

SOCIETY FOR TECHNICAL COMMUNICATION



PROCEEDINGS

SUMMIT

2016
15-18 MAY
ANAHEIM MARRIOTT
ANAHEIM, CA



TECHNICAL COMMUNICATION
SUMMIT '16
STC's 63rd Annual Conference

#STC16

www.stc.org & summit.stc.org

Notice

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. Only light editing was employed to ensure consistency throughout this proceedings publication.

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.

© 2016 Society for Technical Communication

Contact Information

Society for Technical Communication
9401 Lee Highway
Suite 300
Fairfax, VA 22031
USA
+1.703.522.4114
+1.703.522.2075 (fax)
www.stc.org

Colophon

This publication was produced using Adobe InDesign CC. The heading typefaces used are Myriad Pro and Myriad Pro Condensed, which are licensed by Adobe Systems, Inc. The text is set in Minion Pro, which is licensed through Adobe Systems, Inc. This Proceedings publication was compiled and designed by Michael Opsteegh.

2016 TECHNICAL COMMUNICATION SUMMIT COMMITTEE

The Society for Technical Communication's 63rd Annual Conference focuses on important trends in our profession. This publication contains papers submitted in support of the 2016 Summit conference sessions. This year's conference is the result of the efforts of many individuals, including the Conference Manager, Program Advisory Committee, and staff of STC.

Conference Chair

Todd DeLuca, *Black Knight Financial Service*

Program Manager

Jamie Gillenwater, *Transcend Text, LLC*

Track Managers

Design and Testing

Brenda Heuttner, *P-N Designs, Inc*

Career and Leadership

Liz Herman, *Battelle*

Writing and Communication

David Caruso, *National Institute for Occupational Safety and Health, CDC*

Technology and Development

Marta Rauch, *Oracle Corporation*

Technical Communication Summit program, session, and conference information can be found on Sched at <http://technicalcommunicationsummit2016.sched.org>.

Slides from the presentations can also be found on Sched.

TABLE OF CONTENTS

Career and Leadership

Closing the Deal: Getting that Freelance Project	1
<i>Alisa Bonsignore</i>	
Communicate More Effectively in Your Global Virtual Teams: Lessons Learned from a Global Survey	3
<i>Pam Estes Brewer, PhD, Fellow</i>	
Turning Offshore Partnership Challenges into Success	9
<i>Barbara Giammona, Fellow</i>	
Where Does User Documentation Fit Into Content Marketing?	13
<i>Clinton R. Lanier, PhD</i>	
Your Three Hats	19
<i>Rich Maggiani, Fellow</i>	
A Practitioner's Guide to Teaching Technical Communication	21
<i>Michael R. Opsteegh</i>	
Lessons My Cat Taught Me about Instructional Design	27
<i>Jamye Sagan</i>	
An Introvert's Journey To Leadership	31
<i>Ben Woelk</i>	

Design and Testing

Usability Testing	35
<i>David Bachmann and Mandy Morgan</i>	
An Extreme User-Centric Approach to Kiosk Writing	39
<i>Yu-kwong Chiu, PhD</i>	
From Fred Flintstone to George Jetson: Creating Tension in Training Increases Adoption	45
<i>Viqui Dill</i>	

Help Begins with the GUI: TC Meets R&D	49
<i>Leah Guren, Fellow</i>	

Independent Consulting

Understanding and Answering a Request for Proposal	55
<i>Brian J. Lindgren</i>	
The Client Said Yes: Now What?	59
<i>Teresa S. Stover</i>	

Strategy and Trends

Beyond Creating Content: We Can Influence the User's Experience	63
<i>Michelle M. Gardner</i>	
Saving Institutional Knowledge Before Baby Boomers Retire	69
<i>Betsy Kent</i>	
Preparing Legacy Help Projects for Responsive Design	73
<i>Neil Perlin, Fellow</i>	
Providing a Great Search Experience	77
<i>Jeff Stein</i>	

Technology and Development

Collaborating and Contributing in GitHub	81
<i>Nicky Bleiel, Fellow</i>	
Editing Tons of Text? RegEx to the Rescue!	85
<i>Eric Cressey</i>	
Using Word to Automate Writers' Processes	89
<i>Robert Delwood</i>	
The Sound of Help—Improve Your Voiceovers	93
<i>Bryna Ann Gleich</i>	

