacement string may contain content extracted from the search match, and may be replaced in a different order from which it was extracted. Fundamentally, a regex is a mini programming language that you can use to process your content.

You may be surprised at all of the places you can use a regex. Many authoring tools provide some form of regex support. Most “serious” text editors will have fairly robust support for the regex syntax. Scripting languages like Perl, PHP, JavaScript, Python, and Ruby all support regex processing, as well as Unix utilities like `grep`, `sed`, and `awk` (some also available on Windows). Programmers can make use of regexes in compiled programming languages like Java, C#, VB.NET. Anything with a “regex engine”!

The main benefits of using a regex are:

- Powerful searching
- Complex string replacements and intelligent modifications
- Provides a powerful syntax in very few characters
- Text format conversions

Text format conversions can be a big benefit to those in techcomm. How often do you get a file from the development team that's in XML, JSON, or CSV (or some other ASCII format) that you need to repurpose into something useful to ordinary humans? With some knowledge of regex syntax, you can open that file up in a text editor and after a few simple regex passes, you'll have something you can copy and paste into your documentation or send to colleagues for review.

As with all powerful tools, there can be problems. At first glance, regular expressions can appear very complex and overwhelming. Don't let that deter you from trying to understand the syntax! Regex syntax can vary from tool to tool based on the underlying “engine” and implementation. Learn the basics and read up on your tool's specifics and you'll be fine. Watch out for “greedy” matches; by default a regex will match as much as possible, and sometimes you'll end up deleting large chunks of content unexpectedly. Always have a backup, or just press CTRL+Z!

With regular expressions, there is no “one right way” to do something; if it solves the problem at hand, you did it right. You may hear some people say that you shouldn’t parse XML with a regex. There are limitations and possible issues, but as long as you have a “known” model and keep focused on the goal, it’s typically fine.

Regex Syntax

A regular expression is just a series of characters that make up a match pattern. The simplest regex is just a single character or a string of characters. Unless one of those characters has a “special” meaning, you’ll just match on that exact series of characters.

Examples of literal characters — **z**, **zorch**, **FOO**, **F00**

A class of special characters called metacharacters is a combination of a backslash and a character. These are shorthand ways to match on groups of characters. The **\s** matches on any “whitespace” character (a space, tab, newline or carriage return), the **\S** matches on anything that’s NOT one of those. The **\w** matches on a “word” character (basically, non-punctuation characters), and the **\W** matches on anything that’s NOT one of those. The **\d** matches on “digit” characters (0-9), and **\D** matches on anything that’s NOT a digit.

Another group of special characters will match on anchors or boundaries. The **^** (caret) matches on the beginning of a line, and the **\$** matches on the end of a line. The **\b** matches on a “word boundary” and the **\B** matches on a location that’s NOT a word boundary. These all define a location in a string, but don’t match on a specific character.

Use quantifiers to specify “how many” of the previous character, class, or group. The ***** (asterisk) indicates zero or more, the **+** (plus) is one or more, and the **?** (question mark) is zero or one. Specific quantities can be defined with the curly bracket syntax, where **{2}** says to match on exactly two, **{3,5}** says to match on three through five of the item, and **{3,}** says to match on three or more.

A syntax is provided to define groups of patterns. The **.** (dot or period) matches on any single character (except **\n** or **\r**, unless multi-line mode is enabled, to be explained more later). Parenthesis can be used to mark a pattern for capture (also explained later), and patterns within the parenthesis may be

separated by vertical bars, which will match on any of the defined patterns. Characters may be enclosed in square brackets which will match on any one of the characters. To match on a range of characters, use the hyphen within square brackets, the pattern **[a-r]** will match on any character in the range of (and including) **a** through **r** (lowercase). You can negate a character class by including a caret immediately after the open square bracket.

Examples of grouping — **(this)**, **(this|that)**, **[abc]**, **[a-r]**, **[^a]**

Modifiers

Modifiers allow you to define how the match or replacement is made or how the pattern is constructed. Some tools will offer these as options in the UI, while others will require that you use a special method to pass in the instructions. And still other tools won’t support some or all of the modifiers.

Common modifiers (options):

- g** - global replace
- i** - case insensitive match
- m** - multiline mode (treats each line separately)
- s** - single-line mode (“dot matches all”, includes **\r\n**)
- x** - free-spacing mode (comments follow “#”)

If a tool doesn’t offer a modifier as an option in the UI, you can try the inline syntax. Place the modifier in parenthesis where a question mark follows the open paren. For example, **(?imsx)** enables all of the modes, and **(?-imsx)** disables the modes. In practice, you’d likely just use this to enable or disable multiline or single-line modes, as in **(?s)**.

Basic Examples

Find the word

- .. “cat” (lowercase) — **\bcat\b**
- .. “cat” or “dog” (lowercase) — **\b(cat|dog)\b**
- .. “Cat” or “cat” — **\b[Cc]at\b**

- .. “cat” followed by numbers — `\bcat[0-9]+\b`
- .. that contains “cat” — `\Bcat\b`
- .. that starts with “cat” or “Cat” — `\b[Cc]at\b`

Naturally “Greedy”

As mentioned earlier, regular expressions are naturally greedy and will match on as much as possible to fulfill the requested pattern. You’ll need to add code for a minimal match. Use the `?` character to turn a greedy match into a minimal match.

The following example can be read as “match on “this” followed by anything up to the first “that”:

```
this .*? that
```

Or, use character negation to tell the pattern to match on anything until you run into “one of these” (often used with HTML/XML matching). To match any character except the close angle bracket, use this:

```
[^>]+
```

Or, enable multiline mode (if possible) to process each line separately so things can’t get too out of control (most tools default to this mode, so not typically necessary).

```
(?m)
```

Capture Groups and Backreferences

If you want to match on a pattern, then continue matching on what was matched, but you don’t know ahead of time what that might be, you need to use a capture group with a backreference. Parenthesis define the capture group, and matched content within the group is passed to a numeric backreference.

Find any word followed by the same word (dog or cat):

```
(\w+)\s+\1
```

This pattern first matches on a group of one or more “word” characters (basically, matching on a word), followed by a match of one or more whitespace characters, then it matches on whatever was matched in the first (`#1`) capture group. Effectively finding cases where you have “dog dog” or “cat cat”.

Attributes in HTML or XML may be wrapped in single or double quotes. If the content you’re processing uses either format, use a backreference so it’ll match in either case:

```
class=(["']).+?\1
```

This pattern first matches on the literal string “class=”, then the capture group defines a character class of double or single quote (this will match on either one). That’s followed by a “dot plus,” meaning “match on one or more of anything,” followed by a question mark, making the match “minimal.” Then it’s terminated with the backreference to the type of quote that started the attribute match.

Tools will use either `\1` or `$1` (*backslash-number* or *dollar-number*) as the backreference syntax to indicate the captured string. You just need to know which format your tool uses. To get the “number,” count the left parens from the beginning (excluding non-capturing groups like modifiers).

Date matching

Matching and reformatting of date strings is a common situation that you may run into. Regexes are great for that!

Use the following regex to match the date in the form of `yyyy-mm-dd` or `yyyy/mm/dd`:

```
\b\d{4}[/-]\d\d?[/-]\d\d?\b
```

This pattern says: start with a word boundary, then match on exactly 4 digits, followed by a slash or hyphen, followed by 1 or 2 digits (the `?` makes the second one optional), followed by a slash or hyphen, followed by 1 or 2 digits, then terminating with a word boundary.

Because the latter part of this pattern is the same (slash or hyphen followed by 1 or 2 digits), you could make this more efficient as follows:

```
\b\d{4}[/-]\d\d?{2}\b
```

To change format of a date string from `yyyy/mm/dd` to `mm/dd/yyyy`, use this as the match string:

```
\b(\d{4})[/-](\d\d?)[/-](\d\d?)\b
```

and this as the replacement string:

```
$2/$3/$1
```

HTML/XML Examples

Some common patterns for processing HTML/XML content are shown below.

Extract the element name to **\$1**:

```
<([\w-]+)[^>]*>
```

This pattern says: match on an open angle bracket, followed by a group that is made up of a character class matching on one or more word characters or hyphens (valid element name characters), followed by a character class of zero or more characters that are NOT a close angle bracket (basically, anything else in the open element tag), followed by the close angle bracket. The element name will be available as the **\$1** backreference.

The following examples build on this logic and previous examples.

Extract the value of the @class attribute value to **\$1**:

```
<[\w-]+[^>]*class="([^\"]+)"[^>]*>
```

Extract the content from the element to **\$2**:

```
<([\w-]+)[^>]*>(.*?)</\1>
```

Where to Start?

Getting started with a new tool or concept can be difficult. As with most things, start simple, really simple, just get used to your editor by trying simple matches on some literal characters. Then try some matches on strings of a specific length or pattern. Once you feel comfortable with that, try extracting and replacing portions of strings. Think about situations where you need to extract portions of content from files you receive from others. Start simple and get comfortable with the basics before moving on. If you work with HTML or CSV files, use a text editor and match on some of that code.

Tool-specific Issues

Techcomm tools you're likely to have on hand may include the following:

- Adobe FrameMaker
- Adobe RoboHelp
- Microsoft Word
- MadCap Flare
- Oxygen XML
- Text editors and scripting languages

These tools all support regular expressions to varying degrees. Text/code editors are typically line-based while authoring tools are more often going to be paragraph-oriented (paragraph is the same as a "line"). Keep that in mind when crafting your regexes. The default mode for a tool may be single-line or multiline mode, it's good to determine that before doing much experimenting. Not all modifiers are available in all tools (try the inline syntax if no options are provided). Most tools will use the **\1** backreference syntax (if backreferences are supported), but some will use **\$1**.

FrameMaker

Support for regular expressions in FrameMaker is different in structured vs. unstructured documents.

In unstructured documents, the default is multiline mode (one paragraph at a time), but you can force it to use single-line mode (treat the entire document as a single line) by using the inline modifier (**?s**). To match on the EOL (end of line) use the **\n**, and to match on a line break (SHIFT+ENTER) use **\x09**. Use **\t** or **\x08** to match on a tab. To replace a linebreak use **\r** or **\x09**, and to replace a tab use **\x08**. Use the **\$1** format for captured replacement value.

In structured documents, the default is also multiline mode (sort of), and there is no support for inline modifiers, so it appears that there's no way to search or process more than one line at a time. (I say "sort of" above, because it's really not true multiline mode, in fact it's just able to search content within one element at a time.) To match on an EOL, use **\n**, but if that's outside of node (which it usually is) you can't use this to match on a specific word at the end of a line. You'd need to use **word\$** to match a word at the end of an element. Use **\x09** to match on a linebreak and **\t** or **\x08** to match on a tab. To replace a linebreak use **\r** or

`\x09`, and to replace a tab use `\x08`. Use the `$1` format for captured replacement value. If you're working in the XML View, use the "Complex Expressions" option for regular expressions, although this will have limited features.

RoboHelp

Support for regular expressions differs between design view and code view. In design view, single-line mode is the default and multiline mode is default in code view. Inline modifiers are not supported, and capture group replacements are also not supported. Uses "Microsoft-style" regular expressions (not really sure what that means). Newline (`\n`) only matches in code view. You can use regular expressions with the find/replace in files feature.

MS Word

Word uses a "special" MS hybrid regex/wildcard syntax (aka, not "real" regular expressions). The `*` matches anything except EOL (non-greedy), and `@` after a char or char class matches one or more of those characters. Use `^13` to find a paragraph mark and replace with `^p` (replacing with `^13` can cause problems). To find duplicate paras use this syntax: `(*^13)\1`, and to find duplicate "words" use this: `(<[a-zA-Z0-9]@>)\1`.

Flare

With Flare, it's best to use regexes in code view; they seem unreliable in XML Editor view because the search is done on underlying code, which may not be what you're trying to do. The default is multiline mode (one line/paragraph at a time) and inline modifiers are not supported. Use the `\1` format for captured replacement value. Flare offers support for regular expressions in their find/replace in files feature.

OxygenXML

As with other tools with multiple views, regular expressions work slightly differently in code vs. author view with Oxygen. In author view, it seems that a match can't span block-level elements, but in code view it operates exactly as you'd expect. The default is multiline mode (one line at a time), and you can enable single-line mode with the "dot matches all" option (doesn't appear that inline modifiers are

honored). Use the `\1` format for captured replacement value. Oxygen supports regular expressions in find/replace in files, and has extensive options for performing global (project-wide) replacements, such as all files in a DITA map or all files in an EPUB.

Text Wrangler

Text Wrangler is a free, reasonably full featured, text editor for the Mac. Use the "grep" option to perform regex search/replace actions. Default is multiline mode, enable single-line mode with the inline modifier `(?s)`. Use the `\1` format for captured replacement value. Text Wrangler supports regular expression find/replace in files.

Scripting with regexes

Many programming/scripting languages provide regular expression support. This lets you develop scripts to perform batch processing of files as well as perform complex (or simple) string replacements to modify or validate user input. If you find that you're doing the same regular expression in your authoring tool, perhaps it makes sense to move that into a script. Perl and Python are popular scripting utilities, that can be used by beginners as well as advanced developers. JavaScript provides regex support that can be used to augment interactive website applications, and ExtendScript (based on JavaScript) can be used to automate processing in FrameMaker and RoboHelp (as well as other Adobe products).

Use the JavaScript `search()` method to validate or locate a string with a regular expression. It returns the position of the match (or `-1` if no match). The following example returns the position of "STC" (case insensitive) in the string:

```
var str = "Welcome to STC Summit";
var pos = str.search(/STC/i);
```

The JavaScript `replace()` method performs a string replacement and returns the new value. The following example replaces "STC" (case insensitive) with "The" in the string:

```
var ret = str.replace(/STC/ig, "The");
```

You can use similar methods in ExtendScript. The following example strips the full path and file name down to just the “name” (strips the “.fm”):

```
var doc = app.ActiveDoc; var filename = doc.  
Name.replace(/^.*?([\^\]+)\.fm$/i, “$1”);
```

Regular expressions are tightly integrated into the Perl language. It’s great for quick batch processing scripts and is platform independent. The following example will locate the word “cat” in the string and perform some (unspecified) action (note the “m” operator for matching and the “i” modifier for case insensitive):

```
my $str = “The cat chased the rat.”;  
if ($str =~ m/\bcat\b/i) { ... }
```

The following example replaces the word “cat” with “dog” in the string (note the “s” operator for search/replace and “g” modifier for global, no “i” so it’s case-sensitive):

```
$str =~ s/\bcat\b/dog/g;
```

Summary

This has been a very brief dip into the regular expression pool. Remember that regexes aren’t just for geeks or programmers; they can be a very powerful tool for authors as well. Don’t be overwhelmed by the syntax, just start simple and work up as needed. Simplify your tasks through automation, and save time for the fun things!

Resources

RexEgg, www.rexegg.com

Regular-Expressions.info, www.regular-expressions.info

Friedl, Jeffrey E. F., *Mastering Regular Expressions* (O’Reilly), 1997.

Author Contact Information

Scott Prentice
Chief Leximator
Leximation, Inc.
122 H Street
San Rafael, CA, 94901
415-485-1892

Author Biography

Scott has been working in the techpubs field for 25+ years. He focuses on custom online help and EPUB development, FrameMaker (plugin and structured application) development, as well as custom web application development. He is very involved with DITA and created the DITA-FMx plugin for FrameMaker.



Proff, Allie

My Android Dreams of Electric Cats: Are You Capturing Your User's Emotive Analytics?

Analytics and metrics provide a concrete way to visualize and track data about our content and customers. New technology gives us the ability to track our users' emotions, giving us fresh insight into ways we can improve both products and content.

Metrics are powerful. Not only do they make it possible to measure abstract trends such as helpfulness and comprehension, but energy flows where attention goes. Sometimes, just tracking a metric affects the business aspects which you choose to focus and grow. [1]

Advances in artificial intelligence (AI) and machine learning, the ability to analyze big data, as well as biometric developments in wearable technology and voice and face recognition give us new and evolving ways of analyzing users' emotions. Feelings lie behind every decision, even seemingly unemotional ones [2]: they drive user experience (UX) and customer experience (CX) design, product sales, viral content, brand loyalty, and customer satisfaction. Companies who add emotive analytics to their portfolio will truly gain the edge to get results.

What Is Emotive Analytics and How Does It Work?

Analytics uses mathematics, statistics, machine learning, and predictive modeling to find meaningful patterns and knowledge in recorded data.

Emotive software gathers input from a user's voice, face, body language, or other biometrics such as respiration and heart rate, and then determines if the person is happy, sad, angry, disgusted, or another feeling. Once feelings are identified and collected,

the analytics side takes over and starts collating and interpreting the data for output.

Measurements can look at one person over time, one person compared to a crowd, or even analyze an entire crowd. Jaguar Land Rover and Mindshare gave wearables to selected spectators at the 2015 Wimbledon Championships. The biometric and sociometric data such as crowd movement and audio levels provided 45 million data points every day that provided real-time mood analysis used for marketing and communications. [3]

Other terms for emotive analytics:

- Emotional analytics.
- Emolytics.
- Affective computing.

Related terms:

- Cognitive computing: simulation of human thought in self-learning systems.
- Big data: extremely large data sets analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions.
- Machine learning: a type of artificial intelligence that has the ability to learn without being explicitly programmed.
- Artificial intelligence (AI): computer systems able to perform tasks that normally requires human

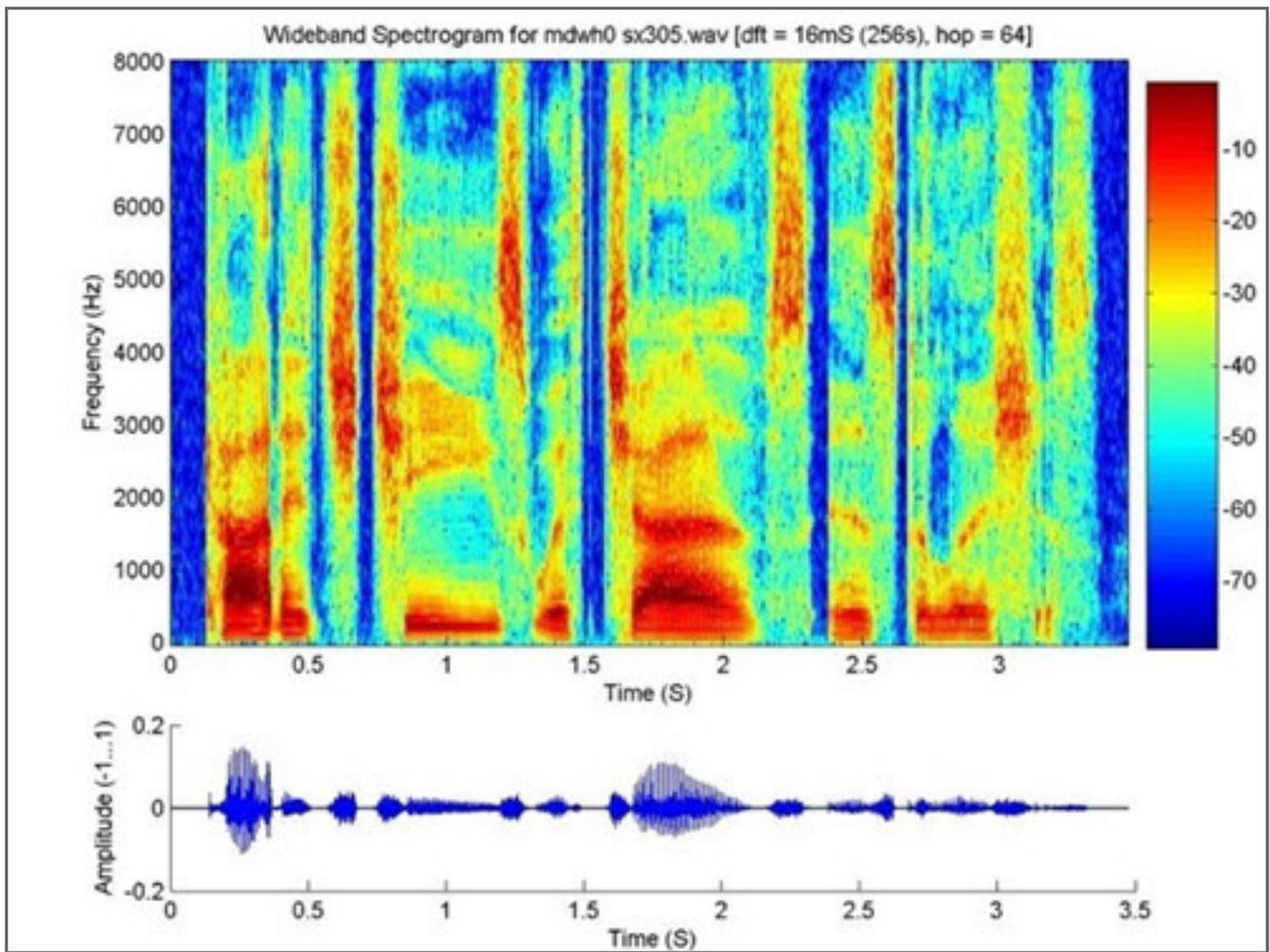


Figure 1. Spectrogram (Top) and Amplitude Waves (Bottom) for the Phrase, "Cottage cheese with chives is delicious."

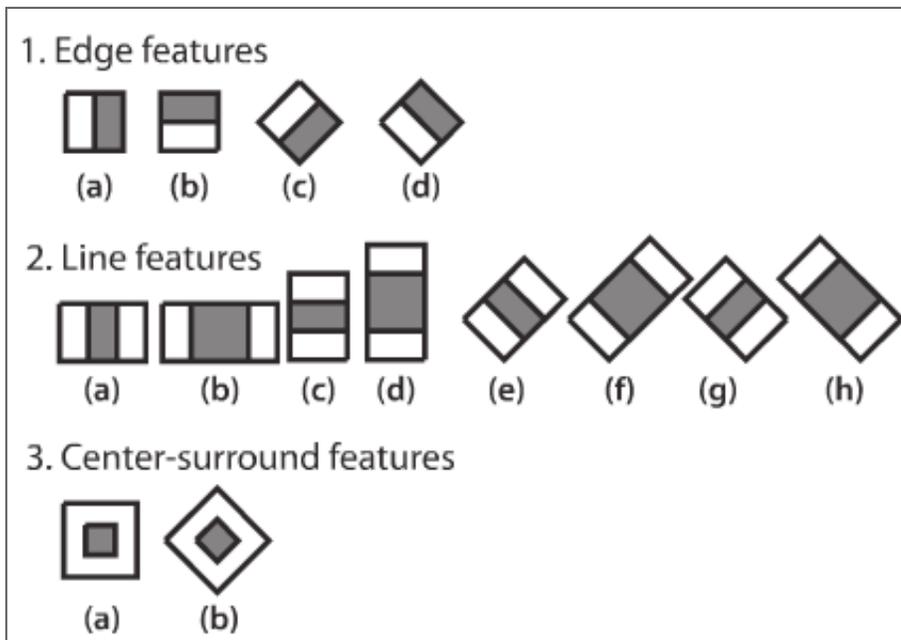


Figure 2. Various Haar-like Features..

intelligence such as visual perception, speech recognition, learning, and decision making.

Voice Recognition

Speech software transforms the analog recording into a digital signal. The audio to digital converter (A/D converter) uses a mathematical process called Fast Fourier Transformation (FFT). The image we usually associate with recordings shows amplitude only, but the digital signal is also broken into a spectrogram of the intensity of various frequencies.

The spectrogram is then broken into bits of sound called acoustic frames, analyzed for phones (pieces of words), and then reassembled like beads on a string.

A spectrogram not only shows phones, but can also be analyzed for pitch (is the speaker an adult or a child?) and other categories such as timbre, intensity, breathiness, and more [4].

Facial Recognition

Facial recognition is a term that can be used for both facial detection (is there a face?) and facial identification (whose face is this?). In this case, facial recognition is synonymous with facial detection.

A still image or video frame is broken into squares of pixels: first larger squares and then smaller squares. The pixel brightness in a square is averaged and compared against other squares, looking for patterns called Haar-like features. The overall process, called the Viola-Jones object detection framework, analyzes series of Haar-like features, takes only a fraction of a second. Some algorithms use color and movement (video) as well.

If you had a frontal view of a person's face, a large analysis might see the person's two eyes (dark areas) and bridge of the nose (a light area) as Haar feature 2a. It doesn't matter if the feature is dark-light-dark or light-dark-light, as long as it fits the general pattern.

Once the software has detected a face, it can then map points corresponding with eyes, nose, mouth, and cheeks. Different software will map different points,

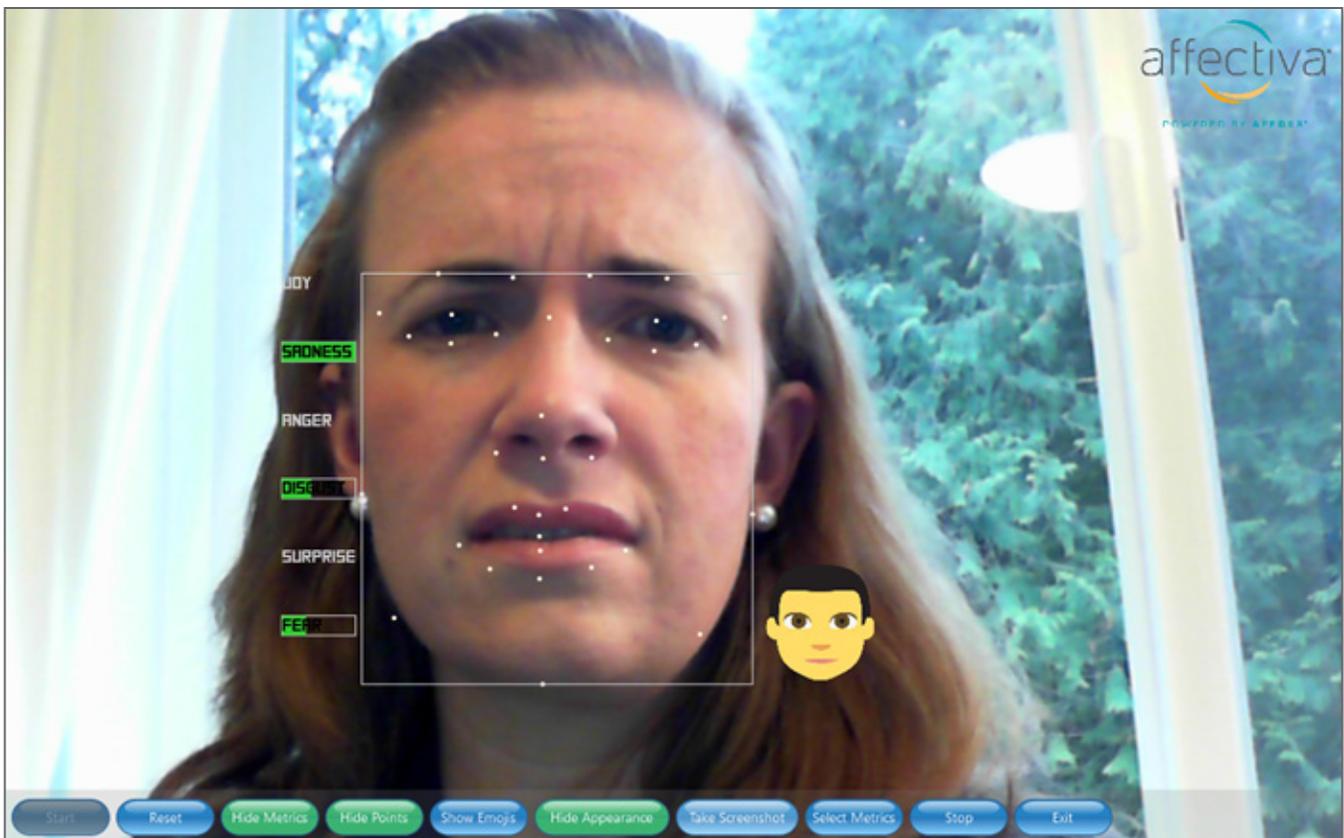


Figure 3. Affectiva Analysis of the Author's Face Showing Sadness, Disgust, and Fear.

but the relative position and movement of points will help determine emotion [5].

Emotive Analytics In Use

Even in 2013, Gartner [6] identified affective computing as a rising field, a trend that appears in the table of contents for the 2016 Hype Cycle for Emerging Technologies [7] and Mobile Technologies [8].

Amazon, Apple, Facebook, Google, IBM, Microsoft, and others are investing in emotive analytics as part of the larger AI, machine learning, and natural language fields. IBM has the artificial intelligence Watson, Google acquired DeepMind in 2014, and there's Facebook AI Research (FAIR). They are investing internally, externally, and acquiring startups [9] that will help their digital assistants such as Alexa, Siri, and Cortana as well as create more customized,

personal experiences for users through current online experiences. Even non-tech companies are getting in the game: Shell is investing in AI to create a virtual assistant that can answer questions around the clock. [10]

The conundrum is that while technology is new and change is easy, we don't know the full impact on society. By the time we know the full impact, technology is so integrated into society that change is hard [11].

However, what seems scary in the present becomes normalized in the future. When sites first started asking for credit card information over the web, people were very worried about data security and privacy. People still have those concerns (and rightly so), but much of the population doesn't think twice about typing in their digits. New industries have grown up around security, hacking, and alternative forms

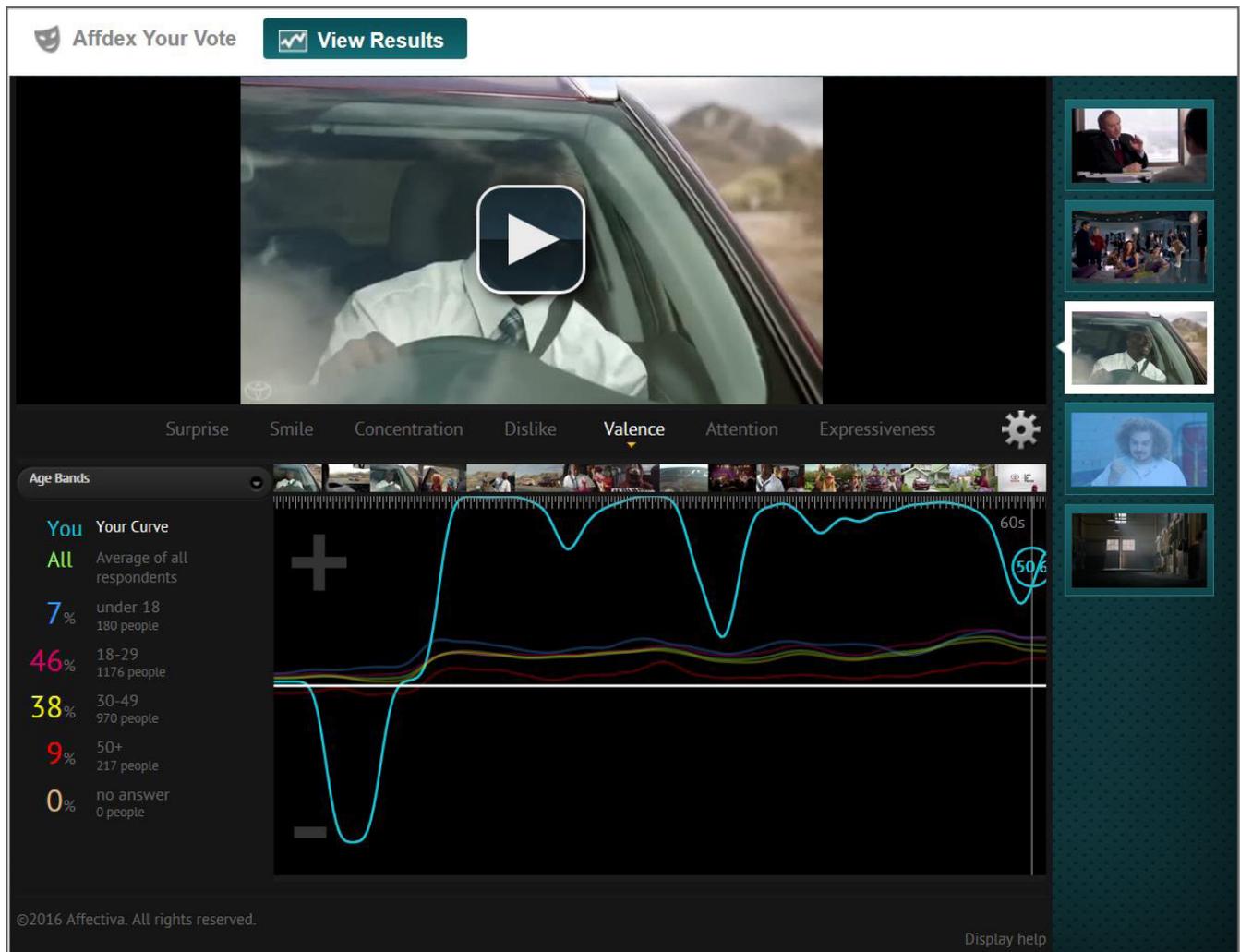


Figure 4. Affdex Dashboard Measuring Users' Emotional Reactions to a Video Ad

of online payment such as BitCoin. Our emotions are even more private and personal than our credit card numbers. It falls on us to create and adhere to standards and ethics, assisted by privacy rights groups such as the Electronic Frontier Foundation, Privacy International, and others.

Marketing

Parent corporations such as Procter & Gamble, Unilever, Coca-Cola, Diageo and Mondelez are all using emotive analytics, and most are hiring emotion-recognition companies to test and refine content and media strategy through language and tone. [3]

Affectiva has software development kits (SDK) and application program interfaces (API) for people to use, and you can see how you react using their Affdex tool, shown below. Once you give permission to use your webcam for the length of the one video, points on your face will be tracked, scored, and added to the database. You can then see how you compare with the other user reactions. Marketing can use reactions to fine-tune or reject various ads, or predict virability.

Call Centers

Emotive analytics software, either a standalone system or built on top of a customer relationship management (CRM) system, monitors both the customer's tone of voice and certain key words as well as the call center agent's tone of voice. Certain flags can trigger a supervisor's notice for immediate attention or follow-up. [12]

Employee Workforce Satisfaction

Employee satisfaction is just as important as customer satisfaction, since happy employees are more productive and less likely to leave. It costs a lot of money to onboard a new employee, and employee loss is expensive not only in dollars but in tribal knowledge as well. At work, we have an ergonomic program called RSI Guard that measures typing and mousing activity and prompts us to take breaks to rest our hands and eyes. At some point in the future, we may have a program that monitors our moods and suggests wellness actions. [13]

Education

Some of the most difficult parts of teaching, both in-person and online, is gauging comprehension and attention. Imagine if online learning courses could analyze a student's face and stop for clarification if the student looks puzzled, or suggest a break if the student's attention is wandering. Emotuit is one company working to improve student engagement and content interaction in K-12, higher education, massive open online courses (MOOC), and enterprise training.

Gaming

Role playing games could change storylines depending on how the gamer reacts emotionally to characters and storylines. Affective gameplay in educational games could improve cognition, memory, attention, and motivation. [14]. Games designed for young children could help them learn about their emotions and what to do with their big feelings. Games inside robots such as Milo already help autistic children practice communication and social skills. [15]

Medicine and Healthcare

Have you ever encountered someone who looked fine but just didn't sound right and then ended up being sick? Doctors may use emotive analytics and AI to help a human or robot caregiver to recognize pain or depression in patients. Telemedicine may also better reach and serve people in rural areas, and emotive biometrics may add to the data doctors collect remotely [16].

Government and Politics

Politicians use many marketing tactics and advanced analytics during the election cycle and to garner and gauge support for policies. People can get very emotional about candidates and issues, and it pays to use emotive analytics. Government organizations might also adopt emotive analytics for security or psychological operations.

Summary

Emotive analytics, though still a developing technology, is being integrated at a rapid pace into multiple facets of life thanks to heavy investment by large tech

companies and general corporations. New technology brings new challenges for standards and ethics.

Exciting and scary at the same time, emotive analytics is here to stay, and will eventually become a necessary metric business within companies' analytics suites.

Resources

General Helpful Articles on Related Subjects

AI and Cognitive Computing. IBM Research. <http://www.research.ibm.com/cognitive-computing>

Category: Cognitive Computing. Content Wrangler. <http://thecontentwrangler.com/category/categories/cognitive-computing/>

Companies and Projects (not a complete list)

Affectiva <http://www.affectiva.com/>

BehaviorMatrix <http://www.behaviormatrix.com/>

Beyond Verbal <http://www.beyondverbal.com/>

CallMiner <https://callminer.com/>

CrowdEmotion <http://www.crowdemotion.co.uk/>

Emotive Analytics <http://emotiveanalytics.com/>

Emotuit <https://www.emotuit.com/>

Hanson Robotics at <http://www.hansonrobotics.com/>, <http://sophiabot.com/>, and

iMotions <https://imotions.com/>

IntraFace (Carnegie Mellon University) <http://www.humansensing.cs.cmu.edu/intraface/>

Kairos <https://www.kairos.com/>

Lightwave <http://www.lightwave.company/>

nViso <http://nviso.ch/>

Real Eyes <https://www.realeyesit.com/>

RoboKind Advanced Social Robotics, Robots4Autism <http://www.robokindrobots.com/> and <https://vimeo.com/94215300>

References

[1] Johnson, Tom. "Technical Communication Metrics: What Should You Track?" I'd Rather Be Writing (March 2, 2012) <http://idratherbewriting.com/2012/03/02/technical-communication-metrics-what-should-you-track/>.

- [2] Ivy Exec. "Spotlight: Emotive Analytics." Ivy Exec. <https://www.ivyexec.com/executive-insights/2017/spotlight-emotive-analytics/>
- [3] Bashford, Suzy. "Why Emotional Analytics Are The Next Frontier." Campaign (March 8th, 2017). <http://www.campaignlive.co.uk/article/why-emotional-analytics-next-frontier/1425800>
- [4] Proff, Allie. "How Does Speech Recognition Work?" Technically Eclectic (February 20th, 2017) <http://technicallyeclectic.com/how-does-speech-recognition-work/>
- [5] Proff, Allie. "How Does Face Recognition Work?" Technically Eclectic (April 18, 2017) <http://technicallyeclectic.com/how-does-face-recognition-work/>
- [6] "Gartner's 2013 Hype Cycle for Emerging Technologies Maps Out Evolving Relationship between Humans and Machines." Gartner (August 19, 2013) <http://www.gartner.com/newsroom/id/2575515>
- [7] Burton, Betsy "Hype Cycle for Emerging Technologies, 2016." Gartner (August 19, 2016) <https://www.gartner.com/doc/3383817>
- [8] Nguyen, Tuong Huy. "Hype Cycle for Emerging Mobile Technologies, 2016." Gartner (July 11, 2016) <https://www.gartner.com/doc/3383817>
- [9] Kosalitcheva, Kia. "Apple Acquires Startup That Reads Emotions From Facial Expressions." Fortune (January 7, 2016). <http://fortune.com/2016/01/07/apple-emotient-acquisition/>
- [10] Mercer, Christina. "Nine Tech Giants Investing in Artificial Intelligence." Tech World (December 13, 2016). <http://www.techworld.com/picture-gallery/big-data/tech-giants-investing-in-artificial-intelligence-3629737/>
- [11] Thomas, Elise. "Are You Happy Now? The Uncertain Future of Emotion Analytics." Hopes&Fears (October 5, 2015). <http://www.hopesandfears.com/hopes/now/internet/216523-the-uncertain-future-of-emotion-analytics>
- [12] Horwitz, Lauren. "Beyond Verbal Takes On Emotions Analytics in Call Centers and Beyond." SearchCRM (June 2014) <http://searchcrm.techtart.com/feature/Beyond-Verbal-takes-on-emotions-analytics-in-call-centers-and-beyond>
- [13] Reback, Gedalyah. "10 Startups Pioneering The New Field of Emotional Analytics." Geek Time (February 19, 2017). <http://www.geektime.com/2017/02/19/10-startups-pioneering-the-new-field-of-emotional-analytics/>

[com/2017/02/19/10-startups-pioneering-the-new-field-of-emotional-analytics/](http://www.technology.com/2017/02/19/10-startups-pioneering-the-new-field-of-emotional-analytics/)

[14] Wilkinson, Phillip. "Affective Educational Games: Utilizing Emotions in Game-Based Learning." IEEE Xplore Digital Library (October 10, 2013). <http://ieeexplore.ieee.org/document/6624219/>

[15] "Meet Milo." Robots4Autism. <https://robots4autism.com/milo/>

[16] Higginbotham, Stacey. "Computers That Understand Your Emotions Are Coming Next Year." Fortune (December 23, 2015). <http://fortune.com/2015/12/23/computers-understand-emotions/>

Author Contact Information

Allie Proff
Knowledge Management/Librarian
Boeing
www.linkedin.com/in/allieproff
allieproff@gmail.com
+1 425.890.8594

Author Biography

Allie has been researching, writing, and organizing technical information for over fifteen years, whether it be in the U.S. Navy, as a high school math teacher, as technical writer, a librarian, and as a content strategist in knowledge management at The Boeing Company.

The session title pays homage to science fiction writer Philip K Dick, whose book "My Android Dreams of Electric Sheep" inspired the movie Blade Runner. The main character, Rick Deckard, is tasked with "retiring" (killing) some rogue androids. After meeting an android so intelligent she doesn't realize she's an android, you're left to wonder if Deckard himself may be an android. Since this topic covers artificial intelligence in the world of emotive analytics and marketing, no marketing or viral reference is complete without cats or cat videos (though puppies are a close second).



Ruggeri, Kathleen and Lisa Adair

Improve Content Quality with One Process Change

At Rockwell Automation, we implemented a machine-editing program to improve our content quality and translation-readiness. By concentrating on the creation of clear, concise, and consistent source content upstream, we improve the downstream translation process, which gains results in reduced translation costs. We simultaneously implemented a terminology management program to provide consistent use of terms across all content. All communicators use the same term database and all translators use the same translation database. By controlling which terms to use (including trademark control), we improve consistency and accuracy. By adding machine editing to our process, we have automated content quality procedures that improved our source content and enabled faster and more cost-efficient translations.

Issues We Were Facing

As is common in many companies, we continually face challenges in doing more with less. More products being developed, faster schedules, and fewer resources made us seriously examine our processes and tools.

Our editors were adding value and improving content quality, but we had only two of them to edit the work of more than 30 communicators in over 600 documents per year. We implemented levels-of-edit (full, medium, or light), but that still wasn't enough. Triage decisions were made daily to prioritize what would be edited.

Editors had to remember key terms, or spend the time to do manual research for the terms in the style guide. Trademarks were also handled in a separate listing, with a painstakingly slow process of searching the document and then comparing to the trademark list.

With editor comments in PDF files, there was no way to track existing quality or improvement and thus we couldn't measure translation cost savings.

We knew that we needed to improve this area of our content development process.

Process Change: Implementing Machine Editing

We decided to investigate machine editing, which is software that analyzes your content for:

- Style
- Readability
- Search Engine Optimization (SEO)
- Terminology
- Tone/Voice

It is an artificial intelligence tool that operates at the sentence level. While machine editing tools cannot understand your content, and cannot do what a human editor can do, there are many advantages.

Machine editing moves the effort to improve content quality **upstream** in the content development process. Complex grammar and detailed style rules from our internal style guide are programmed into the tool, thus automating the tedious tasks of human editing. Communicators can concentrate on writing their content, and then run the tool to correct the grammar and style issues.

Manual Editing	Machine Editing
Limited primarily by time (work measured in hours)	Limited by processing capability (work measured in seconds)
Cannot increase scope/communicators without adding extra hours (headcount limitations)	One-time cost to increase processing power or add licenses with unlimited scope (Everyone with a license can be edited)
Execute one-by-one with reliance on human editor	Execute whenever communicator wants (beginning, middle, and/or end of project)
Slight variations in execution (human nature)	Same rule checked the same way for every user, every time
Evaluate and report manually	Automated analytics

Table 1. Manual versus Machine Editing

There are many options that are available for machine editing:

- ProWritingAid
- StyleWriter®
- WhiteSmoke®
- Grammarly®
- Acrolinx®

We chose Acrolinx, which is a natural language processing engine that guides communicators as they create content.

How We Tested and Implemented Machine Editing

To make sure that machine editing would be appropriate for the highly technical content we produce at Rockwell Automation, we arranged a pilot to test the tool on a small group of users. Eight communicators used Acrolinx with the default rules for six months and, even without specific customization for our style and only a limited set of terms, the feedback was overwhelmingly positive!

The business case to justify the expense was easy to prepare. We already had data from the pilot to prove that it could improve our content and thus positively impact the translation costs.

We hired a consultant to help us customize the rules and train our tool administrators. The consultants also helped us harvest terms from our existing content to create our managed terminology.

We rolled out the tool in a controlled fashion, keeping to small groups in each campus one at a time. We customized the training to use our own content, with real examples of how machine editing was applied to real documents.

Those eight communicators who participated in the pilot became super users, and continue to assist the communicators in their respective groups.

Our mantra through the entire training phase was ‘happy adoption’. We wanted our users to be comfortable with the tool without pressure, and see its benefits.

Acrolinx Overview

Using Acrolinx lets you achieve better consistency in your content across the organization. Because company style rules are customized in the tool, communicators do not have to remember everything that is in our style guide. Because the grammar rules are incorporated in the tool, the communicators produce fewer complex sentences.

Acrolinx eliminates the need to look up terminology in a separate listing; running the tool immediately identifies where incorrect terms are used, and automatically provides the correct term and the appropriate usage. Trademarks are better controlled, which helps maintain our brand integrity.

With cleaner, clearer writing and consistent use of terms and trademarks, translation costs are reduced.

What You Can Customize

Machine-editing tools have varying levels of customization; we chose Acrolinx because of its flexibility.

There are over 100 pre-programmed grammar and style rules. In our implementation, we use over 80 of those out-of-the-box rules for basic good grammar and typical technical communication style issues. We have customized existing rules to account for our type of content, and we've created more than 20 new rules specific to Rockwell Automation. We also created different sets of rules for different types of communicators (technical communicators, marketing communicators, etc.).

Terminology is Now Formalized

Before we implemented machine editing, editors used to edit for everything that they knew, but no one owned or managed terminology. We now have a customized termbase with more than 1000 English terms. [Managing the 81,000 multilingual terms is a related, but separate function.] Up-front decisions are made for which terms to manage, and machine editing provides the guidance to support the communicators (use this term/not that term).

Company trademarks are a key piece of terminology; with over 800 Rockwell Automation trademarks, it was difficult for the communicators to know all company trademarks and how to use them properly. Acrolinx identifies the appropriate terms without having to do any manual lookup. Also, any additions or updates from the Legal department are quickly added to the termbase, and available almost immediately to all communicators, ensuring all content is properly protected in real time.

Checking Your Content

Acrolinx is a plug-in that is available for multiple platforms, including Adobe FrameMaker, Adobe InDesign, Microsoft Word, and Microsoft PowerPoint. Communicators can run the tool at any time during content creation from the Review ribbon – on a sentence, paragraph, chapter, or an entire document.

Acrolinx flags the issues that conflict with the rules and provides guidance on how to correct the flag right in the content. You can run checks individually (for example, just on terminology), or all at once.

The check produces a scorecard that shows how well the content adheres to the rules, and then shows the issues in detail in a list.

Results of Machine Editing

Now that we've been using machine editing since 2014, what improvement have we seen?

We compared content that was edited before we implemented machine editing with content that was machine edited. The biggest impact was in terminology and Trademark usage where we saw more than 50% improvement in the scores. We have much tighter control and consistency, which impacts our customer experience significantly. Many of our customers have manuals open side-by-side while using our products. Having this consistent terminology builds and maintains our brand and makes it easier for them to apply our products.

We also know that the complex grammar rules are helping our communicators write cleaner content and shorter sentences. We saw more than 70% improvement in the scores for shorter and less-complex sentences. Not only does this impact translation costs, but it also helps our customers.

Four times more communicators now have editing support. Instead of a human editor who has only so many hours in the day, the machine editing tool is always available to all communicators that have the tool installed.

Also, as Rockwell Automation adds more global development locations, non-native speakers who write in English get support immediately through machine editing, without time zone delays.

Content Quality Improving Translations

Machine editing helps in producing high-quality content. High-quality content is faster to translate and validate. Quality is built into the source content once (English), which avoids interpreting, and potentially fixing, errors each time in each language.

Because we can identify the specific rules that most effect translations, we engage with the translation team, our vendors, and validators for discussions on how the tool implements these rules. We are learning more about the complexities of language and continually tweaking rules to produce the most simple and clear content. This effort leads to unambiguous translation and better efficiencies. And we are building a process to improve our content translatability over time, continuously and systematically.

Is Machine Editing for You?

There are several factors to consider to decide if machine editing is appropriate for your company:

- How complex is your style guide? There's only so much that a tool can do for you, and it won't catch everything. You may still need some kind of quality check for things that the tool cannot be programmed to find (for example, pagination, template adherence, appearance/format of graphics/tables).
- You need funding for a new tool. Acrolinx has a cost-per-user licensing scheme and a fee for the hosting server. We also found that having a consultant is key to our successful implementation. Remember to account for these fees to assist with configuration and for ongoing support.
- You'll need to put resources in place to maintain the tool and adjust the rules over time; we repurposed our editors to configure, maintain, and assist users in using the tool.
- There is also a soft cost associated with implementing a tool that fundamentally changes how you write your content. There will be an impact to existing translation memory (as terminology is tightened up and sentences are made less-complex). There will also be some productivity dips as people learn the tool and adjust to using it.

But – there are also significant benefits:

- Using a machine editing tool makes you an even better communicator. Complex grammar rules are programmed into the tool to help you write simpler, more translatable content.
- The evaluation and assistance from the tool is instant, and right in your document. You can see the issue and make adjustments with guidance from the tool. You don't have to wait for the editor to return your comments; you can see the issues and correct them, right as you are writing.
- It removes the burden of remembering the many details of style and terminology, which gives you more time to learn the product you are documenting. You can spend your time determining the best way to document the product, and less time looking up terms and trademarks.
- For new communicators, it gets you up to speed faster in the company. Much collective and 'tribal knowledge' is embedded in the tool in the style and terminology.

Only you can decide if implementing machine editing is appropriate for your company and the content you write. At Rockwell Automation, we have found it to be an integral tool in our development process.

References

StyleWriter is a trademark of Editor Software, Ltd.

WhiteSmoke is a trademark of WhiteSmoke, Inc.

Grammarly is a trademark of Grammarly, Inc.

Acrolinx is a trademark of acrolinx GmbH.

About Rockwell Automation

Rockwell Automation Inc. (NYSE: ROK), the world's largest company dedicated to industrial automation and information, makes its customers more productive and the world more sustainable. Headquartered in Milwaukee, Wis., Rockwell Automation employs approximately 22,000 people serving customers in more than 80 countries.

Author Contact Information

Kathleen Ruggeri
Manager, Information Architecture and Content
Strategy
Rockwell Automation, Inc.
1 Allen-Bradley Drive
Mayfield Heights, OH 44124
440.646.4423

Lisa Adair
Lead, Content Quality
Rockwell Automation, Inc.
1 Allen-Bradley Drive
Mayfield Heights, OH 44124
440.646.3174

Author Biography

Kathleen Ruggeri is a Senior STC member with almost 30 years of experience in the Technical and Marketing Communication fields for Rockwell Automation. Kathleen currently manages the Information Architects, responsible for product-level technical documentation planning, and business-level technical content strategies. Kathleen is also the Program Manager for our Content Quality initiative.

Lisa Adair has been in the Technical Communication department at Rockwell Automation for over 20 years. A Senior STC member, she's recently transitioned from her role as Editor to Content Quality Lead. In this new position, Lisa supports existing Acrolinx users, helps bring new user-groups on board, and works closely with the Translations group to assist in terminology management.



Saxena, Shikha

The Art of Writing in Agile

Abstract: This paper describes the need of hour for technical Writers to enhance their technical skills and have a complete understanding of business area/product as well, for which they are writing. Fill in all technical creativity which is our USP and being hand in hand with Agile methodology trends, write in such a way which is simple and short and crisp! The paper provides pointers on how writing in Agile is an Art to pick up and how we can add a dash of our creativity in a document, making it easy to understand, faster delivery and create quality product to be utilized by business.

The present paper describes the gradual transition from Traditional Technical Writer to Agile Technical Writer over a period of four years. It also emphasizes the need for technical writers to embrace change and adopt the new technologies and methodologies proactively. Try to leverage useful principles and apply it in to their unique style of writing, to carry on the journey seamlessly. Learning and implementing new methods and sharing knowledge gained, should never stop to keep your writing fresh and edgy. Monotony and stagnancy in writing are injurious for documentation. Comprehensive and detailed documentation is useful, though just-in-time, precise and to-the-point documentation, adds value and is aligned to the product vision and customer requirements. This paper includes the mantra to enhance the art of writing by applying the principals of Agile Methodology to Software documentation.

Proactively attending Agile awareness and training session for beginners, three years back made it possible for me, to create grounds and mindset, to enter in the field of Agile and apply its principals in software product documentation. It allowed me to research and implement the methods best suited for software documentation. It was something new, exciting and in agreement with my style of simpler, better, and faster documentation. Also, it just fitted well with the Agile projects; I was selected to document. There were a lot of queries, during evolution from Waterfall to Agile and there

were no rules and guidance, set aside, regarding documentation.

Gradually studying existing content over the net and attending online training about Agile enhanced my knowledge. I started understanding Agile rules and working styles; I started picking up, placing puzzle pieces together, practicing and applying Agile to my product, process, and business documentation. Correlating and adopting Agile principles, (that I began to appreciate by now), with my style of documentation, helped me evolve, from traditional Technical Writer to Agile Technical Writer. This enabled me to draw the inference that the Agile model principles and practices are incomplete, and we should also consider the issue of, how to be effective at the content creation and maintenance of documentation in Agile mode.

Transition to Agile means we need to rethink the way we document and should avoid wasteful effort and time. This could be done by producing just enough, in or before time, precise, quality document which is error free and lean. Also, be ready to welcome customer feedback for changes, incorporate it accordingly and iterate the document to bring out the best of it.

The current paper put forward application of some Agile and lean philosophies in bringing about the Technical documentation in IT organization in the financial sector.

Understanding and applying Agile practices to documentation could be helpful to deliver faster, up-to-the-mark, quality, technical product documentation, while also meeting customer expectations.

Among those are:

- Agile environment setup
- Agile mindset creation
- Agile team member selection
- Receptive to Customer Feedback
- Enthusiastic to learn, create and deliver best
- Self-motivated team members to deliver best with least bugs
- Value Time
- Communicate extensively to be on the same page
- Transparency among team members
- Ownership of the product
- Collaboration of co-located and virtual teams
- Intuitive to foresee risks and impediments and raising alarms
- Creative towards verbal expressions
- Adaptive to changing requirements
- Patience and belief to cater problems by bringing quick fixes/solutions to problems
- Learning by mistakes
- Respect Opinions
- Add-on-Value
- Applying DevOps
- Conscious effort to change and be the change
- Teach Agile by Action rather than preach
- Contribute Maximum by being part of this Agile journey

Aligning Software Development and Documentation in Agile

Software developers have the knowledge and us technical writers have the skill to write. Developers are the best person to write about the product they have developed, though technical writers can improve on

taking up and accomplishing this task of understanding the technical product and documenting the same per audience of the document. The best approach is that developer and writer work in collaboration to write documentation learning from each other in the process. The developer can share knowledge during and after development, requirements, transitioning release into production, testing, problems, and solution phases. The writer may take multiple roles as understanding increases. They can act as first testers of the product to express the functionality in a better way, also become the refactors eliminating unwanted details, detecting bugs, and raising the alarm in time, rectifying the product. This way add value and increase the quality of Product document. During development, documentation rough drafts, sketches and inconsistency are acceptable; the document also have scope for improvement leading to the creation of perfect documents. I prefer writing during development, as soon as various functionality in the sprint are complete and iterate and increment the document alongside. Seeking reviews from the development team and business stakeholders, maintain the accuracy and the quality of the document intact! Documenting in parallel with developing software captures latest and relevant information. As refactors, a writer can receive feedback and rework on the document and repeat the cycle to generate up to the mark, working and relevant document, as required by the customer.

How to Create Agile Documentation

Once ready with idea and vision about product and kind of audience document is being created follow simple steps as summarized in this section. Document either a Product Overview, System and Architectural document, User story, Acceptance Criteria User or Procedure manual document, depending upon the request from customer, just visualize the scenario and start writing the document. Share the document rough draft with customer, which is just good enough (with just 'Vision' and 'Introduction' content), to kick-off writing assignment and receive insight and feedback. There will be some document handy to start simple conversation and discussion followed by feedback incorporation to enhance document build.

Following are the simple steps for writing in Agile environment that I worked on, during Agile product documentation:

1. Analyze and understand the functionality of the product being documented from customer.
2. Select the template to document, best suited for requirement.
3. Start with suitably naming the document with version number, date and so on as it will be changed during numerous iterations of that sprint.
4. Write some document information for audience, to be used as reference later during course of sprints.
5. Write vision and strategy of the software product being created.
6. Write introduction about generated document as about the prospective audience, include the document naming conventions, version, release number, page header-footer information and document navigation information.
7. Give introductory information about the product and its architecture and its communication with other systems and functionalities, within the document.
8. Include few program flows and architectural details for your understanding, giving document a shape and conversation with customers to receive further insight about expected working product.
9. Keep the flow of the product screens intact, as you navigate the product yourself, if given chance of product testing. Else add the new functionality as program flows, screens, reports to the document, detailing the fields and their attributes.
10. Mention even smaller details of product usage in simple step from user's perspective.
11. Keep your writing simple and concise with less usage of jargons. It reflects your understanding of the product functioning. The way you interpret the same shows your powerful articulation.
12. Clarity in writing comes from caring about the customer ease. Explaining complex functionality in your clear and simple documentation is an art and this artwork will receive appreciation.
13. Keep active voice throughout document and try a check on grammar, even in first draft. Quality matters and error free documentation add value to the created product. Your artwork should be a visual delight for customers.
14. Revise, refactor and perform content check from customer perspective to give your best and share document with team with confidence. Welcome feedback. It will always contribute towards quality of document and your positive attitude as a team member.
15. Capture product screen images, format, and place in document, provide proper caption and describe each screen field details (field name, field descriptions, attributes and alike) and its functioning. Make use of our special power for "attention to detail". Detect any error or defect you come across, in the field name spellings, message phrasing or so and inform team proactively, to be taken care of, as development is still in progress.
16. Place the document on shared location and communicate team its location, to be viewed and referenced, by team members, for open discussion and providing suggestions.
17. Be open to suggestions and feedback incorporation, it will help to add quality and create workable dynamic document as per customer vision.
18. Be agile in feedback integration and sharing the document back with product team to adjust discrepancies if any.
19. Communicate the exact location of document to team and be ready to incorporate the feedback

to increase transparency within team for anyone to pick it up any time, even in your absence.

20. Check for formatting, make document readable and appealing and not crowded with images and content.
21. Add relevant information about the product at appropriate location with a separate “Notes” section for emphasis.
22. Minimize use of jargons to make document simple to understand, that appeals to customer and business.
23. Leverage your technical and creative writing skills to impress customers with your intellect and add value to documents. Suggest sample documents and furnish supporting documents in time, proactively.
24. Be up-to-date and agile by learning new tools and techniques. Feel young and innovative, by sharpening your technical skills, in addition to writing. People around you will be amazed, as there is no age limit to learning!
25. Interact more with developers and testers and try to shift your pi-skill set to comb-skill sets by learning new skills and sharing knowledge. Try exchanging hats, occasionally, with developers, testers, it will be learning with fun!
26. Create relevant and just required, good enough content, do not overdo it.
27. Create a “What’s New” section to include the latest update for added features in a release and add this section along with each release to give a highlight of newly added functionalities in different sprints of the release.
28. Mentor and guide junior writers by showing your work, furnishing tips of writing proactively, helping them frequently in documentation, to carry forward legacy of agile writing in your absence.
29. Maintain pace of work, to sustain your writing and feedback incorporation and refactoring content in many sprints to come.
30. Enjoy writing and be self-motivated to bring about innovations in writing and convey to larger audience.

When is Document Considered Agile?

Documenting for an agile project is not said to be Agile documentation in true sense.

Created document is stated to be agile when it fulfills following criteria:

1. Agile document has specific customer. There are many customers to create document for, also there are different types of documents and various writing styles. We need to work closely and collaboratively with the customer to capture their requirements and produce documents they visualized.
2. Agile Document have a purpose. Know the purpose before creating the document. Try your best to understand for whom and why you are creating the document. It will give clarity to your thought and hence reflect in your document too.
3. Agile documents are lean and sufficient. Writing just enough and precise is required in agile document. Steps and point wise writing is even fast to refactor too. Easily go back and make the changes in the document without much change in the content structure.
4. Agile documents maximize stakeholder ROI. If benefits of documentation outweigh the investment in its creation and maintenance, then definitely creation of this documentation was the best option available for customers.
5. Stakeholder knows the TCO of document. The stakeholder must understand The Cost of Ownership (TCO) of the document and must be willing to invest in its creation and maintenance.
6. Agile documents describe good things to know. The agile documents should be simple mean that they should include critical details. They should include exclusive information that is not available elsewhere. For example, “delete button will delete the information” is so obvious though the information that “delete button will delete information from front end screen though backend will update this activity for records” is something which adds value.
7. Agile documents are accurate, consistent, and detailed. The agile documents must be accurate, consistent is the need though

detailing a document depend on the customer requirements. If a system is highly critical as medical, surgical software then detailing is necessary. However, less critical systems require bare minimum details which are just sufficient for the current purpose.

Agile Artifacts

Epics, Themes, and User Story are Agile artifacts which technical writers create during initiation of the Agile projects. These are the high-level requirements. These are brainstormed and discussed during project initiation meetings where customer envisions the team with product vision and strategy. Developers and domain experts also take part in discussion and put forward their view. Further, these requirements are simplified to smaller workable tasks to introduce clarity of thought.

Epics are vision of the project they are the dominant view of the final product. These are larger user stories, typically ones which are too big to implement in a single iteration and therefore need to be disintegrated into smaller user stories at some point. Epics take lower priorities in the work stack or product backlog as it works its way towards the top disaggregating into smaller user stories placed higher in stack, based on Just-In-Time (JIT) basis, to increase the overall productivity. It is not useful to disaggregate lower priority epic because it takes investing extra time and effort on it. A portion of it which is more important will take higher priority and needs to be teased out further to bring value. For instance, University registration system, STC Summit 2017, Indian epics - Ramayana and Mahabharata are examples of Epic with numerous themes and stories inside them.

Themes are collection of related user stories. Themes are used to organize stories into releases or in such a way that various sub-teams can work on it. For example, in university registration system themes might be around students, course management, transcript generation, grade administration, financial processing, study loans, and so on.

User stories are one of the primary development artifacts for scrum and extreme programming project teams. They are high-level definition of a requirement, containing just enough information so that the developers can produce a reasonable estimate of the effort to implement it. User stories are important part of agile approach that helps to shift the focus from

writing about the requirements to talking about them. All agile user stories include a written sentence or two upon which forms the basis of series of conversations about the desired functionality. Consumer and product owner envision team of business analysts/ technical writers, developers, testers, domain experts about the product vision, strategies, and requirements. Technical writers write the user stories after understanding the business scenario and product requirements in a format which is self-explainable shown as follows:

As a <type of user>, I want <some goal> so that <some reason>.

For example, informal requirement during team discussion is “students need to purchase a parking pass.” Whereas formal user story, could be written as:

“As a student, I want to purchase a parking pass so that I can drive to school.”

Disciplined Agile Delivery (DAD)

There are three distinct phases in DAD life cycle. There are three standard times to write stories during agile projects. Inception phase is writing user stories while understanding the requirements initially during envisioning activities to identify the scope of the system. Construction phase is writing stories during development iteration, when recent stories are identified, existing ones are further split into smaller stories, to simplify them and making them short, if they are too big for that iteration. Reprioritizing existing stories to place the new ones in the stack or product backlog and removing stories which are no longer needed or out of scope. Stories evolve as the requirements evolve during the increment of the product and they need to be added to product backlog for developers to work on them. In Transition phase, new stories are identified during the end of development while ready to implement. They are of higher priority and placed on the stack on top just as in standard scenarios.

Art of Writing User Story

Agile user story writing techniques are the best way of capturing product functionality and envisioning the group, about the customer requirements, for creating the product. More creative and efficient User Stories, more clear and concise are the information passed

to the developers and more useful will be the newly created final product. This individual task of creating stories is assigned to Technical Writers/Business Analysts, as a member of Agile Teams. Understand who the User or Customer are and what problems they would like to see addressed before you start writing epics and user stories. An excellent way to capture user and customer insights are personas/imaginary users from which you can understand, discover, and derive epics. Brainstorming the problem together as an agile team and sharing knowledge and envisioning team will help to create better user stories and desired product.

Breaking epics into simpler stories will contribute to the team to build valid codes and acceptance criteria, and definition of done will help to create tested complete product ready to increment. Always start writing from epics and simplify it further by breaking it into simpler user stories in as a user/I want /so that format. Refine stories by team collaboration and make them simpler and transparent so that it can be transformed into correct codes.

Prioritize the stories per the functionality need and rearrange them in product backlog stack and then work on them in sequence for code/content creation. In this disciplined agile change management process, lower priority stories are modeled in fewer details, and high priority stories are modeled in greater details. New work items can be added to stack any time during iteration, and they may be reprioritized and removed at any time. Each iteration implements the highest priority stories first. Decide on acceptance criteria with testers and team collaboration. Testers ask questions that elicit examples and clarifications. Brainstorm to determine when story will be considered complete and what criteria is needed for Definition of Done (DOD). User stories are given Acceptance Criteria (AC), ACs are business oriented tests that let us know if the implementation meets product owner expectations. Write ACs so that the product owner verify that the story was implements as intended. Developers and Testers find out when they are done.

Gradually burndown the story/task chart, once stories are completed and done. The product starts taking shape. Burndown chart shows total effort against the amount of work we deliver in each iteration. Use paper cards, sticky notes, and charts to write user stories and display them. Make story burndown visible/viewable to all team members. It will make tracking down the work done easy, furnish solution to any impediments and blockers. Felicitate the show stoppers publicly

in ceremonies and influence others by easy show, of the agile work done and changing mindset. Use other methods like use of wireframes, test cases to describe the desirable functionality.

User Story and Planning

There are two areas where user stories affect the planning process in Agile projects. In agile scheduling work items and user stories is important. Apply numerical prioritizing on scale of 1-20 or use MoSCoW (Must Should Could Won't) approach. Stakeholders have the right to define new requirements, change their minds about the existing ones and reprioritize as they want. However, they are responsible for taking informed decision and keeping transparency in informing the team about the changes promptly. Developers are responsible for estimating the effort required to implement the stories. Developers learn to develop the estimating skills as they are the one who needs to fulfill it ultimately.

Pair programming is another technique which requires two developers to work and implement user story in a single iteration/sprint. If developer is working alone, then epics need to be broken into simpler stories to complete the implementation by a single person in that iteration. During JIT analysis and model storming with stakeholders explore the details behind the story. Carry out discussions and understand verbal and body language and their enthusiasm to explain required product. Be intuitive and help them to bring about their vision in a simple way to the team. Create screen sketches or diagrams to explore what are their expectations from the product. Identify the acceptance criteria or confirmations which stakeholders will use to validate that the user story. Use the backside of story card for the purpose. List the programming tasks required to implement story.

Apply INVEST mnemonics as used for agile software development. It a reminder of features of good quality Product Backlog Items (PBI) commonly written in user story format and used in Scrum, Kanban backlog and Extreme Programming (XP) project. INVEST is contraction for Independent, Negotiable, Valuable, Estimable, Small, and Testable respectively. PBI should be independent of other PBI. Even if they are dependent on each other, it is suitable to combine them together. PBI are not contracts, so they have scope for discussion. These are rather fixed artifacts still there is an opportunity for clarification and

negotiation during iteration. It can be rewritten even discarded depending upon business, market, or technical requirements. PBI must deliver value to stakeholder.

Estimate size of PBI to plan. PBI must be small and not so big that it becomes difficult to plan/task /prioritize with a level of certainty. PBI size should be only a few person-days or few person-weeks. Any PBI in iteration should not take more than 50% of iteration. PBI and its description should be so clear that it makes test development possible.

Break user stories into tasks for the benefit of developers. SMART is contraction for creating effective goals. S- Specific, M- Measurable, A- Achievable, R- Relevant, T-Time-Boxed. Task need to be specific to express everyone, as what is involved in it. Other tasks are saved from overlapping. Example, Changes in screens, reframing sentences, or content, changing the flow of topics, code refactoring and so on. Mark, it done when done completely what was required. For example, Tests, code refactoring and so on. Task owner is expected to achieve a task. He has the backup and collaboration of a team. Every task should be relevant and contribute to the story. It also should fit into customer's expectation, and its presence should be explained and justified.

The task should be time-boxed in specified time duration. There may not be a precise estimate of hours or days though members should know when to seek help. If it is harder than expected then it needs to be split up in further tasks or change of players is required to get the task and hence making the story, done and complete.

Addressing Challenges in Agile Documentation

Studies show that Agile methodologies can be applied well to documentation projects. Even though we frequently face few challenges, which we need to overcome and work around often, to leverage Agile principals beneficially to our use. Some impediments I came across while documenting in agile projects are Keeping pace with rapid Product development due to continuous feedback incorporations, refactoring, rewriting, reformatting, and keeping documents innovative. It consumes time and effort in refactoring, rewriting and discarding the documentation wrote

previously. Frequent updates in documentation to clear backlog is the solution to this problem.

Another method is to travel light means do not include lengthy details and keep your document light and small during an agile journey and write just what is required, at that time, for that audience. Understanding product and context of writing and connecting through multiple sprints and tools is another challenge. Proper knowledge of the technical product and deciphering it into your content development is essential to place the new iterations and corrected tags in the document correctly. The product requirement analysis and mapping and aligning the documents of the product created, to set the new iterations and corrected tags, is complex part of the documentation. This consumes a lot of time and effort to keep up with the primary task of writing new iterations and functionalities added to the product.

The best possible way is to spend time with developers and testers and gain knowledge about the product. Analyze the product backlog time to time and keep track of work done in the user stories. It aids in keeping you well informed and agile in capturing the newly completed functionalities, in the document, in parallel. No formal signoff working documents to act as documentation base is yet another challenge. You start writing documentation based on the knowledge gained during scrums about product. Based on this information you need to assume and structure your documents and share the same with the group at the earliest, to get going. This requires your intuition, proactive approach, and agility, to create the first draft, share with team, and receive their feedback. The first draft now acts as basis of conversation and enable to team to discuss requirements and prioritize the functionalities. This will further increase your understanding of the product and enable you to understand the requirement and write clearly.

Communication, for clarification of thoughts, is important in agile environment. It is challenging to carry on detailed discussion and clarification, if you are stuck on certain points, as people are on the go, in agile environment. People are busy in tasks and lack time to invest on discussions or assume, you are agile enough to build, on your knowledge. Daily scrums assume that everybody is on the same page and similar frequency though it might not be the case always. In such scenarios provision of maximum existing sample models and meta data with the writers could be of some help.

Co-location of teams for continuous interaction is what is expected in Agile environment though this could be one of the challenges. The existence of virtual agile teams cannot be overlooked. When this is, the case matching the time frames of working virtual team members is a challenge. However, this could be overcome by finding solutions as placing documents in shared locations. Continuously work in such a way, that part of team submits and share the work at the end of the day and other half of the team pick it up, as the day starts at their end. Lack of standard set of tools is yet another challenge in Agile documentation.

Exploring Prospects to Automate Documentation Verification with Frequent Changes

Automate documentation and perform document chunks assembly, could be the area where we can focus. Design the systems and workflows such as to create electronic documents. To create documents where it can use pre-existing texts as basis, which can take up data entries and could be linked to updated product back log sheets, to verify the changes done in the system. This could help organizations in minimizing data entry, save the time spent in review, improve correctness and accuracy of the documents, reduce risks caused by human error, bring value to finances spent to raise such system.

Conclusion

The transition from traditional Waterfall approach to Agile approach is challenging for teams. However, learning the new agile ways to hasten the process of documentation and apply principles of agile to software documentation will result in the simpler, better, and faster deliveries. Documentation is an art, and we should use our writing prowess with a dash of technology, to create meaningful product with value. Product teams use documentation as a valuable artifact for reference. Create minimum but useful documentation which contains the valuable information, for reference and use by the business. Such as product overview and release documents, development team use user stories, DOD, testers use acceptance criteria; users make use of user manuals to create meaningful working products, in time and customer reap maximum Returns on Investment (ROI). Technical Writers should align their art of writing

with agile trends and technologies and create useful agile documents, which are molded per customer requirements, to reap maximum benefits! Travel light, keep consistent pace, sustain in Agile Journey, to reach destinations!

Resources

- Cohen, Mike. User Stories Applied (For Agile Software Development, Indian Edition, 2016).
- Shore, James and Warden, Shane, The Art of Agile Development (Tenth Indian Reprint, February 2017)

References

- [1] Ambler, Scott, and Associates. <http://www.agilemodeling.com/essays/agileDocumentation.htm>
- [2] Cockburn, Alistair "Agile Software Development" (2001). <http://alistair.cockburn.us/Shu+Ha+Ri>
- [3] Shu-Ha-Ri: the stages of Learning to Mastery (July 21, 2011). <http://www.3coast.com/shuhari-the-stages-of-learning-to-mastery/>
- [4] Agile Alliance 2015. <https://www.agilealliance.org/agile101/the-agile-manifesto/>
- [5] Cohen, Mike. <https://www.mountaingoatsoftware.com/agile/user-stories>
- [6] Pichler, Roman. <http://www.romanpichler.com/>
- [7] Wake, Bill. "Invest in Good Stories and Smart Tasks" (April 17, 2003). <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

Author Contact Information

Shikha Saxena, Senior Technical Writer, HSBC Technology, Pune, India
Senior Technical Writer
HSBC Technology, Pune, MH, India
shikha_saxena69@hotmail.com. [linkedin.com/in/shikhasaxena69](https://www.linkedin.com/in/shikhasaxena69)
+91 9960638974

Author Biography

Shikha Saxena is a Senior Technical Writer in HSBC Technology, Pune, India. She is involved in Financial Product and Business Architectural documentation within organization. She has created Technical,

Screens, Reports, User, and Procedure Guides. Electronic Procedure Manuals (EPMs), System Operation Procedures (SOPs) in financial domain. Also, involved in different phases of Agile project for user story writing and product documentation. She is an active blogger on technology, employee stories and well-being sites of organizational intranet and managed web content writing, editing, and publication of the same. She is a speaker and participant, in national (STC, Pune, India, July 2016) and International conferences (tekomp/tcWorld February 2016, Bengaluru, India). She was a lecturer, teacher, researcher and mentor to Science and Engineering Graduates and Post Graduates of Uttar Pradesh Technical University (UPTU), in NCR, New Delhi, India before initiating her global career as a Software Engineer/Technical Writer (2009). She is certified COBOL (ICS, Scranton, PA, 1995) and VisionPlus programmer in Mainframe and has product development and testing experience in Cards and Collections domain. She holds Bachelor (Chemistry Honours) and Master (Biochemistry and Biochemical Engineering) degrees. She earned Gold Medal and Jawaharlal Nehru National Award during her Masters in Biochemical Engineering, in IIT, BHU, Varanasi, India in 1993. Shikha is a Certified Clinical Research Associate and Oncology Terminologist and got her blogs published, about her struggle, research, and cure, for her ailment, in global well-being sites of current organization. She loves reading Hindi, Urdu verses/poetries, and cooking. She is passionate about mentoring and informing students, colleagues, friends and underprivileged to change, with changing technology, learn and share knowledge. She is an oil-on-canvas painter, enlighten community in multiple ways and contribute towards making society, a better and knowledgeable place, as there is... Just One life to live!



Keith Schengili-Roberts

Is Darwin Information Typing Architecture (DITA) Right for You? Scenarios for Considering a Move to DITA

DITA XML is the fastest-growing standard for structured content. In the dozen years since its launch, DITA has changed the way many organizations undertake their own technical documentation, as well as changed the discourse around how structured content can better communicate consistent messaging in an efficient and cost-effective manner. Some of the more typical scenarios for documentation teams moving to DITA include using content reuse to make processes more efficient, the desire to reduce localization costs, and the need to ensure accurate content goes out the door. For some other organizations different pain points arise, such as having to deal with an increasingly unwieldy base of documentation, a growing lack of consistency, discovering the limits of traditional desktop publishing software, or the need to break down siloed writing groups. Common themes emerge when documentation teams are considering a move to DITA, and this article looks at when making the move to DITA makes sense.

Understanding DITA

The Darwin Information Typing Architecture (DITA) is a XML standard that has been widely adopted within the technical writing community, and has been acknowledged as the fastest-growing technical documentation standard. It is currently in use by almost 700 firms worldwide.

The acronym DITA stands for its chief components:

- “D” is for “Darwin”, alluding to Charles Darwin’s concept of evolutionary adaptation, as the structure of DITA is based on the ideas of specialization and inheritance
- “IT” is for “Information Typing”, defining the semantic structure of individual topics
- “A” is for “Architecture”, anchoring it in the idea that this is a structured standard that is also an extensible

Another key design philosophy incorporated into DITA is content reuse. The topic-based nature of DITA means that it is a modular, so it is possible to use the same topics in different maps. When used properly, this approach ensures consistent messaging. The modular nature of topics and the way they can be swapped in and out of maps means that writers have to take a non-narrative approach when writing content for a topic, as they cannot be sure of the placement of a given topic within a map. As a result, a best practice is to ensure DITA topics can be read by users as standalone content, concisely conveying exactly what a user needs to know for a given scenario. Content reuse also occurs below the topic level, with conrefs (“content references”) and keys which can be used interchangeably for handling content reuse at the paragraph, sentence, phrase or individual word level. This ability to reuse content at a very granular level as well as at the topic level is one of the most distinguishing features of the DITA standard, setting it apart from other structured documentation standards.

Content reuse also greatly reduces the costs associated with translating content. If, for example, English source topics that have already been translated are used in a subsequent publication and Translation Memory (software used for keeping track of previous translation work) is used, the amount of new content that needs to be translated is greatly reduced. In some cases, the amount of words that need to be translated can be zero. This feature of DITA makes it attractive for companies seeking an efficient and cost-effective way to produce content in multiple languages.

Several formatting output types (including XHTML, PDF, HTML5, CHM and others) are available to the writer when they want to “publish” their DITA content. DITA XML separates form from content, so the XML content is transformed using the Extensible Stylesheet Language (XSL) to produce content in the desired target format. The DITA Open Toolkit (DITA-OT) is a freely-available, open source publication engine that is most commonly used to produce content in the desired target format.

Typical Reasons Why Documentation Teams Move to DITA

The most common reason why firms end up moving to DITA is economic: a need to reduce costs when producing technical content. While there are many other good reasons to want to move to DITA, the economic argument—typically manifesting in some form of budgetary pressure—is a key factor. A key question for many documentation teams considering a move to DITA is: when does it make sense to migrate over to using DITA? If you and your documentation team are experiencing any of the following pain points, then it is time to consider moving to DITA.

Pain Point 1: The Need to Reduce Operational Costs

Budgetary pressures are a constant concern for most documentation departments, with most teams finding it hard to increase or even maintain existing headcount. In a few cases I have run across, there is even a need to increase the amount of documentation while having to reduce headcount. The process efficiencies that come with DITA—particularly when combined with a Component Content Management

System (CCMS)—make it possible to “do more with less.”

Effective content reuse is key to reducing operational costs. Once a topic has been written on a given subject, that means that other writers on the team no longer have to re-write that same content. On the surface this may seem trivial, and many documentation teams using a standard desktop publishing program (DTP) may think that they do the same thing for much of their own content, particularly when it comes to iterations of a standard product manual. A key difference is that DITA content tends to remain stable, whereas a documentation team doing iterations of a standard manual will tend to introduce tweaks and changes that build up over time. That’s not to imply that individual DITA topics are immune to change, it’s just that with standard practice they are subject to less change than an equivalent section that is continually updated in a DTP-based publication. My research has shown that almost all DITA content is maintained but documentation teams either within a commercial CCMS or some form of version control system, ensuring that content must be checked out prior to being edited, and that previous versions can be reverted to. In contrast, changes to DTP-based documents tend to accumulate in a linear fashion over time, ensuring that content needs to be checked and re-checked for consistency from publication to publication.

As it is based in XML, DITA separates form from content. As a result, writers concentrate wholly on creating new content rather than formatting issues, which can take up a surprising amount of time. Several years ago I was the manager for documentation at AMD, and we were moving to DITA. I wanted to get a sense as to how much time the technical writers on my staff spent solely on formatting within the DTP software they were using at the time. I was amazed to find that roughly half of their time was spent on formatting-related issues. When I delved further into what was happening, the amount of time spent on formatting made sense, as it included things like the writer changing header levels across hundred-plus page documents, ensuring that inline formatting was consistent within a document, or the time spent fixing formatting issues that caused the DTP software to crash repeatedly when generating output.

This last point is not an uncommon one. Once, while working as a publication consultant for a different semiconductor company I ran across a case of a senior chip design engineer who had devoted considerable

time and effort creating a process designed to ensure that the content for their multi hundred-page chip design documents went smoothly. He had created a lengthy process document outlining the minutiae of what his fellow engineers should do to produce their publications to ensure that their DTP software did not crash when attempting to generate output. Moving to DITA removed these issues for them at one go, enabling the chip designer, his engineering team and the technical editors that supported them to concentrate on creating content rather than formatting it. Not surprisingly, this increased their content output. It also allowed the senior chip design engineer to get back to his job of designing chips, as opposed to documentation procedures.

Content reuse combined with operational efficiencies allows documentation teams to be more productive, enabling them to create more content in order to meet demand. In my own experience at AMD, we found that year over year we were able to produce more documentation to meet an increasing need, even in years when headcount was reduced significantly. Another example of this came to light when I interviewed a documentation manager who was using DITA at another firm, who reported that when a member of their team retired, they did not feel the need to replace him, due to the process efficiencies they gained by adopting DITA and using a CCMS.

Pain Point 2: Decreasing Localization Costs

The chief reason why localization costs are reduced when documentation teams use DITA is directly related to content reuse. DITA content reuse in English (or whatever the source language is) means that an already-translated topic does not need to be re-translated again.

This is not the only factor that comes into play. The separation of form and content that comes from a DITA-based publication process means that Localization Service Providers (LSPs) can no longer charge for formatting content within a DTP. If a firm had been using DTP software for creating content that was then localized, the LSP expected charges for formatting translated content in that same DTP software is often considerable. If the localized documentation is large—running to hundreds of pages—and the LSP charges for this work by the hour, the expected cost savings for

moving to DITA may be greater than that of content reuse.

A DITA-based localization process can enable more than just cost savings. Some firms have used the cost and process efficiencies that DITA provides in this area to produce more localized content for their customers, or to tackle additional languages that previously would not have been possible from a budgetary standpoint. DITA-based process efficiencies can also enable quicker turnaround times for localized content. For example, if a documentation team produces a document using a DTP-based process, typically they have to wait for the content to be finalized so that it can be formatted, and only then is it sent for localization. Using DITA and a CCMS, it is possible to send finalized topics in a serial fashion to the LSP for work. The LSP does not receive a sudden large influx of content as it would under the old, DTP-based system, and can work on the content piecemeal. This type of process enables faster overall turnaround times for localized content. Even if some of the topics that were originally sent end up being changed, the ability to more readily meet tight deadlines usually outweighs any additional cost.

One of our larger customers has added automation to their DITA-based localization process in order to provide hourly drops of content for their LSPs to translate. They are now often able to provide their customers with localized updates to their product content within the day.

Pain Point 3: The Need to Improve Content Quality

I have already mentioned several of the process efficiencies that DITA can provide, but there are additional indirect features that can be provided when DITA is used with a CCMS.

One of these is the ability to improve content workflows with the aim of improving documentation quality. Within a typical DITA-based CCMS once a writer has automatically finished work on a topic, it can automatically be routed to the next person in the verification chain, such as an editor or a subject matter expert. As the content is topic-based, it is easier for the reviewer to process. The typical alternative using a DTP-based system is for the reviewer to receive a chapter or multiple chapters of content to review at once. Much like the earlier example of an LSP being able to meet more aggressive deadlines when given

finished topics to localize in a serial fashion, the throughput for editors and reviewers is increased as they are able to review more content when it is presented in bite-sized topics.

The topic-based nature of DITA means that it is possible to assign subject matter experts (SMEs) with topics that they must “fill” with content. When DITA first emerged a dozen years ago, it was used almost exclusively within documentation departments. These days I am witness to an increasing number of CCMS deployments where SMEs are expected to be a central part of the content creation process, with DITA being seen as a way to help facilitate this.

There are certain industry sectors where the need for verified content is high, such as the medical device sector. In this business, SMEs are often medical doctors who are not only expected to create content but also to verify it for accuracy before it goes out the door. For much of the semiconductor industry, what they sell is their intellectual property, which is wrapped up in their electric engineering documentation. A purchasing design engineer from another firm needs accurate information on a component for sale, and if that is lacking or incorrect, they are less likely to buy that component. For these two industries and for others, having verified content is a necessity, and topic-based DITA content plus a CCMS for routing workflow provides a crucial check on this process.

Pain Point 4: Company Has Moved to Agile

From a company’s perspective, moving to Agile is not a “pain point”, but in my experience documentation teams are rarely consulted before a firm’s development team moves to Agile. In environments where business factors are pushing for rapid change in product development, Agile methodologies are more likely to be introduced.

Both the Darwin Information Typing Architecture (DITA) and Agile were born out of necessity for software development teams. Technical writers within the software division at IBM established DITA in order to efficiently create effective and collaboratively written topic-based documentation. Similarly, the Agile Manifesto came from a group of software developers seeking a more lightweight way to create their deliverables. All of the various “flavors” of Agile—Lean, Scrum, Kanban, Extreme Programming and others—

are all ultimately derived from the common principles laid out in the original manifesto, each providing their own take on how to create software more efficiently than waterfall. Thought originally created for different reasons, DITA and Agile share common roots in the software development world, and DITA can make Agile for documentation teams possible.

There are many key factors that make DITA-based technical documentation complementary to Agile-based product development. This first is that the topic-based approach in DITA assists with incremental development. One of the tenets of DITA is content reuse, encouraging technical writers to “write once, use many”.

DITA best practices advocate that content is focused squarely on the user, which is the same focus and this ties in nicely with Agile’s similar focus. Technical writers are able to provide early feedback on products through their active use of the product. In this way, technical writers often become an advocate for users; this in turn helps define realistic user stories. The constant change and iterations of content over multiple releases force a change in the typical writer’s mindset from “document everything” to instead “document only what the user needs”. Again, the granular, topic-based nature of DITA helps make this possible.

The DITA best practice of minimalist writing reduces “waste” from a customer perspective, which is also in line with Agile practice. One of the key concepts of Lean Management—an Agile methodology—is to reduce waste wherever possible. This is encapsulated in the Japanese term “muda”. In the case of documentation, this refers to content that is unnecessary in order for the customer to use the product. It serves as a check on technical writers from writing “filler”—typically background or marketing-related content that a user does not need in order to accomplish a particular task or action with the product. One of the philosophical underpinnings of DITA is minimalism, which similarly tells writers to pare content down to its essentials.

Agile encourages continuous feedback and because DITA topics tend to be small, topic-based review is easier for developers than having to review a full chapter or more. In this way, documentation can also support broader communication between teams, customers, audit processes, etc. Work cycles are faster, and documentation feedback becomes more critical.

Thanks to the separation of content formatting built into DITA, technical writers can focus on creating content rather than formatting it. As we have seen, this can save a considerable amount of time. This also eliminates the time wasted when SMEs comment on formatting instead of the content they are supposed to be reviewing.

All of these factors make using DITA within an Agile development environment an ideal choice for documenting content.

Other, Lesser-known Benefits of Using DITA

While the pain points I have already described are the most common reasons I have seen for moving to DITA, there are other, lesser-known benefits that come from making the switch.

One of these is the maturity of the tools that are now available for supporting a DITA-based toolchain. While there are many XML editors available, a professional documentation team will want a tool that has convenience features tailored for working with DITA. There are several of these on the market, including Syncro Soft's oXygen editor and JustSystem's XMetaL. They include features that make it easier for technical writers to work with DITA content, such as providing in-context XML tagging options, making it easier to find and reuse content, and to publish material for review. A typical DITA CCMS will enable information architects and technical writers to search for individual topics and their contents and then assemble them within a map using a graphical user interface. More mature DITA CCMSes provide lightweight tools for enabling SMEs to write, edit and verify content without the users having to learn DITA. They also provide sophisticated versioning and branching of content, and make it easier to both find and create content for reuse. As the number of firms moving to DITA increases, the tools for porting content from legacy systems continues to improve, and it is not uncommon to hear of large numbers of documents being successfully converted to DITA. From the perspective of technology adoption life cycle, it appears as though DITA and its related tools are now well out of the early adoption phase and is beginning to enter the early majority, as the success of DITA and the tools that support it become more well known. While job posts in the United States seeking technical writers with DITA experience hovers at 5%, for the past three years significantly more technical

writer job posts seek XML experience over that of FrameMaker, which has long been the DTP software of choice for technical documentation groups.

Individual DITA topics can be counted, allowing for documentation project measurement. While a lot has been written about justifying the Return on Investment (ROI) for moving to DITA, the topic-based nature of the standards means that it is easy to measure the rate of ongoing content production. For example, in a typical Scrum-based Agile environment, everyone involved in a project gathers together to discuss progress in regular meetings. Using a traditional DTP-based approach, all that can typically be reported is the word count or the number of chapters completed since the last sprint. With DITA, it is possible to match development features to individual topics, making it easier to report in a more realistic manner on documentation progress. If a CCMS is used, workflow status—draft, in review, done, etc.—can also be measured and reported at the Scrum meeting.

The fact that DITA is open source brings with it other benefits. DTP systems are proprietary in nature, which means that asking for significant changes either involves costly development work, and you are left to the whims of the DTP maker for updates and fixes. With the DITA Open Toolkit—the tool most-used for producing output—its workings are readily visible and anyone with the requisite skill set can modify or develop style sheets, modify publication processes as well as access to a growing number of free, open source tools for transforming content. As an example of this, a recently-introduced plugin now makes it possible to take in Markdown-based content that utilizes DITA syntax and to publish it.

When Does a Move to DITA Make Sense?

This simple answer to this question is: “when it is most cost-effective to do so”, but that should not be the only reason. For example, the process improvements that DITA can provide are harder to quantify in terms of dollars, though they may enable a documentation team to work effectively under a new production process, such as Agile.

While I have addressed the most common pain points encountered by technical documentation teams and prior to their move to DITA, it should be remembered that any move to DITA does not come for free.

Additional costs typically include:

- Conversion costs of legacy content
- Change management and training costs
- Cost of CCMS software
- May involve separate purchase of repository software for content.

The associated costs depend on the specific circumstances for each documentation, but while these costs can be considerable—especially at the beginning—the process efficiencies and cost-reductions make them fiscally possible. A move to DITA should also be considered an opportunity for making changes and improvements to your ongoing documentation processes. Base your decision for a move to DITA not only for budgetary concerns, but where you need to take your documentation team and what your users will expect of you five years from now.

Author Contact Information

Keith Schengili-Roberts, IXIASOFT
DITA Evangelist and Market Researcher
IXIASOFT
825 Querbes, Suite 200
Montréal, Québec, Canada
514.279.4942

Author Biography

Keith is the DITA Evangelist and Market Researcher at IXIASOFT. Keith is also an award-winning lecturer on Information Architecture at the University of Toronto's School for Continuing Education. He is also the company's liaison with OASIS, actively working on the DITA Technical Committee, the Lightweight DITA sub-committee. Most recently he has become Chair of the OASIS DITA Adoption Committee, succeeding in this role from JoAnn Hackos. He can often be found presenting at conferences, researching how DITA is being used and sharing those results with the DITA community. Look for his chapter on the DITA marketplace in the upcoming book *Current Practices and Trends in Technical and Professional Communication*, due out in 2017. He lives in the east-end of the Toronto along with his wife and two daughters. Keith's popular industry blog DITAWriter.com has become a focal point on DITA resources and best practices. Connect with Keith on Twitter @KeithIXIASOFT.



Semp, Monique

Leveraging Structured Authoring/DITA Techniques When All You Have Are Unstructured Tools

All too often the “experts” imply that you must use an enterprise-level authoring ecosystem, with its inherent cost and overhead, in order to get any of the benefits of structured/DITA writing. However, my experience has shown otherwise: you can create well-structured, consistently-styled, and reusable content no matter what your standards (formal or not) and tools are.

Introduction

This paper grew from my experience with DITA authoring. Wanting to “learn structure,” I read a lot from the field’s experts and attended workshops and conferences. This led to an assignment with a large enterprise software company with a large team of information architects and writers who were creating DITA content. My next assignment was also for a “DITA shop”, but with only two writers and no “tools team.” I still liked the theory of DITA/structure, but began to reject it as unrealistic for all but the largest companies. Finally, I returned to an old client for whom I’d created a large body of unstructured FrameMaker content. Even though I knew it intimately, I found it troublesome to manage after enjoying the easy consistency and content reuse of the structured/DITA world. So I began to investigate my tools in a deeper way, wishing to leverage the techniques from my structured/DITA writing. As I worked through a couple of releases, I was amazed at how much “structure” I was able to incorporate into my unstructured FrameMaker doc set!

A brief caveat: You should not interpret this paper as a dismissal of DITA/structure, nor specific replacement advice. Rather, I hope to spark thoughtful consideration about the essential features and benefits of DITA/structure, and how to achieve the benefits with the context of your current tools and workflow.

What is Structured Authoring?

There are many definitions of structured authoring, but at its most basic, it’s simply consistency in structure—the order of information for a given type of content—and consistency in writing.

It’s important to emphasize that *structure* is not synonymous with *DITA* or *DocBook*. As we just defined structured authoring, it could encompass words painted on papyrus. *DITA* and *DocBook* are simply the XML-based standards that typically support structured authoring, although one can certainly write poorly structured text no matter what the tool.

The most widely adopted standards-based frameworks for structured authoring are DITA and DocBook, which are both enormous and complex. The Lightweight DITA committee struggles to define a functional subset of the 1,199-page DITA 1.3 spec. And for DocBook, the conventional wisdom is that you’ll discard the vast majority (>90%!) of the spec before creating content.

Faced with the impossibility of using anything approaching the full feature set of such standards, the sensible thing to do is shift the goal to defining a structure that suits your immediate needs and that’s extensible for future requirements.

Structured Authoring and Structured Toolset Features

To say that content is structured means (among other things) that it enables:

- Easy content reuse—Content is referenced, not copied. Variables and conditions enable easy use of *almost identical* content.
- Implicit focus on topic-based writing—The DITA specification defines topic types such as concept, task, and reference. The DITA standard uses a *map* to encapsulate all the desired topics; sort of like an outline.
- Semantic styling—Adds meaning to the content, separate from the content itself. Instead of applying direct formatting to text, such as bold, the semantic tag itself defines the styling. For example, using a *uicontrol* tag makes it easy to change all instances from bold to italic.

Typical structured authoring tools, such as oXygen/XMetaL, Structured FrameMaker, and the DITA Open Toolkit, provide:

- Multi-output publishing—Source files are used as input to a *publishing engine*. Like OOP (object-oriented publishing), the goal is WORM: Write Once, Read [Publish] Many.
- Metadata support—Many uses: <https://en.wikipedia.org/wiki/Metadata#Creation>, and

structured tools generally provide a GUI to easily manage metadata.

- Portable content—Content can be round-tripped among different editors and publishing engines.

Why Isn't Everyone Using DITA or DocBook?

So if DITA, DocBook, and formal structured authoring tools are so great, why isn't everyone using them?! Well, a lot of very good reasons:

- **Lots of legacy content to convert or maintain.** Organizations that need DITA or DocBook are likely to have large bodies of existing content. Converting it to a new system is resource and cost intensive, but so is having separate toolchains for legacy and new content. In such cases, inertia often wins out, and no new tools are introduced.
- **Tools are expensive in terms of TCO (total cost of ownership).** Some open-source tools are free to download but become costly to implement because of the technical skills required.
- **Steep learning curve.** Writing in DITA or DocBook requires training and time, which is difficult to integrate with today's quick-to-market documentation requirements.
- **Complex authoring-publishing ecosystem.** Large companies that implement DITA or DocBook typically have a dedicated tools team, and tools

 Frame (2015)	 Word (2010)	 Flare (11)
<p>For each “chapter,” add a folder to the .book file. The folders serve as containers for separate (topic) .fm files. See http://blogs.adobe.com/techcomm/2014/07/why-upgrade-from-fm7-8-or-9-5-book-building-and-external-references.html</p>	<p>A couple of approaches:</p> <ul style="list-style-type: none"> • Create separate files for every topic, and include them in a master doc. Historically, master docs have had problems, but they do seem workable. • DitaExchange integrates with Word, letting you create DITA topics and maps without knowledge of DITA markup. <p>See References page.</p>	<ul style="list-style-type: none"> • Based on XHTML, whose specification defines a well-formed document—a document that adheres to the syntax rules specified by the XML 1.0 specification. So structure is inherent in the content. See References page. • Fully and easily supports separate files for topics, which can be included in any number of docs (books, help systems, etc.).

Figure 1. Enforce a consistent document structure.

 Frame (2015)	 Word (2010)	 Flare (11)
<ul style="list-style-type: none"> • Use file naming conventions. • Use model <i>template</i> .fm files. • Configure <i>Next Pgf Tag</i> for all paragraph styles. 	<ul style="list-style-type: none"> • Use file naming conventions. • Use model <i>template</i> .dotx/.dotm files. • Configure <i>Next Pgf Tag</i> for all paragraph styles. 	<ul style="list-style-type: none"> • Create topic templates that contain predefined CSS classes, in a given order. • Use a metatag to define the topic type. • Some Flare styles contain logic for the next line’s style; all styles are editable via CSS.

Figure 2. Use consistent styling for topics.

can be very difficult to manage for lone writers or small teams.

- **Long, or even infinite, ROI.** Although some requirements, such as many products or a lot of translation, can make adopting DITA or DocBook a technically sound proposition, it can be difficult for a company to justify using DITA or DocBook based purely on ROI.

Depending on the content, DITA can be a solution that’s looking for a problem.

Adopt the Features of Structure!

So if you’re in the no-DITA, no-DocBook world, what can you do? Adopt the best features of structure!

Writing is writing. You can define the content’s “structure,” write a good short description, and write in a topic-based fashion, even if you’re chiseling content onto a stone tablet. You can write consistently-structured topics, and employ tools such as templates or model docs. And you can write a paragraph in the form of a short description whether or not it’s actually tagged a <shortdesc> XML element.

Unstructured authoring tools have many of the same features as structured/DITA tools for creating consistent and extensible content, reusable content,

semantic styling, metadata, and multi-channel publishing.

Unstructured Tools’ Support for Structure

By definition, a document’s structure is implemented within the context of the authoring environment’s (tool’s) fundamental strengths and weaknesses. This paper examines three typical Tech Pubs tools. Adobe FrameMaker (Unstructured), Microsoft Word, and MadCap Flare.

When you first attempt to “adopt structure”, it might seem an overwhelming task. However, “structure” is not an all or nothing thing. You can choose to adopt the easiest things, the most important things, or some other category that fits your needs and ROI.

And while these techniques do not result in a fully structured body of content, you might want to adopt these concepts is to prepare for a new toolset/ ecosystem or to migrate existing content. Migrations typically require some of these changes, or are easier if they’re implemented.

Note. Some of these techniques and adaptations are short-term solutions/workarounds. As such, they can

 Frame (2015)	 Word (2010)	 Flare (11)
<p>Create a <i>shortdesc</i> paragraph style, and assign this style as the Next Para for all heading styles.</p>	<p>Create a <i>shortdesc</i> paragraph style, and assign this style as the Next Para for all heading styles.</p>	<p>Create a CSS class that identifies a specific variant of an XHTML element.</p>

Figure 3. Style a short description.

 Frame (2015)	 Word (2010)	 Flare (11)
<ul style="list-style-type: none"> Write the topic in a standalone .fm file, and import it as a text inset. Recent thread from the [Framers] list: <i>Using Frame as a little CMS</i>, http://www.mail-archive.com/framers%40lists.frameusers.com/msg65018.html 	<ul style="list-style-type: none"> Master documents. SmartDocs is designed with content reuse in mind; see http://www.thirtysix.net/ RiverFloe is a "document generation machine for Microsoft Word." See http://www.riverfloe.com/ 	<ul style="list-style-type: none"> Topics can be included in any number of projects. Included snippets are automatically updated if you update the source.

Figure 4. Reuse whole topics.

be great quality/productivity aids, but they do not take the place of the ideal—tools chosen as a result of a docs/needs analysis. As well, not all features are in all a tool's versions.

Maintain Consistent “Doc” Structure

In DITA, topic-based authoring is designed in. Separate files for every topic are included in a ditamap, and DITA-aware editors enforce structure as content is created. Hierarchy is typically flat (few heading levels). Some non-structured tools easily support separate files for every topic, while others require using advanced features.

Create Consistently-Styled Topics

DITA topics are inherently consistent with respect to style and structure. The standard topic types—concept, task, reference—are defined with rigid requirements as to the order of information. With specialization, you can create custom topic types, such as troubleshooting. DITA-aware editors enforce the structure of the content by validating the content element (tags) against the Data Type Definition (DTD).

Theoretically, you can achieve the same level of consistency with non-structured tools. You can create templates and model docs, develop file naming conventions, and use paragraph style attributes to ensure consistently structured content. However, these manual mechanisms do require diligence and editorial review.

 Frame (2015)	 Word (2010)	 Flare (11)
<ul style="list-style-type: none"> Customize system variables. Add user variables. Easily manage variables via tools such as Leximation BookVars, http://leximation.com/tools/info/bookvars.php 	<ul style="list-style-type: none"> Use AutoText to define content fragments. Use the AutoText field code to enable easy doc updates for modified AutoText entries. Custom variables (in Advanced Properties) let you create true variables. Building blocks provide even more control. See References for details. 	<ul style="list-style-type: none"> Variables can be used in topics, snippets, and print page layouts. You can define variable sets. You can define a set of doc-specific variables (title, version, etc.), and assign them as UPDATE-IN-TARGET. For each target, specify the variable values.

Figure 5. Reuse content fragments.

 Frame (2015)	 Word (2010)	 Flare (11)
<ul style="list-style-type: none"> • Full-featured conditional text management provided. • Best practice: Create a per-project .fm template that has only conditions. Import its Conditional Text Settings into all a doc's .fm files. 	<ul style="list-style-type: none"> • No dedicated function, but by using hidden text, you can fake it; see References. • Use SmartDocs' conditional functionality, http://www.thirtysix.net/smartdocs/features. • Use Document Automation in Clio, a cloud-based management software system for the legal industry; see References. 	<p>Conditional text feature lets you tag content at the:</p> <ul style="list-style-type: none"> • Topic-file level • ToC level • Target level • In-topic content level • In snippets

Figure 6. Change the content *just a little*.

Write a Short Description, <shortdesc>

In DITA, the <shortdesc> element gets a lot of attention. Not only does it function as an abstract for the topic with sufficient information that experienced users can often skip the details (unlike the days of information mapping, where almost every topic included *signpost* text with little value), typical publishing tools enable the <shortdesc> text to appear with search results, and can even be searched themselves.

But there's nothing magic about the <shortdesc> element itself. Any text can satisfy the best practices of a short description, as defined in the "Guidelines for Writing Effective Short Descriptions" section of the *DITA Best Practices* book; or see the OASIS White Paper, *DITA Feature Article: Short Descriptions Shouldn't Be a Tall Order: Writing Effective Short Descriptions* (18 March 2016), https://www.oasis-open.org/committees/download.php/57803/DITA-Adoption_2016_Writing-Effective-Short-Descriptions.pdf.

Reuse Whole Topics

In DITA, any topic can be included in any number of ditamaps (document containers). Typical DITA implementations include a CCMS (content configuration management system), which makes it easy to keep track of topic use. When you know everywhere a topic is used, you can manage its conditions and variables for the most effective reuse, and know how broad an effect a change will have.

When you use non-structured tools, there are often mechanisms for topic reuse, but keeping track of topics and managing updates and revisions it is up to you.

Reuse Content Fragments

In DITA, <conref> and <conkeyref> elements act as variables that you can include in any topic, as well as customize on a per-document basis. Non-structured tools typically provide this capability via variables.

 Frame (2015)	 Word (2010)	 Flare (11)
<p>Create character and paragraph styles; for example:</p> <ul style="list-style-type: none"> • cite • uicontrol • code • filepath 	<p>Create character and paragraph styles; for example:</p> <ul style="list-style-type: none"> • <cite> • <uicontrol> • <code> • <filepath> 	<p>Create a CSS class that identifies a specific variant of an XHTML element. For example: a div element, which in turn can specify unique formatting to its child elements.</p>

Figure 7. Style content semantically.

 Frame (2015)	 Word (2010)	 Flare (11)
<p>Create paragraph styles such as <i>prereq</i> and <i>postreq</i>. In the Paragraph Designer:</p> <ul style="list-style-type: none"> Use the Autonumber Format to specify the lead-in text, such as “Prerequisites” and “What to do next”. Assign the Next Pgf Tag to a bullet style paragraph. 	<p>Create paragraph styles such as <i>prereq</i> and <i>postreq</i>. Use outline numbering to specify the lead-in text, such as “Prerequisites” and “What to do next”.</p> <p>(For help creating numbered styles that display lead-in text instead of numbers, see the References page.)</p>	<p>Flare has built-in features for creating browse sequences, related topics, and more.</p>

Figure 8. Standardize navigation aids.

Change the Content *Just a Little*

It’s very common to want to reuse content, but change it just a little; for example, steps that vary just slightly for different operating system versions. In DITA, conditions let you manage small output variations in a topic, making it easy to maintain consistency. Some non-structured tools provide advanced condition management, while others require extensive workarounds that are rarely feasible or workable long-term.

Style Content Semantically

Although it’s possible to apply hard-coded formats to DITA content, the temptation to do so is reduced by the typical non-WYSIWYG editorial view. Formatting is generally applied at publication time, based on the content’s applied elements, according to style sheets and tool-specific mechanisms. For non-structured tools, the best practice is to eschew hard-code formats such as bold and italic. Instead, create meaningful

character and paragraph styles to apply to the content. For example, create a cite character style instead of using Ctrl-i to apply italic.

Standardize Navigation Aids

Typical DITA output transforms add standard navigational text based on element types; for example, “Prerequisites” before a <prereq> element and “What to do next” before a <postreq> element. When using non-structured tools, you can style the paragraphs to automatically include standard lead-in text. This way, you don’t need to enter that lead-in text, and global doc changes are easy.

Add Metadata

In DITA, you can create <metadata> elements, with a variety of attributes, and apply the metadata to maps and topics in a very granular fashion. Unstructured,

 Frame (2015)	 Word (2010)	 Flare (11)
<ul style="list-style-type: none"> Supports Extensible Metadata Platform (XMP). Use the File Info feature for book and/or file metadata. For historical info, see http://www.ipc.org/std/lptc4xmpCore/1.0/documentati on/lptc4xmpCore_1.0-doc-CpanelsUserGuide_13.pdf. 	<ul style="list-style-type: none"> Add the Prepare Document function to the ribbon (see References), and configure the (Document) Properties. To add additional metadata (or for variable use), access Document Properties > Advanced Properties > Custom tab. 	<p>Includes a variety of built-in metadata controls:</p> <ul style="list-style-type: none"> Topic types Custom file tags Any XHTML-compliant tagging structure that you want to create.

Figure 9. Add metadata.

 Frame (2015)	 Word (2010)	 Flare (11)
<p>Polished, highly-customized output is “easy.” You need:</p> <ul style="list-style-type: none"> • Good visual design(er). • Good templates. • Disciplined style use. 	<p>Polished, highly-customized output is “easy.” You need:</p> <ul style="list-style-type: none"> • Good visual design(er). • Good templates. • Disciplined style use. • Professional PDF creator (Acrobat or Nuance). 	<p>Good-quality PDF is easy out-of-the-box. Not excellent because:</p> <ul style="list-style-type: none"> • Flare PDF engine does not recognize every CSS element or attribute; for example, CSS <i>text-overflow</i>, <i>white-space</i>, and <i>overflow</i> attributes. • Flare can't process PostScript fonts.

Figure 10. Publish to PDF.

“real” authoring tools (vs. simple text editors) also include features for defining metadata.

Publish to PDF

With DITA content (unless it’s in FrameMaker), it’s difficult to get a PDF that is as polished and precise as you can with traditional DTP (desktop publishing) tools. You must extensively customize stylesheets for use with your PDF transformation engine—typically the Apache™ FOP (Formatting Objects Processor)—which requires knowledge of XSL formatting objects (XSL-FO). Commercial tools such as Antenna House Formatter and WebWorks ePublisher help a lot, but the work is still extensive, and the tools are costly. Unstructured tools generally let you easily control the design of your PDF output.

Publish to HTML5, MOBI, EPUB, and More

With DITA content, output for dynamic (responsive) HTML, CHM, EPUB, and other non-PDF formats is excellent, with many good transforms—templates for transforming the DITA source content to the desired output—freely available from the DITA Open Toolkit. Unstructured tools vary in their support for non-PDF outputs.

Summary: Tools Support for Features

To sum up, three popular Tech Pubs tools provide varying degrees of support for the best practices associated with structured authoring, authoring standards, and structured tools (which sometimes require additional tools for more complete implementations).

 Frame (2015)	 Word (2010)	 Flare (11)
<p>The File > Publish feature lets you create Responsive HTML 5, Mobile App, WebHelp, EPUB, Kindle, and more. You can configure many aspects of the transformation:</p> <ul style="list-style-type: none"> • Style mapping from FrameMaker styles (para, char, etc.) to CSS classes. • Topic split/merge settings. 	<ul style="list-style-type: none"> • Word has a Save As Web Page function, but it’s widely derided. • Best approach is a 3rd party solution such as WebWorks ePublisher, http://www.webworks.com/, to enable single-sourcing for multiple outputs. 	<ul style="list-style-type: none"> • Easy to produce a variety of online, print outputs, and even Microsoft Word (which can be helpful to collect review comments). • Supports CSS @media statements, which let you assert different CSS values based on media type or display type. See References.

Figure 11. Publish to non-PDF.

Feature	 Frame (2015)	 Word (2010)	 Flare (11)
Topic-based structure			
Consistent topic structure			
<shortdesc> mimic			
Content reuse: whole topics			
Content reuse: fragments			
Content reuse: slight diffs			
Semantic styling			
Para-styled navigation aids			
Metadata			
Multi-channel publishing			

Figure 12. Summary of non-structured tools' support of structured writing features.

Conclusion

Full-on structured writing is based on useful principles, but in its purest form is technically demanding. However, you can borrow useful bits from structured authoring to the extent that your existing tools and workflows support it. No matter your tool—whether you get to choose or it's imposed upon you—look to the tool's advanced features and integrations with other tools to let you adopt some structured/DITA best practices. But when you adopt structured authoring mechanisms for non-structured tools, recognize that the tradeoff for a less technically demanding approach is the necessity for more editorial review.

Resources

Structured Authoring

Topic-based authoring: https://en.wikipedia.org/wiki/Topic-based_authoring

Importance of short descriptions, an OASIS White Paper, *DITA Feature Article: Short Descriptions Shouldn't Be a Tall Order: Writing Effective Short Descriptions* (18 March 2016): <https://www.oasis-open.org/committees/download.php/57803/>

DITA-Adoption_2016_Writing-Effective-Short-Descriptions.pdf

DITA Best Practices, A Roadmap for Writing, Editing, and Architecting in DITA; by Laura Bellamy, Michelle Carey, Jenifer Schlotfeldt, © 2012

Metadata

Uses for metadata: <https://en.wikipedia.org/wiki/Metadata#Creation>

Historical information about the Extensible Metadata Platform (XMP): http://www.ipc.org/std/lptc4xmpCore/1.0/documentation/lptc4xmpCore_1.0-doc-CpanelsUserGuide_13.pdf

Standards and Specifications

DocBook: <http://www.docbook.org/>

DITA: <http://dita.xml.org/>

S1000D: <http://www.s1000d.net/>

XHTML, CSS specifications: <https://www.w3.org/>

DTD: https://en.wikipedia.org/wiki/Document_type_definition

RELAX NG schema language for XML: https://en.wikipedia.org/wiki/RELAX_NG

Schematron rule-based validation language: <https://en.wikipedia.org/wiki/Schematron>

Tools

oXygen XML Editor: <https://www.oxygenxml.com/>

DITA Open Toolkit: <http://www.dita-ot.org/>

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

FrameMaker

Using folders as containers for topics: <http://blogs.adobe.com/techcomm/2014/07/why-upgrade-from-fm7-8-or-9-5-book-building-and-external-references.html>

Using Frame as a little CMS: <http://www.mail-archive.com/frameusers@lists.frameusers.com/msg65018.html>

BookVars plugin to manage variables: <http://leximation.com/tools/info/bookvars.php>

FrameUsers.com website and email group: <http://www.frameusers.com/community/>

Microsoft Word

MS Word Helpers group on LinkedIn: <https://www.linkedin.com/groups/1851284/>

Word Master Documents: <http://www.addbalance.com/word/masterdocuments.htm>, <http://techwhirl.com/wp-content/uploads/2010/09/Microsoft-Word-masterdocs.pdf>, <http://word.mvps.org/FAQs/general/whymasterdocscorrupt.htm>

DitaExchange, which integrates with Microsoft Word: <http://ditaexchange.com/>; and a mention of DitaExchange by Scott Abel, the Content Wrangler, <http://thecontentwrangler.com/2011/06/01/yes-you-can-do-dita-with-microsoft-office-and-sharepoint/>

Tips for using AutoText in Microsoft Word 2010: <http://www.groovypost.com/howto/microsoft/how-to-guide-for-using-autotext-quick-parts-in-office-2010/>

Detailed information about Microsoft Word Building Blocks: http://gregmaxey.mvps.org/word_tip_pages/building_blocks_autotext.html

How to fake conditional text in Microsoft Word: <http://www.technicalcommunicationcenter.com/2013/12/05/how-to-use-ms-words-hide-text-function-to-create-conditional-text-in-word/>

SmartDocs, which provides single-sourcing, variables, conditional text features, and more for Microsoft Word: <http://www.thirtysix.net/>, <http://www.thirtysix.net/smartdocs/features>

RiverFloe, a “document generation machine for Microsoft Word”: <http://www.riverfloe.com>

Clio, which provides doc solutions for Microsoft Word, for the legal industry: *Using Conditional Text in Document Automation*, <https://support.goclio.com/hc/en-us/articles/204459577-Tutorial-Using-Conditional-Text-in-Document-Automation>

Details about creating paragraphs with automatic lead-in text: <http://www.shaunakelly.com/word/numbering/numbering20072010.html>; look for “Now we tell Word about the numbering itself for Level 1”.

How to add the Prepare Document functions to the ribbon: <http://www.addictivetips.com/microsoft-office/prepare-word-2010-document-for-distribution-prepare-menu/>

Blog post about using Git for Word docs: <http://blog.martinfenner.org/2014/08/25/using-microsoft-word-with-git/>

MadCap Flare

Well-formed document in XHTML: https://en.wikipedia.org/wiki/Well-formed_document, http://docstore.mik.ua/oreilly/web2/wdesign/ch31_04.htm

Using CSS @media statements: <https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

UTF-8 format, used by Flare: <http://microformats.org/wiki/using-utf-8>

Users of MadCap Flare group on LinkedIn: <https://www.linkedin.com/groups/86373>

Author Contact Information

Monique Semp
Principal
Write Quick, Inc.
San Francisco Bay Area, CA
707.769.9541

Author Biography

As a former engineer, Monique’s early tech writing was user manuals for the applications she designed and coded. After becoming a tech writer, she was often the “tools person” on her team, extending the team’s tools’ capabilities to make it easier to create better, more consistent content. As Monique matured as a writer and honed her doc design and writing skills,

she learned a diverse range of tools and technologies. Structured authoring and its implementation in the DITA spec/commercial tooling was a natural evolution that continues to influence her work, even for contracts requiring unstructured tools. She finds it gratifying to share her realizations that although writing in DITA is great, it can be a solution looking for a problem, and that frequently the answer is to leverage the best practices of structure while using traditional, unstructured tools.



Simon, Kelley

Creating a Free Intranet Using Drupal

In 2014, Royal Credit Union, a mid-sized credit union located in Western Wisconsin and Eastern Minnesota, designed and implemented a new team member portal using an open-sourced software called Drupal. This case study outlines the process Royal's project team followed to both identify Drupal and configure and design the new site.

Royal Credit Union is a not for profit organization located in Wisconsin and Minnesota, with 26 branches, 180,000 members, and about 590 team members. This case study explores how Royal went from drowning in email to creating a team member portal using Drupal that has become our main source of business communication. Best of all, we did it for free.

At Royal, all internal communication was sent via email and a shared folder system existed on our servers that stored documents and information for all areas of the credit union. Large amounts of information were flowing through Outlook, and as the credit union grew, so did the amount of emails that each team member received daily. Royal had an intranet site that housed internal procedures and shortcuts to network drive locations.

Limitations of the old site include:

- Repository only
- Only two users were able to publish content to ensure security of content
- Limited search options
- Needed to be entirely supported by the IT department, as the tool was developed internally.

Royal's content management challenges were similar to other mid-size organizations using email as both communication tool and information repository. Royal team members needed to promptly retrieve accurate information, review all information about a topic, publish their own content, and interact with all

other areas of the credit union. In 2013 Royal began exploring different content management systems.

Our Goal

Royal's goal was to design and implement a team member portal that addressed the communication needs of all users, while reflecting our core values.

Specifically, we wanted to:

- Centralize information relevant to all team members.
- Promote inter-department communication.
- Reduce the amount of email received.
- Reflect Royal's Core Purpose (To create a positive impact in the lives we touch) and our Core Values.

Research

When research began, the intention was to purchase a content management solution, with a preliminary budget of about \$15,000. The project team began by compiling internal information and requirements needed for our new system.

Tasks included:

- Assembling a project team that included areas that created corporate-level communication regularly. In Royal's case this was Talent Services (HR), Training, Marketing, Administration,

- IT, and Operations.
- Inventorying all documents currently on the static intranet site.
- Categorizing emails using the “all team members” distribution list.
- Creating an Information Use questionnaire.
- Holding focus groups with the majority of departments. We felt this was important in that a) we needed to know what resources team members used daily for their jobs, and b) we wanted to promote buy-in.

Once we received the results of our questionnaire and focus groups, the project team began an initial discussion of scope for the project, specifically:

- What are we specifically trying to achieve?
- What kind of look and feel should the site have?
- What basic information components need to be included on the site in Phase 1?
- What do our users need to/want to do (roles and authorizations) on the site?

It is important to note the content management project was considered a corporate initiative. We had the buy-in from our executive team, allowing the technical writing department to get cooperation and resources from all areas during the research phase.

Software Search

Once we had requirements identified and documented, we narrowed down the search to a content management company specializing in intranets for financials. As we progressed with our research, it became apparent that while this company could give us a great site, it was not as agile and as easy to customize as other options. As an organization, we were drawn to flexibility and customization in design and function, and that is how Drupal became a viable option.

Drupal is an open-source content management software that uses modularity as one of its core principles (meaning that there are core modules that come with Drupal, and then the user adds modules as they want to increase functionality). Drupal can create dynamic web experiences, integrate digital frameworks, and has excellent security. Anyone can download the Drupal starter kit, as it is open-sourced

software. There are no licensing fees and it is totally free.

Royal compared Drupal to “out of the box” content management solutions and discovered the following:

- Drupal is a blank slate, meaning that we could set it up any way we wanted, for free.
- Additional functionality in Drupal meant downloading a new module for no additional cost.
- We had control over all areas of the site, including how and where content displayed, who could use it and how, and themes (design templates).
- Drupal is designed with a WYSIWYG editor environment, meaning that non-developers could design and administer the site.
- Drupal is flexible enough to work with existing infrastructure, such as Active Directory, which was utilized for user authentication and to feed information into the corporate directory.
- Drupal met our requirements regarding consolidating information, multiple author publishing, having a robust and secure search.
- Drupal definitely fit into our budget!

The biggest disadvantage to using Drupal is also its biggest strength. Once Drupal’s core modules are downloaded, it must be built from the bottom up. Everything needs to be configured. That can be daunting to new users, or people who are not familiar with programming. Fortunately Drupal.org contains a vast online library of references, instructions, and user forums.

Royal was fortunate that our systems administrator had used Drupal in the past, was familiar with how it worked, and knew how it could be configured. When the final content management comparison was presented to our executive team, the idea that we could create our own intranet site, just the way we wanted, for no licensing fees or additional future costs made the decision to use Drupal pretty straightforward.

Work Breakdown Structure

Once the decision was made to move ahead with Drupal, we created our work breakdown structure,

Systems Administrator	Technical Writers
Install Drupal Core, preferably on a LAMP Stack (Linux, Apache, MySQL, and PHP).	Determine what content will be on the portal and how it should display. Create guidelines.
Add standard modules to Drupal, such as Views (Drupal 7), c-Tools, WYSIWYG Editor, PathAuto, Token, Date, and Theme.	Determine what roles and authorizations users will have.
Configure modules, such as selecting a basic theme and building on it.	Learn the basics of Drupal: nodes, content types, user authorizations, views, and themes.
Configure basic authorizations and roles.	Develop content types, basic pages, standard views, and web forms.
Identify integration opportunities (such as with our current phone system, which contains our corporate directory).	Meet with any department that will have a department page and work with them on a design.
Create site back-up procedure.	Create training resources for Drupal.

Figure 1. Tasks for the systems administrator and technical writers

outlining the tasks for both the systems administrator and technical writers.

Content Design Considerations

As the system administrator was configuring our Drupal site, the technical writing department worked on information design considerations. We first identified all team member information flowing

through email and categorized it, based on author, audience, and importance. Some information, such as time-sensitive or job-specific messages would continue to be sent through email, but most messages were categorized as suitable portal content. A snapshot of our content design is below.

A node is one of Drupal’s basic building blocks, and most content on a site is a node. These nodes are created using administrator-defined Content Types (Corporate Message, Announcement), and can also have

categories assigned to them. Content Types (and categories) instruct Drupal on how a specific type of node should display. An example of categories under the Corporate Message content type and how they function is below.

When signed into the site, authors pick what type of content they need to post (based on their authorizations), and that content will display as

Content Type in Drupal	Definition	Author (Role in Drupal)	Examples
Corporate Message	Business related communication meant for all team members	Corporate Leaders	Messages from our executives, marketing, talent services, training, and systems departments.
Announcements	Non-business related communication posted to promote Royal’s Core Purpose and Values.	Authorized Users (All team members signed into Drupal)	Job Changes, New Hire Welcomes, Thank You’s, Awards/Recognition, Fundraising, Community Service Opportunities, Photo Albums, Condolences, Anniversaries, Births

Figure 2. A snapshot of our content design

configured. Drupal makes the setup of content types, categories, views, basic pages, blocks, and roles straightforward and easy to maintain.

Results

After three months of work, the new Royal Credit Union team member portal was launched in the spring of 2014. The project team achieved the goal of implementing a portal that addressed the communication needs of all users, while reflecting our core values. It became the home page for all users.

With Drupal’s robust functionality, Royal introduced information consolidation and new functionality to our team members in a site that included:

- A main menu with quick links to common internal documents and external websites.
- Department pages for areas that post corporate messages.

- Branch pages that consolidated all commonly asked questions about a branch and their messaging.
- Corporate Directory, which included not only their pertinent contact information and a picture of the team member, but custom field information, such as if they spoke a different language, if they were CPR certified, and if they were a notary or signature guarantee. These were fields that could then be easily searched within Drupal.
- Project pages, consolidating all the information needed to support initiatives and product rollouts, resulting in team members receiving accurate information quickly and easily.

The portal has become a powerful business communication tool, but it also promotes Royal’s corporate culture.

Some examples of this include:

- All posts include the team member’s picture, personalizing the message.

Category	Primary Author (Role)	Content (Node) Example	Displays on Site
Advocacy	Community Engagement Team Members	CUNA Government Affairs Conference 2017 Update	Home Page Center Community Engagement Home Page
Benefits	Talent Services Team Members	Retirement Seminar Announced!	Home Page Center, Talent Services Home Page
Births	Talent Services Team Members	Congratulations-It’s a Girl!	Team Member Corner, side bar first block
Development	Training Department Team Members	Secure Password Training Announced.	Home Page Center, Training Department Home Page
General Corporate Message	Any Corporate Leader	Toastmasters Meeting Tomorrow	Home Page Center, middle of the page.

Figure 3. Example of categories under the Corporate Message content type and how they function is below

- New hires are welcomed with a message by their supervisor on the portal, and their new co-workers then use the comments feature to welcome them to the department.
- New jobs, promotions, work anniversaries, and merit achievements are recognized.
- A polls module allows for a fun and interesting way to promote engagement by “taking the pulse” of team members with a single question.

Challenges

While out of the box content management solutions give you a design framework to start with, Drupal’s modular format promotes design from the ground up. While creating nodes (content), blocks, and basic pages

are straightforward, categorizing them, connecting them, and creating views to display them can be a challenge. Fortunately the resources on Drupal.org, as well as training sessions and user tutorials on YouTube can assist you in pretty much any task you would like to accomplish in Drupal.

Royal also found that ongoing development of the portal came at a slower pace, only because it takes time to build up knowledge and implement new functionality. If Royal would have gone with an out-of-the-box system (with the technical support of a company) we would be able to implement new features faster, but we would also be paying for the assistance.

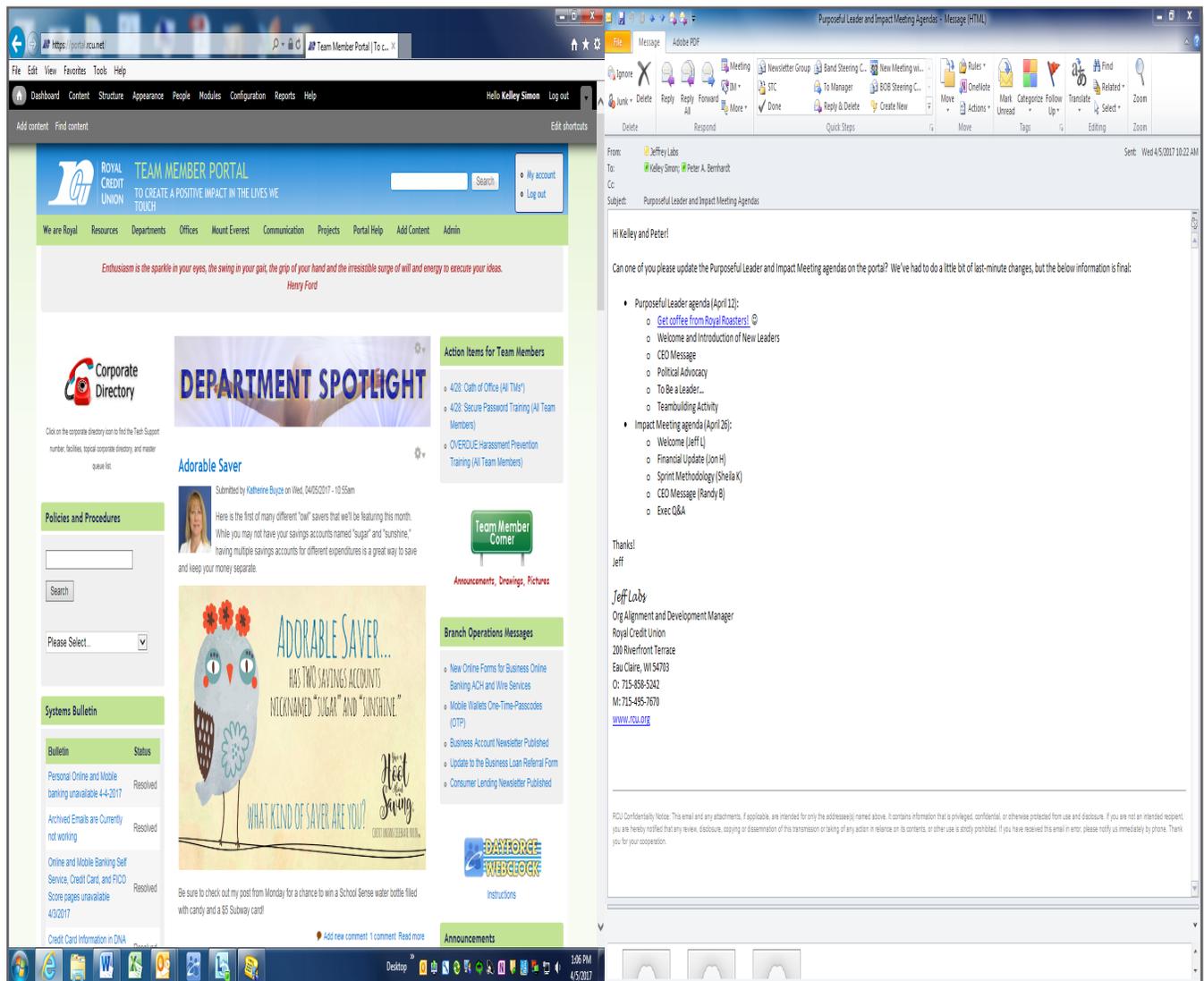


Figure 4.: Example of Royal’s Team Member Portal Home Page

Conclusion

In conclusion, Royal's choice to join the Drupal global community was a solid business decision. It fulfilled our identified end user and administration requirements, and we continue to learn new ways to promote our corporate culture using this versatile tool. Royal highly suggests considering Drupal as an option when reviewing your own business communication needs.

Resources

Byron, Angela., Berry, Addison., De Bondt, Bruno.
Using Drupal, Second Addition (Sebastopol, CA:
O'Reilly Media, Inc.), 2012

Drupal.org. <https://www.drupal.org/> (accessed April 3, 2017).

Author Contact Information

Name Kelley Simon
Technical Writer II
Royal Credit Union
200 Riverfront Terrace
Eau Claire, WI 54703
715.833.8104
kelley.simon@rcu.org

Author Biography

Kelley Simon graduated from the University of Wisconsin-Eau Claire in 1999 with a Bachelor's in English: Technical Writing emphasis. She has worked at Royal Credit Union for 17 years as a procedure writer, project manager, and administrator of Royal's team member portal. Ms. Simon also works as a grant writer for non-profit organizations within her community.



Stevenson, Sean

Getting to Compliant: Responding to Government Requests for Proposal (RFPs) for the Technical Communicator

There are two kinds of technical communicators: those who have been thrown into Business Development, and those who will. Technical communicators already have the skills they need to write superior responses to government Requests for Proposal (RFP), they just need adequate context. This proceedings paper presents the processes and vocabulary necessary to jump into proposal writing. The goal of this paper is to prepare a technical communicator to walk into a proposal and have a strong sense of what they will be expected to do, and where they can succeed. This paper is aimed at the technical communicator at a large business. Those at small businesses will use a subset of what is presented here.

Why Work in Business Development?

The average technical communicator begins their career in Business Development (BD) by being sent there by their boss.

Some stay, and there are some notable reasons why:

- **Storytelling.** It is hard to tell a story and get paid for it in business, but you can in BD. When you walk into the proposal room, an excellent first question to ask is, what kind of story are we trying to tell?
- **Respect.** Technical writing and training is firmly in the periphery of a company's interest. You can move to the center by winning new business—you will work side by side with executives, and can lend them your expertise. You will get high visibility.
- **Competition.** There's excitement in winning. It gets the adrenaline pumping. If you're an adrenaline junkie who wrote their college papers late at night you'll do well. Luckily, there's also

plenty of structure in the process to tame even the worst procrastinator.

Bring Your Strengths to Bear

When you walk into the proposal call yourself the writing subject matter expert (SME). As a technical communicator, you have expertise in at least one and possibly all three of the following skills: writing, editing, and desktop publishing. You know how intimately related these are. Most business writing is reused source material from another project edited and formatted to meet the current requirements. Proposals are no different.

The major criticism levelled against BD writing is that is jargon filled. This is fair and it is your job to fight it.

As the writing SME, concentrate on these elements and you will have more than paid for yourself:

- Determine if a legitimate approach to a problem is in the text you are reading. Don't be afraid to tell someone there is no substance in a section. Of course, you will then be tasked to fix it.

- Please don't say "world-class" or "industry leading." These are meaningless because they are impossible to prove.
- Make sure that the writing provides proof of where your company has performed similar requirements.
- The writing itself must be logical and consistent, underscoring that the approach is logical and consistent.
- Consider leading off a section with a story of a similar problem and how your company solved it.
- Don't talk about your company when you can address your customer's needs.
- You are uniquely qualified to know a good procedure from a bad one. Look for the missing steps in what you are reading. Don't be afraid to make changes.

Any large proposal will have a technical editor do a pass on it. This could be you.

Jump in and concentrate on the following areas:

- Insist a section is clear, and where it falls short, make the writer explain it to you.
- Page count is critical. Pare sentences down to their essence.
- Be a leader. Insist that terms be used consistently. Often, a proposal room will have what's called a "Wall of Truth." It's a style guide for that proposal. Put things there that are agreed upon, like the name of your team, the name of the customer, the name of your program, etc.
- Writers on proposals work until you ring the bell. You must tell them that pens are down and that you need time to edit their sections.

Desktop publisher is an overlooked role, but every major proposal will have a desktop publisher.

This is a good place to start your career in BD, and steadily add editing and writing support:

- Your knowledge of Microsoft Word (styles, heading numbers, placing figures, etc.) helps immensely.
- Configuration management is key. The desktop publisher is the enforcer. If you are not walking into a situation with a defined folder structure and processes, create them.

Minimize Your Weaknesses

The aim of responding to RFPs is not the same as technical communication. The aim of technical communication is to communicate. The aim of proposal writing is to write a winning proposal. Strive for quality in your proposal, but remember that schedule pressures dictate that an 80% solution can and probably will win. This initial proposal is probably your only shot. Make it count. These limitations can drive a technical communicator crazy, but not respecting them will be seen as a weakness on your part.

The government will not consider late proposals. If you want to work in BD, you cannot have a cavalier attitude to deadlines. This goes for reviews as well—prepare for a review as hard you would to turn the final proposal in.

Politics plays a big role in BD, and technical communicators may not be experienced with the stakes involved. The larger a proposal the more politics plays a role, especially in a loss. Proposals can be very subjective. You failed to convince the government of the greater value of your proposal. Take part in a lessons learned exercise if your company does it, and see how you can better your proposals for next time.

Roles and Resources

Here are the major roles on a proposal, and a brief description of what they do:

- Capture manager – engages customer in advance of RFP, develops win themes
- Proposal manager – leads all aspects of proposal from developing kickoff to delivery
- Volume leads – large proposals have these for technical, management, past performance
- Pricing lead – develops price (what really wins you a job, but don't tell anyone)
- Writer – typically either a technical writer or a SME from operations or engineering
- Graphic artist – creates approximately one graphic or graphical element per page
- Configuration manager/ desktop publisher – ensures compliant formatting

- Editor – edits for style, clarity, consistency, and grammar
- Proposal coordinator – administrative position, entry-level

When you show up, you will be working for the proposal manager. The proposal manager oversees day to day proposal operations. That person will assign you to a volume and you will report to a volume lead. Make sure you read Sections L and M of the RFP so you can help without veering off into non-compliance. From there, follow your instincts to provide a fresh perspective on what may not be working in the proposal. This will also help your fellow writers. Many hands make light work.

The RFP

Most United States Government proposals take the same form. They are made up on lettered sections, each of which has its own focus.

Here are the major sections:

- Section B – Supplies or Services and Price/Costs
- Section C – Statement of Work
- Section F – Deliverables
- Section K – Representations and Certifications
- Section L – Proposal Preparation Instructions
- Section M – Proposal Evaluation Criteria

The scope of this paper necessitates focusing on two sections: Sections L and M.

The first thing you should do when you get assigned to a proposal is to ask to see the RFP. Then read Section L. Section L tells you what you need to address, how much space you have, how it will look, and when it is due.

Most proposals include these major topics that require a response: past performance, technical approach, management approach, and price. Past performance is proof that your company has experience providing a similar product or service. For example, to meet this requirement a company may provide three projects of similar size and scope to the project being bid on. If Section L lists past performance first, make that the first part of your proposal. As a neophyte to proposals,

a technical communicator will have very little to do with the pricing volume.

A management volume addresses how to manage the program: Responsibility, Accountability, and Authority (RAA), Organizational Chart, schedule, etc. The technical volume addresses the technical details of your company's approach, detailing how your approach to accomplishing the requirements will lead to a high-quality product.

The next thing you read is Section M. Section M will tell you how they will evaluate your proposal. As a writer and desktop publisher you will take your cue from Section L, making sure you address the sections and subsections spelled out there. As an editor, pay close attention to Section M. A substantive approach will address the evaluation criteria, showing how the company will solve the major problems in the RFP.

Compliance

Remember, the government is telling you what they want. Compliance to the requirements of the RFP drive your proposal. Many proposals emphasize a company's expertise and not HOW they plan to meet the government's requirements. This is called non-compliance, and if your proposal is non-compliant, you cannot win.

Compliance entails addressing the criteria in Section L and Section M with a substantive approach. It is very important to clearly show how you are compliant. When the Source Selection Committee evaluates your proposal, a compliant proposal with no significant strengths is rated more highly than a non-compliant proposal with significant strengths. Respect the contracting specialists who drafted the RFP, and assume that they have structured it the way it is for a reason. Use your skills as a reader to see if what is on the page answers the spirit and the letter of the RFP. Use your writing skills to edit and rearrange your text to clearly map to the requirement.

Schedule

The government defines the schedule. The scope of this paper allows us to address Draft and Final RFPs. It's important to remember that an RFP is, in fact, a late step in the government acquisition process. Before this, the government has done market research,

defined its acquisition strategy, and developed a statement of work (which becomes part of the draft RFP). There is usually also a draft of Sections L and M, but not always. If you are called in, assume there is a draft section L and section M.

When the Final RFP is released, then the schedule tightens significantly. So, thank your stars for a Draft RFP. It gives you some time to get ahead of the game. Don't waste it. Then again, most technical communicators aren't being brought in during the draft. You are going to be brought in according to the schedule in Figure 1.

These are the major milestones in a proposal. The review titles here are industry standard. If you say Red Team everyone will know what you mean. The kickoff is scheduled according to the best estimate of when the Final RFP is going to be released. Everyone would prefer to be brought in at the front of this schedule, but the most likely place you will be brought in is right after a Pink Team where the proposal was deemed to be in trouble. Someone said, let's get a writer/editor in here to help turn our storyboards into text. That is a lot of work and days 20 to 30 will seem like forever.

If you are brought in after Red Team that means the proposal is in trouble. Unless you are seasoned, this is too late in the process to provide significant help. Any help you provide will be appreciated, but a bad Red Team is difficult to recover from. At this point, pray the government is late answering questions about the RFP and extends the due date of the proposal.

Your skills are essential to bringing the proposal home and delivering on Day 30. Expect to work late the week before the proposal is due. Your job is to ensure that last minute changes meet your quality standards for clarity and compliance. Make sure to celebrate the day the proposal is delivered.

Conclusion

Writing proposals is an ideal adjacent skill for any technical communicator to learn. This paper has endeavored to provide insight into what strengths the technical communicator brings, how to read and respond to an RFP, navigating a proposal schedule, and the other functions they will interact with. Knowing these can aid the technical communicator in having advanced situational awareness when they enter the proposal room. Perhaps they may find it suits them better than technical writing. Business Development aids your company in a way that other areas never can. Perhaps at this point of your career, you are ready for this level of responsibility. Also, it's fun to win.

Resources

Newman, Larry. Shipley Proposal Guide, Third Edition (Farrington, UT: Shipley Associates), 2008.

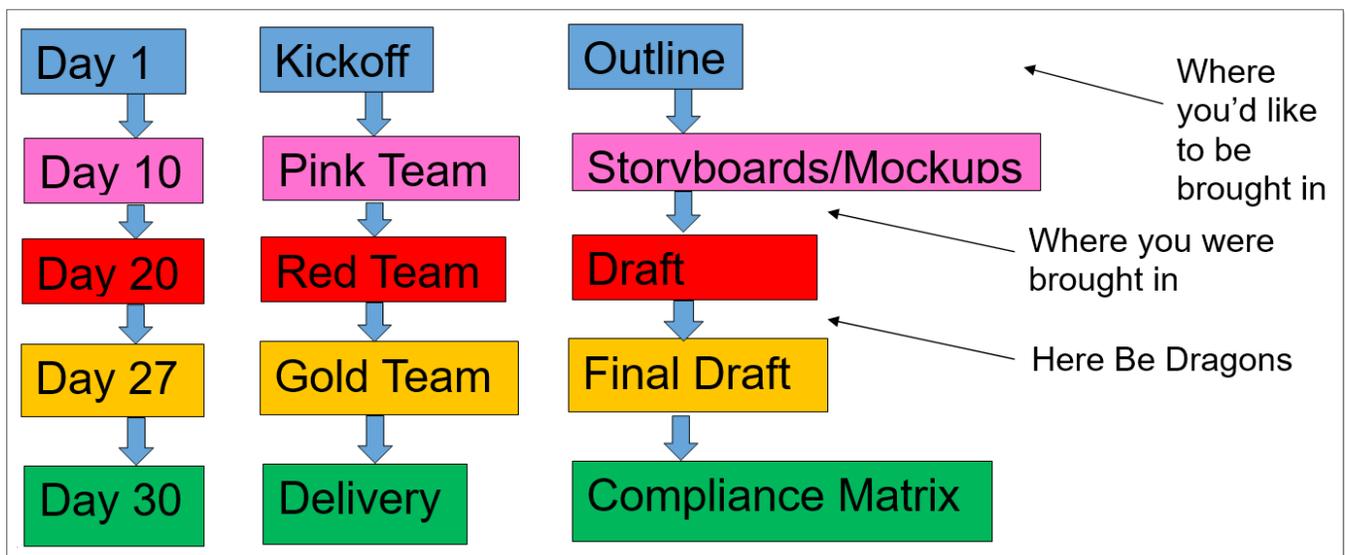


Figure 1. This is a typical schedule for a proposal with a 30-day turnaround.

Author Contact Information

Sean Stevenson, Immediate Past President, STC
Washington DC-Baltimore
Proposal Manager
EMCOR Government Services
2800 Crystal Drive, Suite 600
Arlington, VA 22202
571.403.8900

Author Biography

Sean Stevenson, Immediate Past President of the Washington, DC-Baltimore chapter of STC, is a technical communicator working in business development. He started as a creative writer who taught writing on the side. Realizing he needed a better career, he took a job at Northrop Grumman in Maryland as a Technical Editor in 2008. Over time, he found his way into business development and in 2013, took a job as a Proposal Manager at EMCOR Government Services in Arlington, VA. Sean has an MFA in playwriting from the Actors Studio Drama School and a BA in Humanities from the University of Chicago. He teaches yoga in his spare time and wants to teach English as a Foreign Language again.



Strimling, Yoel

So You Think You Know What Your Readers Want?

This paper presents the results of an applied research study to determine how readers define documentation quality (DQ), as well as how writers assume readers define it. Using a narrow yet comprehensive set of distinct, empirically based information quality (IQ) dimensions that cover all categories of IQ – Intrinsic, Representational, Contextual, and Accessibility – questionnaires were sent asking respondents to rank these IQ dimensions, and the data was analyzed statistically. The outcome of this research indicates that writers make some significant mistakes when assuming how readers define DQ – but there are some concrete steps that can be taken to address them.

Introduction

As technical communicators, we put a lot of time and effort into creating the highest quality documentation we can. We write because we want to help our readers to do the tasks they need to do or to understand the concepts they need to know.

Because we are professionals, we take pride in our work and want it to be the best it can be. But how do we know if what we are writing is what our readers want? How do we know that the information that we are sharing with our audience is helping them do or know what they need to do or know? We might be writing documentation with one standard in mind, and be satisfied with it, yet our readers might look at the same documentation and be very unsatisfied. A disconnect like this between what we are producing and what our readers actually want makes it very difficult to justify writing documentation at all – why should we write things nobody wants?

The best way to align ourselves with our audience's needs is to get direct, meaningful, and actionable feedback from them. But this is not always possible, for one reason or another. Instead, we often end up relying on our "gut instincts" and assume that readers define high-quality documentation in the same way that we do. In lieu of feedback, what we need is a proven model of how readers actually define

documentation quality (DQ), which we can then use to ensure that what we produce meets this definition.

Defining DQ

Numerous attempts have been made to define "quality", and it is beyond the scope of this paper to go into detail for each one. Reeves & Bednar (1994) conclude that there is no such thing as a "universal definition" of quality – different definitions are appropriate under different circumstances. However, all quality definitions point to the same thing – it is the user who is the final arbiter of what "quality" is and what it is not.

This is especially true for documentation, which is always written for a potential audience, and must always keep their needs in mind (Barnum & Carliner, 1993; Bursaw, Alfred, & Olliu, 1993; Redish, 1993; Robinson & Etter, 2000) – for documentation, it is the reader who is the final arbiter of what "quality" is and what it is not (Mead, 1998).

Current Definitions of DQ

A review of the literature shows that there are many different possible definitions for DQ. However, there are a number of problems with these definitions.

First of all, many of the traits are vague – what is meant by “familiar to the reader” (Brown, 1995) or “worthwhile” (Bush, 2001)? Does “easy to read” (Quesenbery, 2001) mean the same thing as “highly readable” (Betz, 1996), “interesting to read” (Spyridakis, 2000) or “readable” (Gregory, 2004)? Is the ISO/IEC26514:2008 standard’s “easy to understand” the same as the DQTI’s (Hargis, et al., 2004) “easy to understand”?

Secondly, it is almost impossible to objectively measure some of these quality traits. While terms such as “complete”, “accurate”, or “clear” (for example, in Tarutz, 1992) are relatively straightforward, definitions used by Albers (2005) or Manning (2008) are completely subjective. Pirsig (1974), in his classic *Zen and the Art of Motorcycle Maintenance*, says that “quality is the relationship of the two [objective quality and subjective quality] with each other”, so we cannot have a definition of DQ with only one type and not the other.

Lastly, the fact that there are so many different definitions of quality is itself a problem. While it is true that different definitions are appropriate under different circumstances, there is no single trait in any of the definitions that can be found in all of them. It is highly unlikely that the definition of DQ changes so much from situation to situation (and from reader to reader) that there is not at least some overlap between them.

Criteria Needed to Define DQ

To properly define DQ, we must meet the following criteria:

- The definition must be from the reader’s point of view: Because it is the readers alone who determine if the document we give them is high quality or not, any definition of DQ must come from the readers’ perspective. Writers can come up with any number of quality attributes that they think are important, but at the end of the day, what they think is not as important as what the readers think.
- The definition must be clear and unequivocal: Both readers and writers have to “be on the same page” when it comes to what makes a document high quality. Misunderstandings of what readers actually want from the documentation are a recipe for unhappy readers.

- The definition must cover all possible aspects of “quality”: “Quality” is a multidimensional concept, and we must be sure that any attempt to define it is as comprehensive as possible. A definition that emphasizes one dimension over another, or leaves one out altogether, cannot be considered to be a usable definition.
- The definition must have solid empirical backing: To be considered a valid definition of DQ, serious research must be done to give it the proper theoretical underpinnings. Years of experience or anecdotal evidence can act as a starting point, but if we are serious about our professionalism and our documentation, we need more.

Building a Comprehensive Definition of DQ

To meet all of these DQ criteria, I turned to the information quality (IQ) literature, which puts a strong emphasis on empirical studies from the users’ point of view.

Using Wang & Strong’s Data Quality Model

In 1996, Dr. Richard Wang (co-director of the MIT Total Data Quality Management Program) and Dr. Diane Strong (director of the Management Information Systems Program at the Worcester Polytechnic Institute) published a groundbreaking study of what “data quality” meant to data consumers, which they used to create a “hierarchical, conceptual framework of data quality from the data consumer’s perspective” (Wang & Strong, 1996). Their underlying assumption was that, to improve data quality, they needed to understand what “data quality” meant to data consumers – data quality cannot be approached intuitively or theoretically because these do not truly capture the voice of the data consumer.

They collected ranked data quality attributes from data consumers, consolidated them into 15 dimensions, and then grouped these into the following four categories:

- **Intrinsic Data Quality:** Data must have quality in its own right. This category is made up of the following dimensions:

- Accurate: The data is correct, reliable, and certified free of error.
 - Believable: The data is true, real, and credible.
 - Objective: The data is unbiased (unprejudiced) and impartial.
 - Reputable: The data is trusted or highly regarded in terms of its source or content.
- Contextual Data Quality: Data must be considered within the context of the task at hand. This category is made up of the following dimensions:
 - The Appropriate Amount: The quantity or volume of the available data is appropriate.
 - Complete: The data is of sufficient breadth, depth, and scope for the task at hand.
 - Relevant: The data is applicable and helpful for the task at hand.
 - Timely: The age of the data is appropriate for the task at hand.
 - Valuable: The data is beneficial and provides advantages from its use.
 - Representational Data Quality: Data must be well represented. This category is made up of the following dimensions:
 - Concise: The data is compactly represented without being overwhelming (that is, it is brief in presentation, yet complete and to the point).
 - Consistent: The data is always presented in the same format and is compatible with previous data.
 - Easy to Understand: The data is clear, without ambiguity, and easily comprehended.
 - Interpretable: The data is in an appropriate language and units, and the definitions are clear.
 - Accessibility Data Quality: Data must be easy to retrieve. This category is made up of the following dimensions:
 - Accessible: The data is available or easily and quickly retrievable.
 - Secure: Access to the data can be restricted, and hence, kept secure.

Based on these categories, Wang & Strong concluded that high-quality data must be:

- Intrinsically good
- Clearly represented
- Contextually appropriate for the task
- Accessible to the consumer

They suggested that this framework could be used to develop questionnaires to measure perceived “data quality”, and that the categories and their underlying dimensions could be used as the constructs to be measured.

Based on the results of Wang & Strong’s “data quality” study, it seems to be very easy and appropriate to apply what they propose to the question of measuring DQ.

To understand DQ, we cannot rely on an intuitive or theoretical approach; we must get to the “data consumers” – that is, our readers. To improve DQ, we need to understand what DQ really means to our readers.

And, as with data quality, high-quality documentation must be:

- Intrinsically good
- Clearly represented
- Contextually appropriate for the task
- Accessible to the reader

Data Quality Is Not the Same as DQ

It is important to make a point clear here before we continue comparing data quality to DQ, however. Wang & Strong’s study focused on “data quality” – are “data” and “documentation” synonymous and interchangeable?

“Data” is abstract, raw, and meaningless without context (Kumar, 2009). However, when data is organized in a logical way and given context that can be understood by an “interpretant”, then it becomes “information” (Chisholm, 2012). In other words, “information” is “data” in a meaningful form.

Documentation, then, is not “data”, but rather “information”. Documentation is intended to be used

by readers (“interpretants”) in a particular context for a particular reason.

While Wang & Strong’s originally stated object is “data quality”, it would be more accurate to call it “information quality” (Lee, Strong, Kahn, & Wang, 2002) – as soon as data consumers (again, “interpretants”) are asked to rate characteristics of data, it can no longer really be called “data”. Data is not intrinsically “wrong” or “right”, it just “is” (Ericson, 2012). (In fact, the “MIT Total Data Quality Management Program” that Dr. Wang is the director of has been renamed the “MIT Information Quality Program”.)

Wang & Strong’s framework is then really a framework for measuring the quality of both data and information from a data and information consumer’s point of view (Strong, 2015). If so, the parallels between it and DQ are valid, and we should be able to use their framework to create a model for accurately measuring what our readers consider to be high-quality

documentation – and then make plans to improve what needs to be improved.

Methods

Questionnaires

Based on the assumption that Wang & Strong’s IQ categories and their underlying dimensions were valid and applicable to DQ, questionnaires were developed to measure perceptions of what makes high-quality documentation. Some minor modifications were made to the IQ dimension definitions (the word “data” in the definitions was changed to “information in the documentation.”).

Two questionnaires were created – one for writers (<https://www.surveymonkey.com/r/PDPH9LX>) and one for readers (<http://www.surveymonkey.com/r/977H525>). The questionnaire for writers asked them

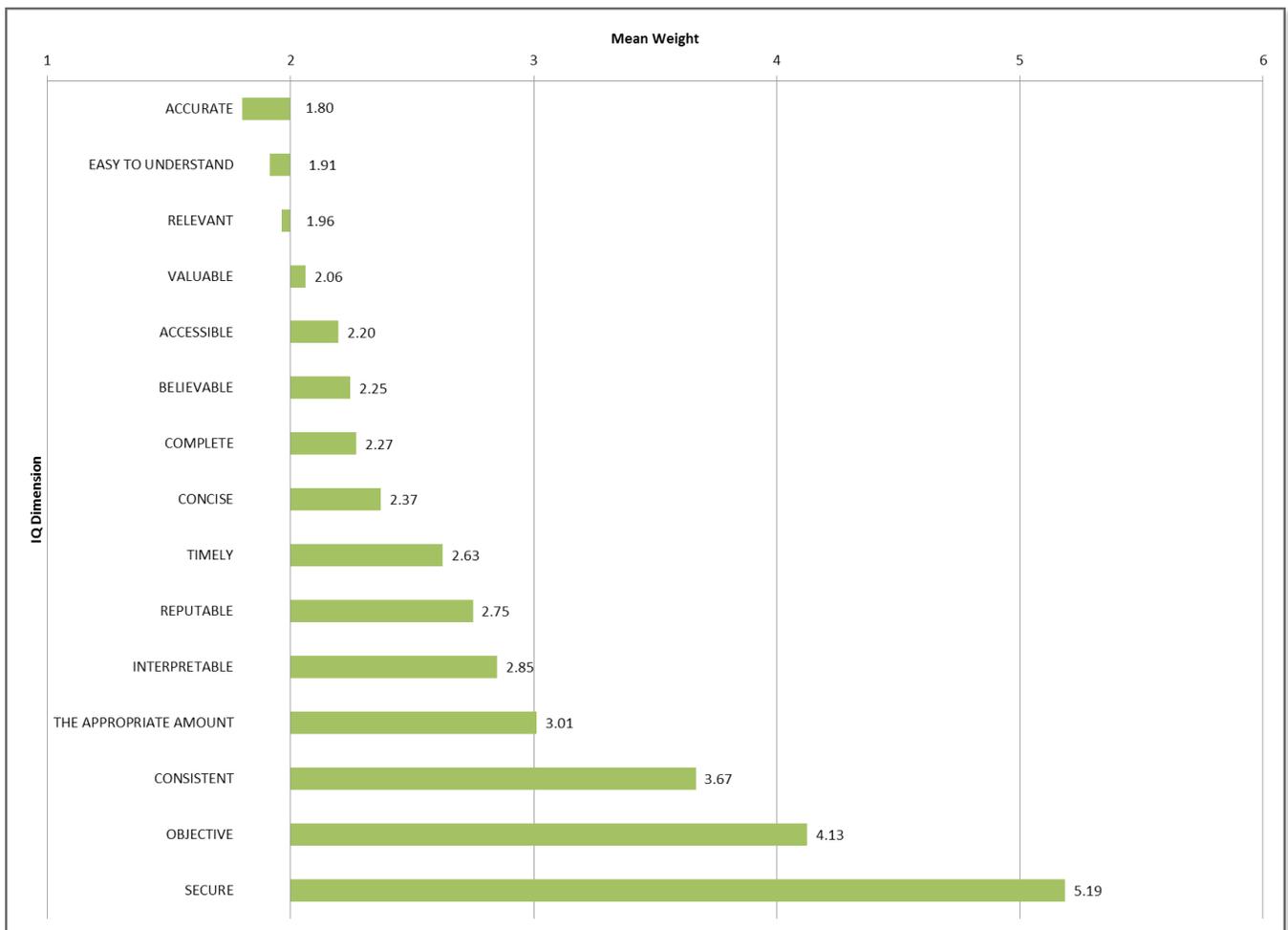


Figure 1. RR Mean Weight Ranking Results

to rank the 15 IQ dimensions from a reader’s assumed point of view (called the Writers’ Assumptions of Readers’ Rankings (WARR) group). The questionnaire for readers asked them to rank the 15 IQ dimensions by their perceived importance as they relate to documentation from their own point of view (the Readers’ Rankings (RR) group).

The link to the writer questionnaire was posted on numerous online technical communication forums and social media pages, and the link to the reader questionnaire was sent to several technical communicators (contacted via the STC website and SIG groups, as well as numerous technical writing groups on Facebook) and to customer service personnel from various companies to send to their readers. This was done to ensure that there was a broad, worldwide range of writers and readers from different fields answering the questionnaires, and that the people answering the questions were the people who actually created, read, and used the documentation.

Ranking and Data Analysis

As in the Wang & Strong study, ranking was done on a nine-point Likert scale (which is also recommended by Wiley, 2006), with 1 being “extremely important” and 9 being “not important at all”. The mean weights of the responses were calculated, and the lower the weight, the more important the dimension. A cutoff of < 2.00 was used to decide which dimensions were to be considered the most important.

The mean weights per IQ dimension between the WARR and RR groups were compared via a one-way ANOVA (run at <http://statpages.org/anova1sm.html>). This was done to determine if the differences in mean weights between the selected groups were significant or not.

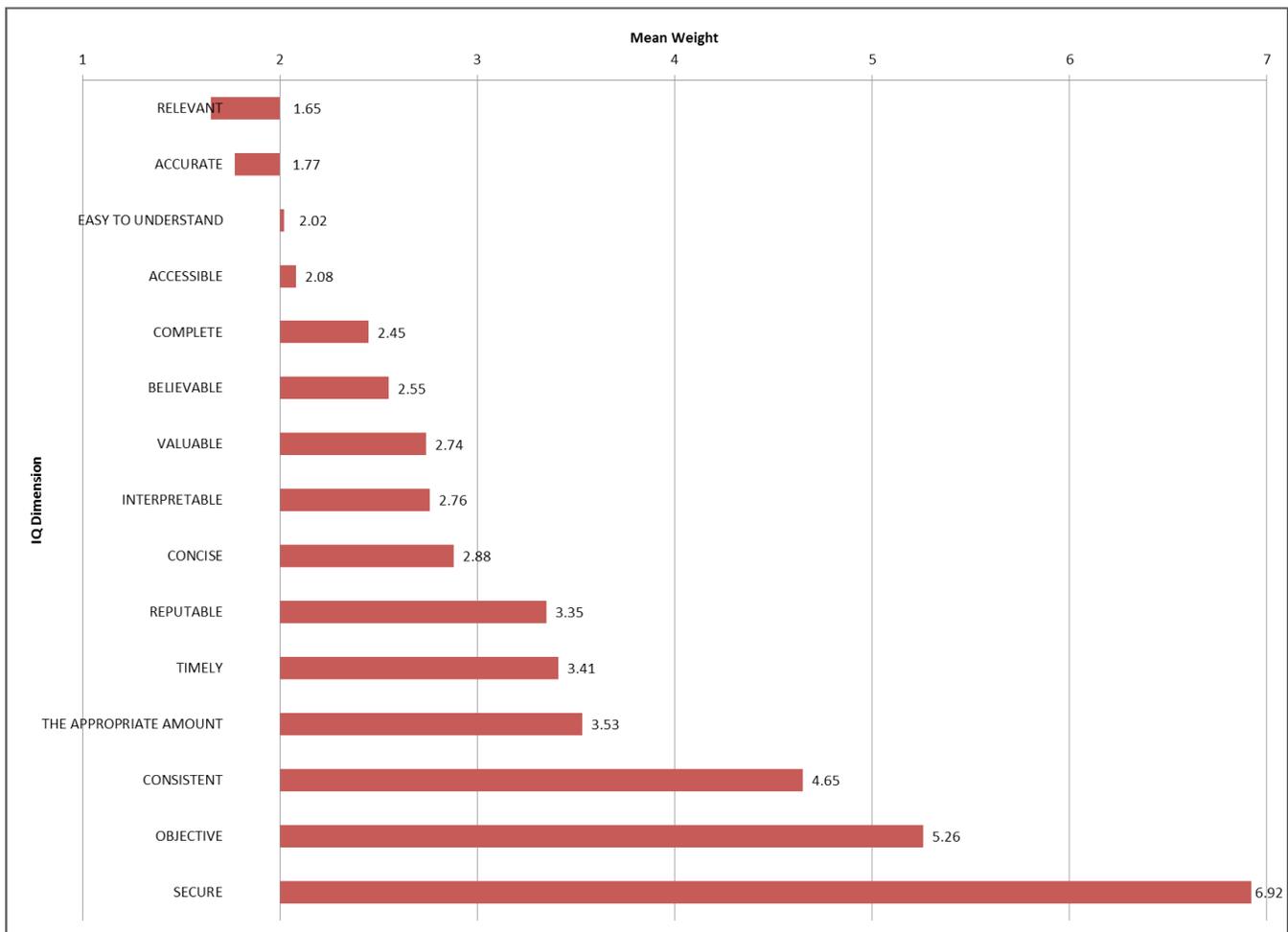


Figure 2. WARR Mean Weight Ranking Results

Results

RR Mean Weight Ranking Results

A total of 81 readers responded to the questionnaire, but only 80 of them ranked all of the IQ dimensions. Using a mean weight cutoff of < 2.00 , the following IQ dimensions were determined to be the most important for readers:

- Accurate (1.80), from the Intrinsic IQ category
- Easy to Understand (1.91), from the Representational IQ category
- Relevant (1.96), from the Contextual IQ category

The full range of IQ dimension mean weights for the RR group is shown in Figure 1. RR Mean Weight Ranking Results.

WARR Mean Weight Ranking Results

A total of 66 writers responded to the questionnaire, and all of them ranked all of the IQ dimensions. Using a mean weight cutoff of < 2.00 , the following IQ dimensions were assumed by writers to be the most important for readers:

- Relevant (1.65), from the Contextual IQ category
- Accurate (1.77), from the Intrinsic IQ category

The full range of IQ dimension mean weights for the WARR group is shown in Figure 2. WARR Mean Weight Ranking Results.

Comparisons between Groups (WARR/RR)

Comparing the differences between the mean weights for each IQ dimension between groups enables us to determine if the differences between them are statistically significant. If the mean weights of an IQ dimension are significantly different (in this study, $p < 0.05$) between two groups, then we can state with some certainty that the two groups consider the importance of that particular IQ dimension differently.

Figure 3 shows a comparison of the mean weights between the RR and WARR groups.

Analyzing the comparison statistically, it was found that there were some significant differences between how writers assumed readers ranked the IQ dimensions and how readers actually ranked them:

- Writers thought that the Secure IQ dimension was significantly less important ($F = 19.9577$, $p < 0.0000$) to readers than it really was.
- Writers thought that the Objective IQ dimension was significantly less important ($F = 9.5802$, $p = 0.0024$) to readers than it really was.
- Writers thought that the Consistent IQ dimension was significantly less important ($F = 6.8994$, $p = 0.0095$) to readers than it really was.
- Writers thought that the Valuable IQ dimension was significantly less important ($F = 6.2277$, $p = 0.0137$) to readers than it really was.
- Writers thought that the Timely IQ dimension was significantly less important ($F = 4.9567$, $p = 0.0275$) to readers than it really was.

Analysis

Readers' Definitions of DQ

The results of the readers' rankings show that, above all, readers expect the documentation they get to be accurate, easy to understand, and relevant. Each of these IQ dimensions represents one of the IQ categories (Intrinsic, Representational, and Contextual, respectively). While this result might seem self-evident, it provides a strong empirical underpinning for the claim that DQ can be defined using a small yet comprehensive set of clear and unambiguous IQ dimensions.

Writers' Assumptions about Readers' Definitions of DQ

The results of the writers' assumptions of readers' rankings show that writers think that readers define DQ using only the Relevant and Accurate IQ dimensions (from the Contextual and Intrinsic categories respectively). The Easy to Understand dimension (from the Representational category) barely misses the < 2.00 cutoff, but it is clear that it would have been counted had the sample size been larger.

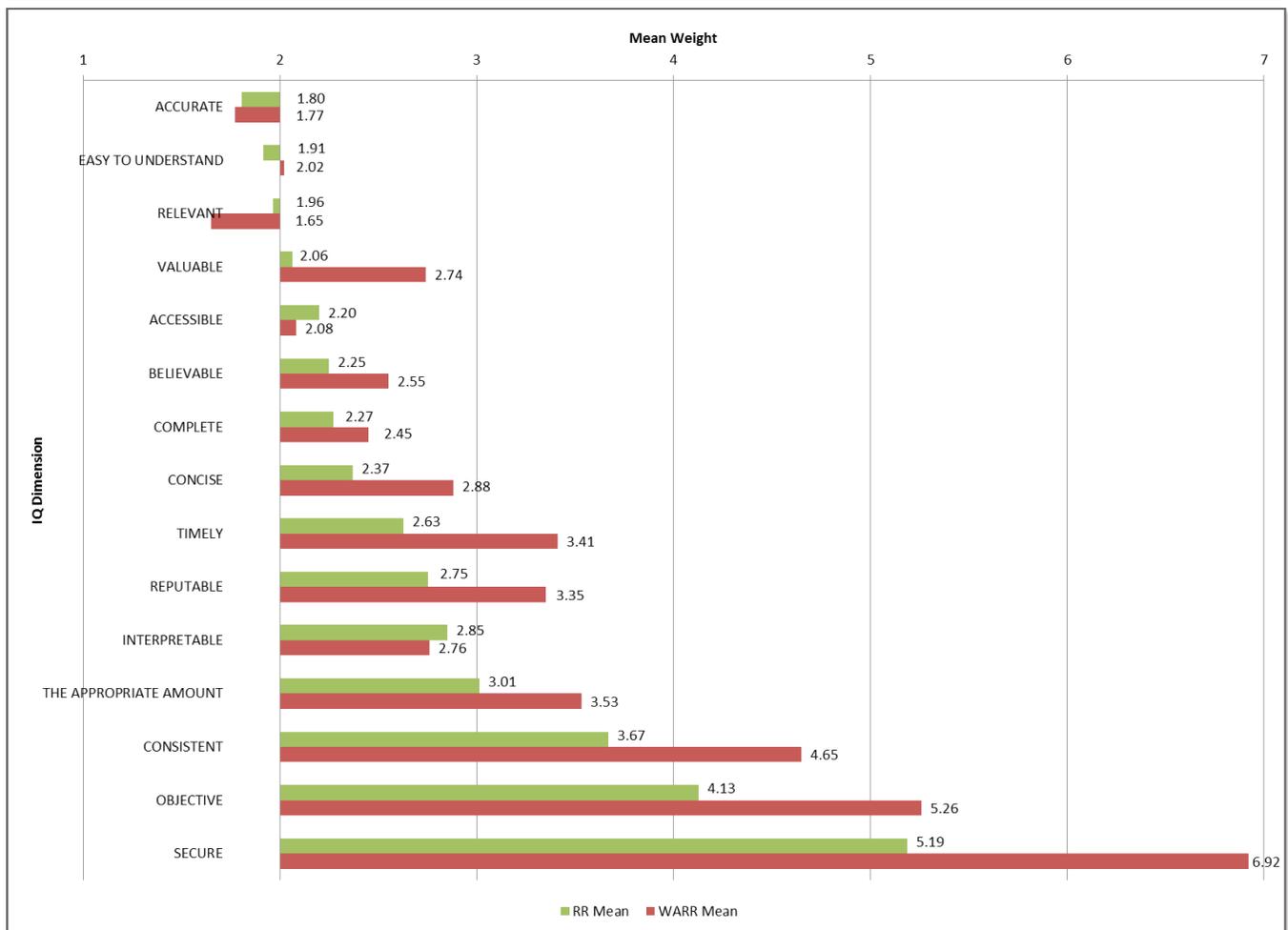


Figure 3. WARR/RR Mean Weight Comparison

This order is interesting, and might reflect writers' beliefs that readers do not consider the grammar, style, and clarity of the documentation to be that important. In truth, though, readers do understand the importance of these, and rank the Easy to Understand dimension second (after Accuracy). However, the differences between the groups for these three dimensions were not statistically significant.

Where Writers Get It Wrong

Of the five IQ dimensions that had significant differences between their perceived importance by writers and their actual importance to readers (Secure, Objective, Consistent, Valuable, and Timely), only the Valuable IQ dimension was ranked highly by readers (2.06). Indeed, this IQ dimension might have even made the < 2.00 cutoff had the sample size been larger.

It is important for us to look more carefully at this result. What are readers telling us when they say that

they want the documentation to be "valuable"? Why are we significantly underestimating the importance of this to readers? And what can we do to address it?

Making Our Documentation Valuable

The wording of the Valuable IQ dimension's definition in the questionnaire gives us a clue: "The information in the documentation is beneficial and provides advantages from its use." Compare this to the definition of the Relevant IQ dimension (note that both of these IQ dimensions are in the Contextual quality category): "The information in the documentation is applicable and helpful for the task at hand."

Readers want the documentation we send them to be more than simply "applicable" and "helpful" – they want it to be "beneficial" and "provide advantages". They want to look at the information and understand

that if they use it, it will improve their situation in some way.

Readers look at the documentation and say “What’s in it for me? Why should I care about this? What value will this information add to my work? How will this make my life easier?” They feel that there must be an additional, emotional level to the documentation. Readers are busy people, and are often under a great deal of pressure to get their tasks done – reading documentation is not usually high on their list of priorities. If they feel that they are not wasting their time with the documentation, and that the writer truly wants to make life easier for them, then they will consider the document to be high quality. Accuracy, clarity, and relevance are critical – but for readers, there also needs to be an extra dimension of value.

It is no surprise that the Contextual quality category is represented twice (Valuable and Relevant) in the readers’ rankings of DQ. Documentation is never read in a vacuum, and is only used in context. Writers, who are not the intended audience of the documentation, can easily lose sight of this and create content devoid of all connection to the context in which the documentation is to be used. But readers cannot use this kind of content.

How do we as writers add value to our documentation? We must understand who our audience is, what they want from the documentation, and in what context they will be using the information we give them. This can be done via user stories, use cases, personas, journey maps, and similar tools that put the writer in the reader’s place. We must also ask ourselves “if I were the reader, would this information help me do my job better?” We must understand that readers look to documentation not only for conceptual and procedural information, but also for ways to make their workload easier.

Conclusions

It is clear that readers define high-quality documentation using a small yet comprehensive set of clear and unambiguous IQ dimensions - Accurate, Easy to Understand, and Relevant. Each of these three IQ dimensions (together with the Accessible IQ dimension) represents one of the Intrinsic, Representational, Contextual, and Accessibility IQ categories, and these must be our working definition of DQ.

We can use these dimensions to classify and sort existing internal or external feedback, which can then be presented to management as clear metrics about the documentation and to determine where more emphasis might need to be invested. We can also use them as unambiguous terminology for discussing and analyzing documentation needs with SMEs and other writers. This will ensure that everyone involved understands what readers want and how to get there – which should be the goal of all people involved in creating documentation.

The bad news, however, is that we significantly underestimate the importance of documentation value to our readers. But this can be solved by considering strategies that put us more directly “in the readers’ shoes” – by thinking about how readers use documentation to make their lives easier, by realizing that there is an underlying emotional component to using documentation, and by carefully considering the context in which the documentation is used. This “feel-good, make it worth my while” factor in documentation cannot be ignored. It is not enough for us to give our readers accurate, clear, and relevant information – we must also ensure that the information we give them enables them to feel that it was worth it for them to read what we have written.

References

- Albers, M. “The key for effective documentation: answer the user’s real question.” *Usability Interface* (May 2005): 5-8.
- Barnum, C. & S. Carliner. Introduction. In Barnum, C. and S. Carliner, eds. *Techniques for Technical Communicators* (pp. 1-11), Needham Heights, MA: Allyn and Bacon, 1993.
- Betz, M. *Delivering Customer Satisfaction: Our Experiences with Responding to Customer Feedback*. (Proceedings, STC Summit Conference, 1996).
- Brown, D. (1995). “Test the usability of research”. *Technical Communication* 42.1 (1995): 12-14.
- Bursaw, C., G. Alred, & W. Oliu. *Handbook of Technical Writing* (4th ed.). New York, NY: St. Martin’s Press, 1993.
- Bush, D. “Editing is magic”. *Intercom* (June 2001): 39 & 43.
- Chisholm, M. “Data quality is not fitness for use.” *Information Management* (12 August 2012). <http://>

- www.information-management.com/news/data-quality-is-not-fitness-for-use-10023022-1.html
- Ericson, J. "Living with imperfect data." *Information Management* (29 June 2012). <http://www.information-management.com/blogs/mdm-data-governance-reporting-innaccuracy-10022790-1.html>
- Gregory, J. (2004). "Writing for the Web versus writing for print: are they really so different?" *Technical Communication* 51.2 (2004): 276-285.
- Hargis, G., M. Carey, A. K. Hernandez, P. Hughes, D. Longo, S. Rouiller, & E. Wilde. *Developing Quality Technical Information: A Handbook for Writers and Editors* (2nd ed.). Upper Saddle River, NJ: Prentice Hall, 2004.
- ISO/IEC 26514. *Systems and software engineering – Requirements for designers and developers of user documentation*. Geneva, Switzerland: International Organization for Standardization, 2008.
- Kumar, M. "Difference between data and information." *DifferenceBetween* (12 October 2009). <http://www.differencebetween.net/language/difference-between-data-and-information/>
- Lee, Y., D. Strong, B. Kahn, & R. Wang, R. "AIMQ: A methodology for information quality assessment." *Information & Management* 40 (2002): 133-146.
- Manning, S. *Using Content Management to Improve Content Quality*. (Proceedings, STC Summit Conference, 2008).
- Mead, J. "Measuring the value added by technical documentation: a review of research and practice." *Technical Communication* 45.3 (1998): 353-379
- Pirsig, R. *Zen and the Art of Motorcycle Maintenance*. New York, NY: William Morrow & Co, 1974.
- Quesenbery, W. "On beyond help: meeting user needs for useful online information." *Technical Communication* 48.2 (2001): 182-188.
- Redish, G. *Understanding Readers*. In Barnum, C. and S. Carliner, eds., *Techniques for Technical Communicators* (pp. 14-41), Needham Heights, MA: Allyn and Bacon, 1993.
- Reeves, C., & D. Bednar. "Defining quality: alternatives and implications." *The Academy of Management Review* 19.3 (1994): 419-445.
- Robinson, P. & R. Etter. *Writing and designing manuals* (third edition). Boca Raton, FL: CRC Press, 2000
- Spyridakis, J. "Guidelines for authoring comprehensible Web pages and evaluating their success." *Technical Communication* 47.3 (2000): 359-382
- Strong, D. Personal communication (2015).
- Tarutz, J. *Technical Editing: The Practical Guide for Editors and Writers*. Reading, MA: Addison-Wesley, 1992.
- Wang, R. & D. Strong, D. "Beyond accuracy: what data quality means to data consumers." *Journal of Management Information Systems* 12.4 (1996): 5-34.
- Wiley, A. "Customer satisfaction measurement." *Intercom* (July/August 2006): 53-54.

Author Contact Information

Yoel Strimling, Senior Member

Senior Technical Editor/Documentation Quality SME
CEVA Inc.

2 Maskit Street

P.O. Box 4047

Herzeliya 4612001

Israel

+972.9.961.3721

yoel.strimling@ceva-dsp.com

Author Biography

Yoel Strimling has been an editor for almost 20 years, and currently works as the Senior Technical Editor/Documentation Quality SME for CEVA Inc. in Herzeliya Pituach, Israel. Over the course of his career, he has successfully improved the content, writing style, and look and feel of his employers' most important and most used customer-facing documentation by researching and applying the principles of documentation quality and survey design.



Tincher, Louise

The Wonderful World of Proposals

This session will show attendees how to leverage their existing technical writing skills in proposal writer, editor and manager roles and decide if they wish to pursue careers in this environment. If you've seriously considered proposal writing, but not yet taken the plunge, this is for you. It will provide an overview of proposal types, roles, tools, tactics, and strategies. We'll look at how proposal writing is both similar to and different from other kinds of technical writing. How can you sell your technical writing skills effectively to employers in this area? What personality types and situations should you expect to encounter? And, last but not least, can you be happy in this environment? I'll draw on my experience in sales, and technical writing and editing, as well as proposal writing, editing and management for specific examples of the good, the bad, and the ugly in this potentially rewarding arena.

Proposal Writers, Editors and Managers are Always in Demand

Businesses categorize functions as “line” and “staff”. Line positions – e.g., production and sales – contribute directly to the organization’s bottom line. Staff positions – e.g., middle managers, corporate administrators – provide specialized advice and support to the organization. Technical writing positions tend to fall into the second (staff) category. In case of a crunch, staff positions are the first to be cut. However, proposals are a key part of the sales process, a line function. Hence proposal writers, editors and managers remain in demand even in a down economy. I believe that technical writers can provide great value by bringing organization and order to the proposal process.

- Clarity – Customers want to know what they’re buying.
- Interviewing – What does the sales team (your internal customer) want?
- Proofreading – To present a professional looking proposal.
- User Focus – What does the end (external) customer want?

Depending on your role, you may also need instructional, project management, administrative, and archival skills. I train my authors on Microsoft Word functions and templates, develop schedules and compliance matrices to keep proposals on track, maintain logs and archives key resource materials, monitor and distribute solicitation announcements. Last, but not least, you will need patience, tact, discretion and flexibility to mesh with your team.

Key Skills for Proposal Writing

All of your basic technical writing skills apply here.

- Accuracy – Don’t promise more than the company plans (or budgets) to deliver.
- Brevity – There are frequently page limits.

Technical Writers in Proposal Teams

Proposal writing is a team sport, often executed by a group of rugged individualists. Technical writers, as professional communicators, are uniquely qualified to serve as the glue that holds the team together. Team

members / stakeholders are experts in their own areas, but may be oblivious to others' concerns. Organize and manage the process so that your stakeholders can concentrate on their areas of expertise.

Over communicate with the team about schedules, requirements, expectations. In kick-off meetings, training and editing reviews differentiate between corrections and stylistic choices. Stakeholders don't generally care to see corrections of typographic, spelling, punctuation, or formatting errors. I provide a brief summary of these and the rationale behind them (e.g. number ordered lists, use bullets for unordered lists). On the other hand, stylistic changes require more coordination. Review the writing style of your company and the end customer via previous proposals, solicitations, websites and other available documents. Be open to stakeholder input on specialized technical, industry, market, and even customer-specific jargon and usage. A proposal is a sales document, and in sales the customer is always right!

Members of a proposal team are privy to information that may be confidential (salaries) and proprietary

(intellectual property), as well as strategic (Win themes). Remember to use care in disclosing information both inside and outside of your organization. When in doubt check with management before sharing information with anyone but the source and the proposal leads. This includes handling proposal files in a secure manner. Your discretion will be appreciated.

Environment: Pros and Cons

Proposal writing is an essential business function that is always in demand and can transfer easily across industries. It can be high stakes, with the entire organization having an interest in the outcome. Proposal writing can be high stress. It is often deadline driven and teams usually include at least one procrastinator. Proposal writing is a high visibility job in direct contact with a wide variety of people inside and outside of the organization. You can make powerful allies and build a strong network. You can also receive unfair blame and unreasonable demands.

Role	Focused On
Executives	Big picture; overall company cost, profit, impact
Managers	Division-level cost, profit, impact
Corporate Staff <ul style="list-style-type: none"> - HR - Compliance - Contracts - Legal 	<ul style="list-style-type: none"> - Compliance with company policies and procedures - Compliance with applicable laws and regulations - Teaming agreements, non-disclosure agreements - Protection of proprietary information
Sales \ Marketing \ Business Development	<ul style="list-style-type: none"> - Making sales to customers - Expanding existing markets and developing new markets - Bringing new products to market
Technical Professionals <ul style="list-style-type: none"> - Developers - Engineers - Scientists 	<ul style="list-style-type: none"> - Proposing executable solutions - Providing realistic costs and schedules - Getting the job done right
Technical Writers	Internal and external communications

Table 1. Proposal Team Members / Stakeholders.

Can You be Happy Here?

So the real question is whether you can be happy in this environment. Can you accept imperfection? Sometimes good enough is good enough. Can you be flexible? Customers and stakeholders frequently make last minute changes. Are you patient? You'll need to work with different personalities under stress. Are you tactful? Can you be assertive without becoming confrontational? Are you healthy? It really helps in a crunch. If the answer to these questions is yes, then consider proposal writing as a career option.

Author Contact Information

Louise H. Tincher
Proposal Manager
UES, Inc.
4401 Dayton-Xenia Road
Dayton, OH 45432
937.426.6900

Author Biography

Louise H. Tincher is Proposal Manager for UES, Inc., a small business specializing in materials science and research for government and industry. She is a senior member of STC and served as president of the Southwestern Ohio Chapter. She currently volunteers with the STC Policies and Procedures SIG and the Greater Ohio Valley Chapter of Women in Defense.

Louise has spent 20 years in technical writing, roughly half that in the area of proposals. Her proposal-related work included stints as 1) administrator/analyst/bid writer for a company selling commercial robot systems, 2) editor of requests for proposals for an Air Force Program Office, 3) bid writer for an agency collecting delinquent retail, student loan, and tax debt, 4) proposal manager for a small research laboratory pursuing government grants and contracts. Prior to becoming a technical writer, she spent eight years (1986-1994) selling computer systems to small businesses.



Todd, Becky

Engineering Content Champions

In the world of crowdsourcing, content champions play a key role in curating technical documentation even though this is not their primary responsibility. This paper focuses on setting your content champions up for success.

Organizations that use crowdsourcing models rely on a collaborative approach for developing and managing technical documentation. In these environments, it is not uncommon for content champions to emerge. These content champions have natural instincts for content management, even though managing content is not their main role.

In order to better understand how to set our content champions up for success, let's first explore the writing environments in which they appear and, hopefully, flourish.

Crowdsourcing Documentation

In the software development community, there is a distinct difference between crowdsourcing and open sourcing. Simon Phipps, former President of the Open Source Initiative, distinguishes the two approaches based on ownership of the outcome. He defines crowdsourcing as "leveraging of the marginal interest and free time of a large group of people to complete a task" while open source development is tied to a community "of equal peers, with no one participant necessarily benefiting from more than any other."

If we apply the above definitions to a crowdsourcing technical documentation, then we have an approach that relies on occasional authors writing content on a volunteer basis. These volunteer authors are often subject matter experts (SMEs) who have deep product knowledge and an understanding for the user's pain points.

Note: In addition to the concept of ownership, software licenses and legal requirements may be significantly different between open source and crowdsourced projects. For organizations that do not use open source licenses, this can be a key way of differentiating between or defining acceptable models of content contribution. For the purposes of this paper, we will not go into detail about licensing or legal requirements. Instead, we will use the term crowdsourcing to broadly describe the approach of using many authors to develop content.

Benefits of Crowdsourcing

Content written by SMEs can be of high value to end users for a few reasons:

- **Solves specific issues.** Frequently, content developed by volunteer authors takes the form of documenting specific use cases that may otherwise go undocumented. In many cases, these take the form of detailed how-to articles that are in context of how a user actually uses the product or service.
- **Unlocks knowledge from different roles.** In addition to detailed how-to articles, volunteer authors from different roles can contribute content for audiences close to their specialization. For example, a developer with a deep understanding of a technical topic could provide a series of technical examples and tutorials written for a developer audience.
- **Minimizes bottlenecks for publishing content.** Crowdsourcing opens the doors to having a

seemingly vast pool of authors and reviewers, which can be a huge advantage for reducing review and publishing bottlenecks.

- Challenges of Crowdsourcing
- Crowdsourcing is not without its challenges, however. Introducing volunteer authors into the mix can lead to the following types of issues:
- **Distributed editorial responsibility.** The editorial oversight for planning, producing, reviewing, and publishing content may be tied to roles outside of a technical writing team. When this happens, it may become unclear who is responsible for various aspects of the publishing process. Further, it may become unclear where to report content problems or who should resolve them.
- **Content quality drops.** An unfortunate side effect of crowdsourcing is the tendency for documentation quality to erode over time, especially if you started with a set of content that was meticulously maintained by a dedicated team of content professionals. Writing is a skill as much as any other, and not everyone excels at it.
- **Quantity does not equal quality.** Too much content can actually be detrimental to the user experience. For example, what happens when you have three different getting started guides, each with a different point of view? These types of issues can cause a lot of confusion for new users. You may also find yourself in a situation where there is too much content to reliably maintain.

It is not unreasonable to argue that the second and third points above are both symptoms of the first. For this reason, it is important to still routinely revisit your content and ensure that you have a solid content strategy. It is also important to ensure that you have a strong editorial role within the organization to manage content processes.

Content Champion, Defined

Now that we know more about the benefits and challenges of crowdsourcing, let's talk about who the content champions are. Earlier, we stated that content champions play a key role in curating technical documentation even though writing documentation is not their primary responsibility.

It is important to understand that a volunteer author can come from any team. If we exclude technical writers for a moment, this means that you may have people from support, engineering, or even professional services acting as content creators. People in these roles have a practical working knowledge of the product or service, but not all of these volunteer authors will become content champions.

What sets a content champion apart is:

- A deep understanding of user pain points and use cases combined with a tendency to ensure that the documentation supports them, and
- A sense of ownership or responsibility for a particular area of documentation.

In addition, content champions may work more closely with content managers outside of their team for planning purposes.

Engineering for success

Now that we've identified the qualities that define a content champion and understand some of the benefits and challenges of crowdsourcing, let's review a few practical ways to ensure that you set your champions and other volunteer authors up for success.

Keep the Process Simple

A simple, straightforward process is a crucial component for success with crowdsourcing documentation. First, a simple process helps to reduce barriers that can get in the way of an author wanting to submit content. In an ideal scenario, this means that they can get started easily and follow an intuitive workflow all the way through to publishing.

Keeping the process simple also reduces the burden on the technical writing team so that they don't spend a majority of their time troubleshooting environmental problems or being bogged down with technical issues.

You may find that it is useful to define more than one workflow based on the needs of different types of occasional authors.

Let's take a look at a few basic examples:

- **Support engineer:** Desires a one-click process for writing content in a knowledge base from the context of a support issue.
- **Marketing copywriter:** Desires the ability to publish content on a particular date and time to align with a product release announcement.
- **Product manager:** Wants a low-friction way to release documentation along with a new feature.
- **Technical writer:** Prefers working as a power user and wants to understand the publishing system inside and out so that they can optimize their ability to contribute.

Given the above requirements, you may determine that you need at least two workflows:

- A simplified workflow that supports casual authoring. This may take the form of an online, what you see is what you get (WYSIWYG) editor with a single authoring and review step.
- A full authoring workflow with the ability to return content for editing after a review is completed. This gives the author full control over publishing.

Build a Writing Toolkit

A writing toolkit is an essential set of resources for occasional authors to reference when writing documentation.

It consists of the following resources:

- **Style guide.** A style guide is a document that describes writing and brand standards in detail. Depending on the complexity of the guide, this can include everything from formatting guidance to the use of colors and typefaces. Note that occasional authors may not always associate the term *style guide* with a set of writing and formatting standards. For this reason, it may make sense to refer to this as a formatting guide when introducing it.
- **Guidelines.** Writing guidelines are a set of documents that focus on how to produce documentation and may include everything from writing checklists to templates and best practices. These guidelines are targeted at the occasional author audience and frequently take the form of tutorials. For example, occasional authors

frequently seek advice on what and when to write.

- **Instructions.** A simple set of instructions for writing and publishing content make a handy reference for both new and experienced authors. The instructions should clearly describe the following things so that anyone who needs to can publish content in an autonomous manner:
 - How to access the authoring tools and set up the authoring environment.
 - How to use the version control system (if applicable).
 - How to create, edit, and delete documents.
 - How to submit document updates for review (if required).
 - How to publish a document update.

Once the writing toolkit is in place, take time to socialize it and conduct trainings. Otherwise, volunteer authors may never realize the resources are available to them.

Cultivate a Culture of Documentation

Finally, it is important to cultivate a documentation-first mindset at all levels of an organization. This can be challenging to do, but is critical to the success of crowdsourcing. This is where having content champions spread throughout your organization becomes valuable. Like technical writers, they are there to defend the need to document things and will strive to ensure that content is given proper consideration.

Additionally, if everyone at the organization takes pride in having good documentation, you're more likely to see documentation quality rise over time. For this reason, it is important to recognize and celebrate the successes of content champions and volunteer authors.

Conclusion

Crowdsourcing documentation is an increasingly popular approach to creating technical documentation, especially when you have few authors or simply seek to unlock the full potential of knowledge at your organization. Without a culture of documentation led by content champions, crowdsourcing models are likely to flounder.

Resources

Todd, Becky. "Design thinking: Building a developer experience from scratch." (January 2017) https://www.slideshare.net/slideshow/embed_code/key/IVToPyZ9IQphTQ

Vigilante, Nick. Personal interview. 13 April 2017

References

"Documentation Guide." Write the Docs (2017) <http://www.writethedocs.org/guide/>

Editing definition, Wikipedia. <https://en.wikipedia.org/wiki/Editing>

Johnson, Tom. "Crowdsourcing docs with docs-as-code tools -- same result as with wikis?" I'd Rather Be Writing (8 March 2017). <http://idratherbewriting.com/2017/03/08/crowdsourcing-docs-with-github-docs-as-code-tools-same-as-wikis/>.

Phipps, Simon. "Crowdsource Is Not Open Source." Computerworld UK. (30 November 2010). <http://www.computerworlduk.com/it-business/crowdsource-is-not-open-source-3569582/>

Procida, Daniele. Documentation-driven development – lessons learned from the Django Project. (PyCon 2016) <https://www.youtube.com/watch?v=bQSR1UpUdFQ&feature=youtu.be>

Software license definitions, Wikipedia. https://en.wikipedia.org/wiki/Software_license

Author Contact Information

Becky Todd

Senior Developer Experience Writer

Atlassian

303 Colorado St. Ste. 1600

Austin, TX 78704

<https://www.linkedin.com/in/toddbecky/>

Author Biography

Becky Todd is a Senior Developer Experience Writer at Atlassian. She has managed enterprise content at various levels throughout her career and is a self-professed lover of Git. At Atlassian, she acted as a product owner for the development and launch of the updated developer site and is involved with building the developer experience writing team. She is active in the Meetup community, and has spoken at recent Write the Docs Meetups in London, UK and Austin, TX.

She is the founder of the InfoDevDC Meetup group in Washington, D.C., co-organizer of the Write the Docs ATX Meetup in Austin, and served as Second Vice President of the Washington, DC-Baltimore chapter of the STC.



Vega, Laurian C.

Novices *and* Expert Users, Not Novices *or* Expert Users

A user interface is like a joke; if you have to explain it, it isn't very good. Like a joke, everyone should understand a good user interface (UI). But, just because everyone can understand the UI does not mean that all users will approach it with the same domain knowledge and familiarity with modern UI elements. For any system there is going to be a spectrum of user knowledge and a budget for one interface. To create an adaptable user interface that will work for novice and expert users requires design and usability knowledge. In this presentation I will review the spectrum of user skills and knowledge, examine how to design for both, and cover common user interface components that can be adapted to support both novices and experts.

Introduction

Supporting novice users has a great business case: everyone starts as a novice and novices are usually the largest consumer of online applications [1]. The problem is that novices are not the only users of software. At the opposite end of novices are experts. These two sets of users can have vastly different experiences and vastly different tactics for being successful with the same exact software. Yet, with software design budget constraints there is usually only enough money and time to make one user interface (UI).

To design for both experts and novices requires and understanding that they have different workflows and information density needs. Often, the focus is on novices because everyone starts as a novice. However, this can be to the detriment of the expert. In an article by Jakob Nielsen [7] and another by Bruce Tognazzini [8], both experts in the field of User Experience and Interaction Design, they argue that the focus on the learnability of a UI has made the idea of designing for the expert a taboo. This is because experts are going to need different user interface elements that may be visually displeasing to interface designers. Experts can require extensive features, visual clutter, and

a much larger information display density than the novice. Roughly, this means that the user interface that encourages learning is not necessarily the right interface for the expert. UIs that encourage learning have open spaces, clutter reduced, and text to tell the user how to be successful; features that are contrary to experts

Designing for both is not impossible. For example, the Google homepage (www.google.com) shown above does a great job of supporting users along the spectrum of novice to expert. When users first visits the homepage they are visually encouraged by the inviting logo on a stark white background with an open text box underneath the logo. Two buttons are anchored below the text box that give clear calls to action. For the novice user this interface is simple; the help text is even incorporated into the button text, "Google Search." The workflow of entering search terms is clear to the novice user without even a click. What is unexpected is that the expert user is also supported on this same page, and not just because the user interface is easy to understand. The experts are supported by all of the items in light gray text that are anchored to the top and bottom of the page. Examples include quick links to Gmail and Images. To novice users these items are ignored because they blend into the background. To the expert these are powerful tools

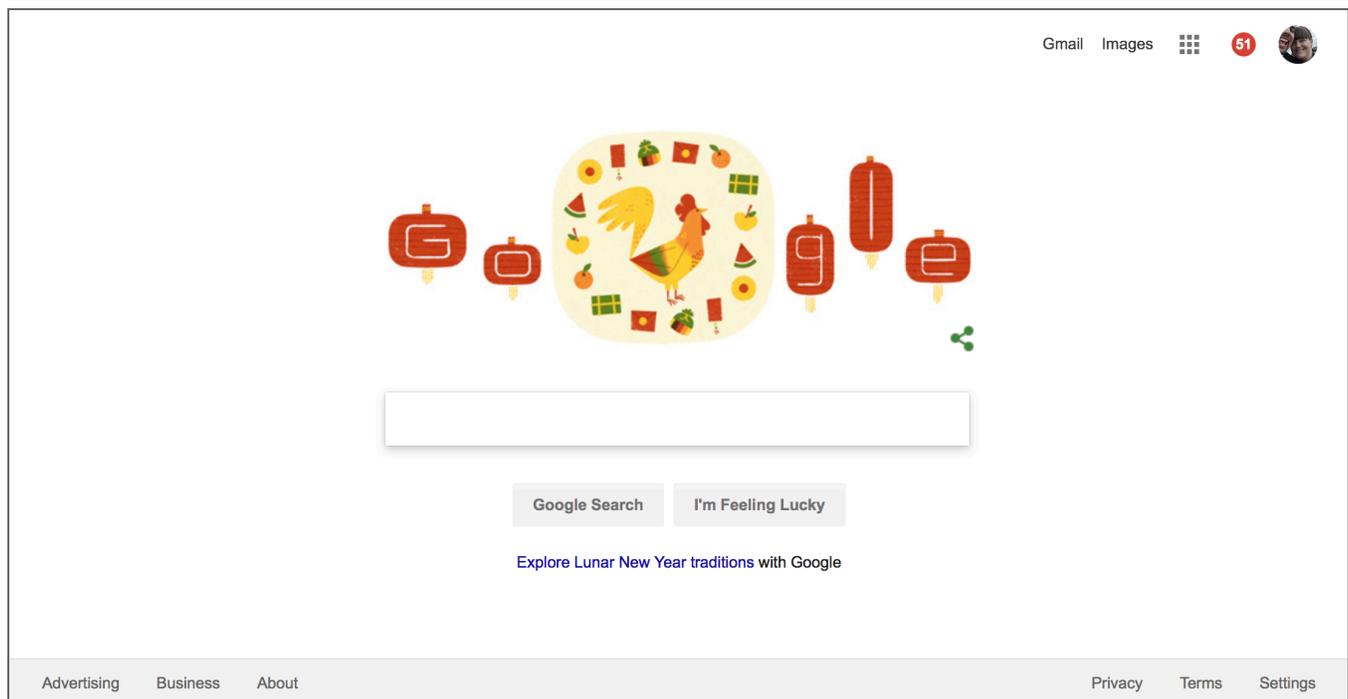


Figure 1. The Google homepage, which is perfect for novices and experts.

that help them work quicker and with less frustration. Both users can find a home in the same interface.

All of the factors discussed above can impact how easily a novice and expert user is able to navigate an application. The rest of this article is dedicated to understanding how to replicate the success of the Google homepage by understanding how user needs can vary, creating tools that reify user needs, and making use of powerful design patterns.

Defining diversity of user needs

User interfaces must be designed to accommodate a spectrum of users. There are multiple factors that impact expertise:

- How much time the user spends with the user interface. Users who spend a short amount of time with the software are going to have varying levels of knowledge, or even repeated frustration, with the same user interface. Prior successes or failures are going to impact the success of future interaction with the software [2].
- How familiar the user is with modern UI elements and technology in general. This has three implications. The first is that UI elements can be styled differently, and this styling can impact how well users perform with the same UI element.
- As an example, one study found that younger users preferred and were more familiar, and thus performed better, with skeuomorphism (or flat design) [3]. The second implication is that as new UI libraries are released people who were familiar with older libraries are going to have to transition. This means that users are going to have to learn new methods of interaction to accomplish the same goal. Thirdly, users who are more familiar with multiple types of technology are more proficient at transitioning skills between applications [4]. Similar to learning multiple UI element stylings, when people are used to interacting with multiple kinds of technology they are more adept at transitioning those skills.
- The tasks the user completes impacts the features used. Users who complete focused and discrete tasks are going to use a smaller set of features within a user interface when compared to users that interact with a broad set of features to complete a diverse set of tasks. Inherently, users with discrete tasks can more easily master their path through the user interface to complete a task than users with more nebulous task paths [5].
- Users with domain expertise understand jargon and specific task elements more easily. When software is created for a niche set of users, the software is going to reflect that jargon back to the user group. However, there are going to be

novices within the niche user group who have not yet learned the domain knowledge.

- Users have difference preferences. As an example, one study looked at top versus side navigation and found that a preference for one other the other resulted in more usability errors in the one that was disliked [6].

Designing for Diversity of User Skills

When designing a UI think about your workflows and not your UI features. To create a user interface that will work for the entire spectrum of users starts with understanding the common workflows. The list of workflows can then accommodate workflows that are targeted towards novices and workflow that are targeted towards experts. This is because expert and novices users are going to use different features and accomplish different task than “average” users. They are going to approach the UI with different mental models of how to be successful while using the application. The way to accommodate these different approaches to tackling the same goal is with user personas.

User personas are fictional descriptions of prototypical users that are being targeted by the application team to ensure that key user needs are being supported. Personas usually include a name, age, gender, expertise, domain knowledge, and workflows. Personas have a way of making the spectrum of user types and background knowledge come to life and allow the development team to properly personalize tasks so that real users are more successful.

To demonstrate the power of personas is best demonstrated with an example. Zappos, an online shoe store, could have a persona for a first time user and single mom of three with a high domain knowledge of shoes. They could also have another persona of a young

male return shopper with little knowledge of shoes. These two dichotomous personas allow the team to then step those two different users through different tasks. How would the mom search for shoes for weightlifting versus how would the young man search for dress shoes for a job interview?

By stepping through those workflows the team can assess which features have overlapping and conflicting user types versus which features are unique to a skill level. For example, the weight-lifting mom could use advanced filtering and search features and then want to compare two pairs of shoes in a side by side feature listings. In contrast, the young man could casually browse over a series of days and add shoes he is considering to his shopping cart while making his purchase decision. Both users would go through the same check out and quick search tools. However, advanced features such as filtering and side by side comparison might only be used by expert users. Whereas adding to and persisting items in a shopping cart may be unique to novice users.

Once the software development team understand the user types, start user testing your user interface by finding users that match your personas. Finding users that match the created personas is as easy as a short survey that can be included in your participation solicitation. Having between three to five users of each

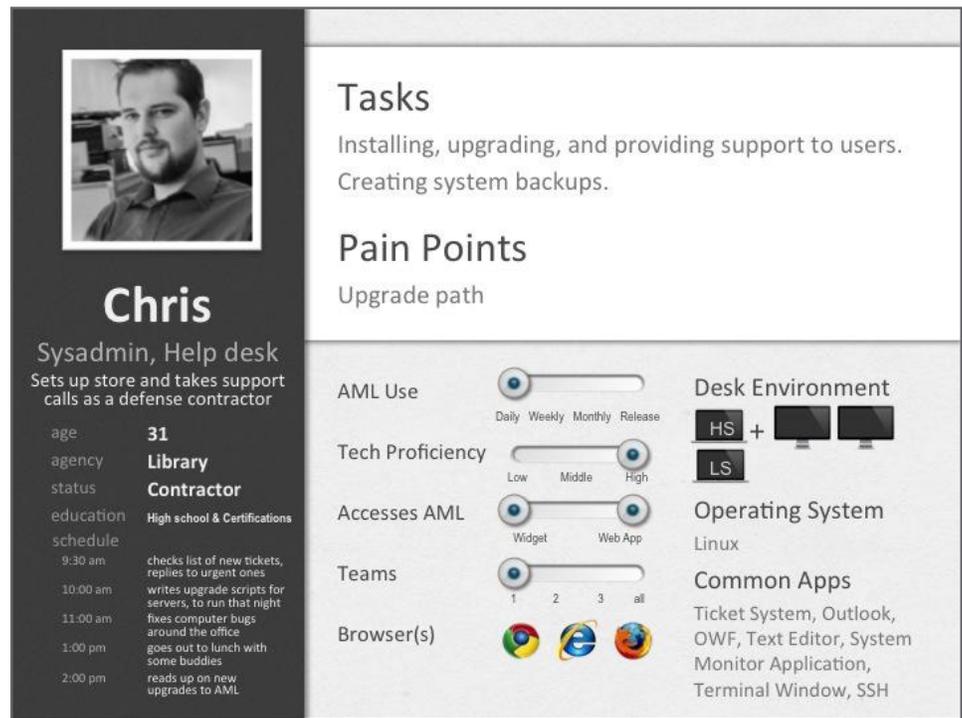


Figure 2. A sample persona that discusses domain expertise, application use, and technical proficiency.

user type should demonstrate a valuable set of issues to discuss for improving the software. Rinse and repeat this method until no further design and usability problems are discovered.

Common Design Patterns

There are three types of design patterns that are useful to review when considering UI elements for novices and experts: the patterns you should not use, the ones that you should use, and the ones that you should use with caution. These patterns are listed and described below.

Non-Useful Tools for Supporting Experts and Users

The following list of design patterns should not be used when designing an interface for a diverse skill spectrum:

- Do not make two separate interfaces. Not only is this costly, but it is unnecessary.
- Do not ask users about their expertise level. Users do not report correctly.
- Do not use different tones. Users can tell when you are speaking down to them. Also, if the language is useful for one type of users it is useful for others.

When conducting user testing or beta testing, do not test with only novices or advanced users. Consider interacting with a spectrum of user experiences. (This includes creating personas for users with a medium expertise level.)

Useful Tools for Supporting Experts and Users

The following list of design patterns are useful to consider when designing an interface for a diverse skill spectrum:

- Provide default options that demonstrate the easiest path to first-time completion.
- Provide a Quick Search, but also provide an Advanced Search. Boolean search should always be within an Advanced Search.

- Provide users to option to not see help text and other first-time user options such as tutorials.
- Provide locations where domain knowledge is explained through information icons and pop-up and on-click hovers.
- Provide advanced options within the context of a user's interaction flow. The context will provide explanation for how to use the action and facilitate the user interacting with more advanced features.
- Make use of progressive disclosure. Progressive disclosure is where more complex options are given through additional selections. This allows for cognitive load to be measured and not overwhelmed. However, when using progressive disclosure, always provide a method to backtrack and to move back to the earliest state.
- Have sections with complicated features explicitly labeled as Advanced — particularly if it is an option with changes that are irrevocable.
- Anchor experiences for advanced users and novice users so that they start and end with the same experience. This allows for users who want to explore to have the same sense of completion as the novice.
- If you are going to provide advanced features that are hidden until users have obtained skills, use a method where experience is measured. For example, one measurement of expertise is the number of times the user has logged in. Which ever method is used, how experience is measured should be explicitly listed within the UI and the interface should provide an easy to discover.

Potentially Dangerous Tools for Supporting Experts and Users

The following list of design patterns are useful to consider when designing an interface for a diverse skill spectrum, but should be used with caution:

- Use custom icons with care. Icons are especially powerful at conserving space and making multiple actions or information easily understood. However, custom icons are costly to maintain and create. They increase error rates for all but experts. If you must use custom icons use help text to describe them.
- Create custom profiles where users can select to see (or not see) help features. This kind

of personalization is enticing for users who insist that they are experts and are annoyed at additional help features like tutorials or help text. But, be wary. Particular in systems that require extreme domain expertise, users will need to use these types features. A user may quickly select to stop seeing help text, but then become lost at a later point with a new feature. If you must use custom profiles for tracking help features, create settings for every help feature that can be easily turned on and off within each task.

Conclusion

When thinking about novices and experts it is useful to keep in mind that all users are novices when new UI elements or features are introduced. Even experts will have to go through the process of evaluating, understanding, and learning how to incorporate new features into old workflows. In Alan Coopers seminal work, *About Face*, he argues that there is no such thing as an expert. Website and software change making even an expert user a “perpetual intermediate” [9].

To design for a spectrum of user expertise define what tasks and characteristics are typical of that kind of user. Consider even making these definitions into personas that can be readily reviewed and used to assess which kinds of user interface elements may be useful to incorporate. Then weigh and choose which UI elements to use for those user types. Last, always user test. With these skills any user interface can support and enable success for all users.

References

- [1] Clarkson, Joshua J., Chris Janiszewski, and Melissa D. Cinelli. “The desire for consumption knowledge.” *Journal of Consumer Research* 39.6 (2013): 1313-1329. <https://doi.org/10.1086/668535>
- [2] Hone, Kate. “Empathic agents to reduce user frustration: The effects of varying agent characteristics.” *Interacting with computers* 18.2 (2006): 227-245.
- [3] Robbins, William Hunt. “Design practices in mobile user interface design.” (2014).
- [4] Prümper, Jochen, et al. “Errors in computerized office work: differences between novice and expert users.” *ACM SIGCHI Bulletin* 23.2 (1991): 63-66.
- [5] Santhanam, Radhika, and Susan Wiedenbeck. “Neither novice nor expert: the discretionary user of software.” *International Journal of Man-Machine Studies* 38.2 (1993): 201-229.
- [6] Pratt, Jean A., Robert J. Mills, and Yongseog Kim. “The effects of navigational orientation and user experience on user task efficiency and frustration levels.” *Journal of Computer Information Systems* 44.4 (2004): 93-100.
- [7] Cipan, Vibor. “User interfaces for beginners, intermediates or experts?” *UX Passion*, June 25 (2010): 2010.
- [8] Nielsen, Jakob. “Novice vs. expert users.” *Alertbox*, February 6 (2000): 2000.
- [9] Cooper, Alan, Robert Reimann, and David Cronin. *About face 3: the essentials of interaction design*. John Wiley & Sons, 2007.

Author Contact Information

Laurian C. Vega, PhD.
 LaurianVega@NextCentury.com
 Senior User Experience Engineer
 Next Century Corporation
 2701 Technology Drive, Suite #100
 Annapolis Junction, MD 20701
 443.545.3100

Author Biography

Dr. Vega is an accomplished user experience engineer who wakes up every day knowing that she can make a software application more useful, usable, and delightful. She has a doctorate in computer science with a concentration in human-computer interaction from Virginia Tech. Dr. Vega is currently the User Experience lead for seventeen products that visualize up to teraflop sized datasets for as many as 200,000 users. She manages a team of 5 user experience engineers and works in a larger product development team of 60 software developers. These programs present real-time visualizations of data being reduced, analyzed, and presented in sub-second rendered interfaces. Additional projects that she has worked on include an online store for a community of 200,000 users and an open-source online widget desktop.



Wilson, Jane

Creating User Documentation in an Agile World

When the Agile Manifesto was introduced in 2001, it did not make any allowance for or reference to the creation of user content. Since then, technical communicators have been adopting agile practices and adapting themselves to agile methodologies, but it sometimes feels like we are fitting a square peg into a round hole. It doesn't have to be that way! This presentation will address ways to smooth off that square peg and adapt traditional documentation processes and procedures to make them fit in an agile world.

What Is Agile?

Agile is a development methodology in which software is created in short iterations called sprints. In February 2001, a group of seventeen software designers wrote the Agile Manifesto, which defined a series of prescriptive principles for iterative software development. The first principle encapsulates the spirit of Agile: "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." Through short iterations and constant feedback from customers and customer representatives, Agile development allows teams to adapt quickly, fail fast, then course correct rapidly when necessary. This promotes the development of products of higher quality that more closely meet the customers' needs than those created using traditional waterfall methodology.

Other hallmarks of Agile include self-organizing teams and scrum ceremonies. Agile teams work closely together and define their own work. Large features are defined as *epics*, then are broken down into smaller components called *user stories*, which are, in turn, often broken down into sub-tasks. Each of these elements has a *definition of done* which must be met to complete the story. Members of the team are responsible for defining epics and stories, then sizing them (using *story points*) and choosing how much work to take on each sprint.

Agile (or Scrum) practices include a series of ceremonies involving the entire team:

- **Planning:** Meetings held at the beginning of each sprint in which a team prioritizes work, grooms the backlog of work, and chooses what stories will be taken up in the coming sprint.
- **Daily Scrum:** Short stand-up meetings in which team members report on progress and bring up any blockers to accomplishing work.
- **Sprint Reviews:** Meetings held at the end of each sprint to demonstrate what has been accomplished in the sprint and get feedback from customers.
- **Retrospectives:** Meetings held at the end of each sprint to review what went well in the previous sprint, what did not go well, and how processes can be improved.

Agile development practices are designed to bring transparency and simplicity to software development.

Agile for the Technical Writer

The question, though, is how can a technical writer efficiently and effectively work to create content within an agile team? Neither the Agile Manifesto nor any standard Agile processes mention the development of documentation. As a result, teams often handle documentation uniquely. Some writers

are considered part of the scrum team, while some work independently of the scrum. Some teams include complete user documentation as part of their definition of done, some do not. Often it is up to the writer or the writing team to forge a place for themselves in the development processes. We have found that sometimes it feels like putting a square peg in a round hole, and it is sometimes necessary to bend the Agile rules a bit to ensure a smooth fit. The following are processes I have used and my team has established to incorporate technical writers effectively into development scrum teams.

The Writer's Role on a Scrum Team

In Agile methodology, only three team member types are defined: Product Owner, Scrum Master, and Team Member. The Product Owner represents the interests of the customer and helps to prioritize work based on customer needs. The Scrum Master runs all scrum ceremonies and is responsible for making sure that work is moving forward by addressing all blockers that occur. The Team Members are the ones who take the stories and complete the work. In theory, team members should be interchangeable in that any team member should be able to complete any user story that is on the board for the team. However, this breaks down when writing work is introduced into the mix.

In our teams, we treat writers as specialized members of the scrum teams.

They usually cover multiple scrum teams, but on each team, they have the same duties:

- Work with the Product Owner and team to create doc stories
- Participate in scrum stand-up meetings
- Participate in planning and demo meetings
- Maintain the content plan.

Because writers cover multiple scrum teams, we often have to remind those teams that a writer is multi-tasking. The biggest challenge is covering all the necessary meetings for multiple teams and still having time to complete the writing work. In addition, the writers need to balance story load and must speak up when they are getting too much work.

Documentation Stories

One of the most controversial issues we have tackled is how to track the work the writers do. The ideal state is when writers are able to create content in tandem with the developers writing the code. This way the two tasks are completed at the same time, and the product is ready to ship. In practical terms, though, documentation often lags behind the code. This happens because development and design change rapidly, especially when a team is working in Agile, where the developer often begins with only minimal design specifications. In addition, when a writer is juggling multiple teams, sometimes one team or feature needs to be prioritized over another.

When we began working in Agile, we tracked documentation work as part of the development story, but we found that we were under a lot of pressure to close a story too quickly. We were often in the position of not getting the information we needed, or access to the completed feature, until the end of a sprint, then having to complete all of the documentation at once. As a result, documentation quality suffered.

We began to create separate documentation stories apart from the development stories. These documentation stories reflected all the work done by technical writers as part of the scrum team. This allowed us to begin our work as the features were being developed, but gave us the luxury of completing them after the development story was closed. We tend to complete documentation within a day or two after development is complete, and no later than the Feature Complete date for a release. The definition of done for an epic still includes completed user documentation, but we can group and complete our work in a way that is logical for us.

In our system, there are two types of documentation stories. Most stories track work to document functionality being created by the development team. We also have stories relating to documentation-specific work, such as creating a new DITA map or conducting an end-to-end review. These stories are much less common, however. Sometimes, teams like to put all documentation work into a single catch-all story, but, while it's easier to create that type of story, it tends to put more pressure on the writer and be less descriptive of the actual work being accomplished. Instead, stories should be defined around a specific chunk of work (much as a development story is defined). Sometimes a development feature can affect multiple areas of the content and may spawn

multiple documentation stories, and sometimes the documentation for several development stories can be combined into a single documentation story, if it only affects a small amount of content. The key is to divide the work logically in small pieces. It is also invaluable to link the documentation stories to the development stories with which they are associated.

Assigning story points to a documentation story allows us to track and measure the work writers are completing, and it allows the writers data to use to raise an issue when they are being overloaded by multiple teams.

Quality Review

We conduct three levels of quality review on our content:

- We have an internal quality process that is run by our technical editor in the Information Development team. This is usually completed for each release cycle.
- At the story/sprint level, the scrum team reviews the documentation created in each story for completeness and flow. Usually the developer working on the feature and the Product Owner are the main reviewers, and this is part of the definition of done for the documentation story.
- At the end of a release cycle, we invite all stakeholders to participate in an end-to-end review of all user documentation, to ensure that we have caught all areas needing documentation. Final sign-off on the documentation comes from Product Management.

Note that our teams are currently working on quarterly and monthly release cycles. As we move to more continuous delivery, we foresee these quality reviews becoming more and more compressed.

Final Suggestions

With more and more software development teams switching from traditional waterfall to agile methods of development, technical communicators need to be flexible about their processes to work cleanly and efficiently. Scrum teams vary greatly in how they apply Agile methodology, and, unfortunately, writers sometimes feel that they should not take an active role. I would advocate for writers to become

full participants in the scrum process – attend all ceremonies and speak up if you are having problems.

I would like to encourage all technical writers working in an Agile environment to learn about Agile, advocate for the writer's role on the team, and do not hesitate to make your presence known. Let the team know early if you have a blocker or if Agile processes are not being followed. Technical communicators also have the skills – organization and attention to detail – that make them ideally suited to leading an Agile team as scrum master, if they choose to do so.

However you make the leap, and however your team works, the keys to success are to adapt and learn as much as you can. Be the expert.

Resources

Sutherland, Jeff, Rini von Solingen, and Eelco Rustenburg. *The Power of Scrum* (North Charleston, SC: CreateSpace), 2011.

Sutherland, Jeff. Scrum Master Training by Scrum Inc. (2015). <https://www.scruminc.com/>.

References

Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. "Twelve Principles of Agile Software." Agile Manifesto (2001). <http://agilemanifesto.org/>.

Author Contact Information

Jane Wilson
Director, Technical Writing
GE Digital
2623 Camino Ramon
San Ramon, CA 94583

Author Biography

Jane Wilson is a technical communicator with more than 18 years of experience working in content development, design, and management in the software industry. She currently works as Director

of Technical Writing at GE Digital in San Ramon, CA, where she leads a team of technical communications professionals to deliver quality content for the Brilliant Manufacturing/Automation applications.

A strong advocate for volunteering, Jane is completing her second term as STC Treasurer, and was recently elected to be the incoming STC Vice President for 2017-2018. Previously, Jane served two terms as Treasurer for the STC Atlanta Chapter, in addition to participating on several committees. She has been an STC member since 2003, and is currently a member of both the Atlanta and the East Bay Chapters.

Jane is a graduate of Georgia Tech (BS, Industrial Management), University of Florida (MA, English), and University of Georgia (PhD, English).



Woelk, Ben

Follow the Yellow Brick Road: A Leadership Journey to the Emerald City

Have you ever felt like your professional journey has a surprise around every bend? Have you thought about how the characters in the Wizard of Oz faced issues similar to what you face on your own journeys? How can you apply the lessons their challenges provide? Let's take a journey along the Yellow Brick Road and see what lessons we learn about leadership from the characters we encounter along the way. (Note that the leadership examples in this paper are drawn from both the book and the film versions of The Wizard of Oz.)

Keirsey Temperament Types and Leadership

David Keirsey has written extensively on temperament and leadership. The leadership attributes in my spotlight talk are drawn primarily from *Please Understand Me II*. The application of temperament types to the four main characters in the Wizard of Oz comes from the following passage in Keirsey, ***Please Understand Me II***, chapter 2:

Once upon a time, in the land of Oz, four individuals set out on a strange and dangerous journey. Each of them was lacking something vital to his or her nature, and each wanted to find the great Oz and ask him for his help.

Lion was lacking courage. Although a powerful, magnificent beast, he had grown cowardly and lost his self-respect, and he wanted the Wizard to give him back his nerve. "As long as I know myself to be a coward I should be unhappy," he said.

Dorothy was afraid she had lost her way home. "Toto," she said to her little dog, "I don't think we're in Kansas anymore." Though never really lost at all, Dorothy felt

stranded and alone, and she wanted Oz to return her to the security of her Aunt and Uncle's farm. "There's no place like home," she said.

Tin Woodman believed he had no heart. As tender and sensitive as he was, he felt stiffened with rust and unable to love, and he wanted Oz to help him feel a warm, loving heart beating in his chest. "No one can love who has no heart," he said.

Scarecrow thought he had no brain. Although the most ingenious of the four, he considered himself witless and worthless, and he wanted Oz to make him smart. "Brains are the only things worth having in this world," he said.

Arm-in-arm, these four very different characters set off to the Emerald City to seek the Wizard's help.

The key characters in the Wizard of Oz all provide examples of the four main temperament types in Keirsey Temperament Theory:

- Dorothy Gale as Guardian
- The Scarecrow as Rational
- The Tin Woodman as Idealist

- The Lion as Artisan

Throughout the course of their travels to the Emerald City, each character exhibits leadership attributes in different settings.

Dorothy Gale: The Guardian Leader

Keirse describe Guardians as the “glue” that holds society together. Guardians are concerned with order, right actions, and providing a secure environment for those under their charge. They are also helpful and concerned with the welfare of others. Guardians are stabilizing leaders. In the *Wizard of Oz* book and movie, Dorothy is the glue that holds the travelers together and provides stability to the party. Although Dorothy provides stability and security to the party, she is seeking a way to return to her secure environment: home with Auntie Em and Uncle Henry.

Dorothy does not receive anything from the Wizard and has had what she needs to achieve her goals after taking the silver/ruby slippers.

Lion: The Artisan Leader

Keirse describes Artisan leaders as being practical, with an eye to the realities around them. They deal with concrete problems (clear and obvious problems, not abstractions about what might occur.) and will do whatever it takes to solve them. They are expeditious and move rapidly to solve a problem. They are at their best in on-the-spot decision making. They are impulsive and prefer to “fly by the seat of their pants.” They are risk takers.

A few things about the Lion are interesting. He says in *The Wonderful Wizard of Oz* that all of the beasts of the forest are afraid. Repeatedly, he demonstrates that he possesses the courage he believes he lacks. He appears to confuse courage with inward confidence, not realizing that courage is doing what needs to be done against whatever odds, even though you may be scared to death inside as you are doing it.

In the movie, the Lion is given a gold medal that reads COURAGE. In the book, the Wizard gives him an unknown green liquid substance that he drinks out of a bowl. (At least one commentator suggests that this may be a bowl of gin.)

Tin Woodman (Tin Man): Idealist Leader

In the *Wonderful Wizard of Oz* (the book), the Tin Woodman races into action when the travelers face a fierce physical threat—the Khalids, beasts with bodies like bears and heads like tigers. Although, the Tin Woodman (Tin man) is concerned that he not inadvertently injure any animals, including insects, he has no qualms about dispatching animals that attack the party. The Tin Woodman believes that because he has no heart that he has no feelings at all and certainly is not able to express love. Despite this, as an Idealist, he cares deeply for the wellbeing of the party.

The Wizard of Oz provides the Tin Woodman/Tin Man with an “artificial” heart. In the movie, he provides him a heart-shaped clock. In the book, he provides a velvet heart stuffed with sawdust. Once the heart is in the Tin Woodman’s chest, he accepts that he has feelings.

Scarecrow: Rational Leader

Rationals are innovative, typically highly intelligent, and problem solvers. However, the Scarecrow believes he has no brain and considers himself to be witless. To the contrary, the Scarecrow is clearly the most creative in terms of solutions, able to analyze the situation and pose innovative solutions.

In the movie, the Scarecrow receives a diploma. In the book, his head is overstuffed with bran, pins, and needles, with all the requisite puns intended (bran for brain, pins and needles for sharpness).

Wrap Up: Which Wizard of Oz Character are You Most Like?

Are you a Guardian? An Idealist? An Artisan? A Rational? What role would you have played along the journey? How does this inform your leadership style?

Resources

Baum, L. Frank. *The Wonderful Wizard of Oz* (Chicago, IL: George M. Hill Company), 1900.

Fleming, Victor, Director. *The Wizard of Oz* (Loew’s Inc. Hollywood, CA), 1939.

Keirse, David. *Please Understand Me II* ((Delmar, CA: Prometheus Nemesis Book Company), 1998.

Myers, Isabel Briggs. *Gifts Differing*. Palo Alto, CA: Consulting Psychologists Press, 1980.

Stutz, Jonathan. The Scarecrow's Brains: Leadership Lessons from the Wizard of Oz, in CEO World,

Woelk, Ben. An Introvert's Journey to Leadership (*Proceedings*, STC Summit Conference, 2016).

Woelk, Ben. Guest Editor. *Intercom*: Personality, Temperament, and Technical Communication, Vol. 64, #2 February 2017

_____. The Wonderful Wiki of Oz. (<http://oz.wikia.com/wiki/>)

_____. Leadership Lessons from the Wizard of Oz. <https://spectrain.wordpress.com/2010/04/08/leadership-lessons-from-the-wizard-of-oz/> Accessed April 17, /2017.

Author Contact Information

Ben Woelk, Associate Fellow
ISO Program Manager
Rochester Institute of Technology
148 Selborne Chase
Fairport, NY 14450
585.354.6247
Ben.woelk@gmail.com

Author Biography

Ben Woelk, CISSP is Associate Fellow, Former Director, and Scholarship Committee Chair for STC. Ben serves as the Information Security Office Program Manager at the Rochester Institute of Technology, and is a former Co-Chair of the EDUCAUSE HEISC Awareness & Training Working Group. Ben is the author of *Shockproofing Your Use of Social Media: Staying Safe Online* (Kindle), and served as guest editor of the June 2013 Intercom Risk Management issue and the February 2017 Intercom Personality, Temperament, and Technical Communication. Ben provides mentoring for new and emerging leaders and has created an online Introverted Leadership Community on Slack.



Society for
Technical
Communication



**2017
Summit
Proceedings**

May 7-10, 2017
Washington, DC



Save the Date!

2018 Society for Technical Communication Summit

May 20-23, 2018

Orlando, FL