Release Notes Automation	103
<i>Tanya Ivanova and Barry Grenon</i>	
Programming Language Storytime	111
<i>Sarah Kiniry</i>	
REST API Documentation Best Practices	117
<i>Marta Rauch, Fellow</i>	
Getting Creative with Technical Communications	121
<i>Liz Roscovius</i>	
Let The Force of APIs Be with You: Building the Developer Experience Right	125
<i>Nabayan Roy</i>	
The Making of a Learning Platform: Designing Addictive, User-Centric Content	129
<i>Pavi Sandhu</i>	
Power to the People: Targeted Content through Dynamic Filters	135
<i>Dustin Vaughn</i>	
DevOps Docs: Fast, Great Content for the Cloud	139
<i>Vanessa Wilburn</i>	

Technical Editing

Cash in on your Financial Editing Skills	145
<i>Jessica Kreger</i>	

Writing and Communication

BP/Horizon Oil Rig Explosion, GM Switch Malfunction, and Newark Water Contamination Debacle	149
<i>Carolyn Boiarsky</i>	
Building Quality Experiences for Users in any Language	159
<i>John Collins</i>	

Single-Sourcing and Localization: Maximizing Content Across Languages	165
<i>Laura Dent</i>	
Self-Publishing for Print and eBook Distribution	169
<i>Richard Hamilton</i>	
Applying Minimalist Writing Principles to Improve User's Guides: Readers and Writers Win	173
<i>Michael Harvey, Fellow</i>	
Frenemies: Tech Comm and Tech Support Working Together	177
<i>Rick Lippincott, Associate Fellow</i>	
Simultaneous Product and Information Development	181
<i>Janice Manwiller</i>	
Editors and SMEs: Building Productive Working Relationships	185
<i>Ann Marie Queeney</i>	
Using PDF as Source	189
<i>Liz Roscovius and Mike Sawyer</i>	
Improving the Review Cycle: Concurrent Editing	195
<i>Mike Sawyer</i>	
FAQs Should Support Documentation, Not Replace It	199
<i>Stephanie Saylor</i>	
Your Secret Weapon: The Documentation Handbook	205
<i>Kate Schneider</i>	

CAREER AND LEADERSHIP

Closing the Deal: Getting that Freelance Project	1
<i>Alisa Bonsignore</i>	
Communicate More Effectively in Your Global Virtual Teams: Lessons Learned from a Global Survey	3
<i>Pam Estes Brewer, PhD, Fellow</i>	
Turning Offshore Partnership Challenges into Success	9
<i>Barbara Giammona, Fellow</i>	
Where Does User Documentation Fit Into Content Marketing?	13
<i>Clinton R. Lanier, PhD</i>	
Your Three Hats	19
<i>Rich Maggiani, Fellow</i>	
A Practitioner's Guide to Teaching Technical Communication	21
<i>Michael R. Opsteegh</i>	
Lessons My Cat Taught Me about Instructional Design	27
<i>Jamye Sagan</i>	
An Introvert's Journey To Leadership	31
<i>Ben Woelk</i>	



CLOSING THE DEAL: GETTING THAT FREELANCE PROJECT

Alisa Bonsignore

“How do you close the deal?” is one of the most frequent questions that I’m asked by my fellow freelancers and small business owners. We’ve all struggled with the frustration of the client that drops off the face of the earth after seemingly positive talks. The answer lies not in closing the deal, but in opening it.

If you’re independent, you’ve been there. You’ve had positive talks about the project. You’ve presented your proposal. You’re ready to get started and... crickets. The prospective client stops returning calls or emails as if they’ve vanished into thin air.

(Cue the sad trombone.)

So what can you do about it? To borrow from Inigo Montoya from *The Princess Bride*, “When a job went wrong, we went back to the beginning.” So let’s start there.

Step 1: Set the Stage

How are you presenting yourself? Does your image — both in-person and online — convey an aura of professionalism and confidence?

In person: Are you dressed like a professional? Are you personable? Are you memorable for the right reasons? Would people want to work with you?

Online: How do you appear? Do you have a professional website? Samples of your work? Professional references? An updated LinkedIn profile (without typos)? Or does it link to your Facebook page with photos from your drunken weekend?

Never forget that you are your own PR department, and Google is your resume.

Step 2: Find Prospects

Prospects come in three different ways: cold calls, inbound inquiries, and the much-beloved referral.

Cold calls: People still advocate for cold calls, though I’ve personally had no success. At the very least, make it an educated cold call. Do your online research to not only find the right person to target, but also to figure out what their pain points are.

Inbound inquiries: This is why we have websites and LinkedIn profiles. Sometimes people are looking for something specific, and there we are. But they come unvetted. Be sure to do your legwork, not just about the company, but about your primary contact as well.

Referrals: These clients have been sent along by people who know us. And because no one wants to burn a professional bridge by making a bad connection, referrals are the highest quality leads. Do some quick research into the company, but also look for a way to make a solid connection with your primary contact. Google will quickly reveal their interests, family status and favorite vacation spots. Use those to form a solid bond.

Step 3: Project Research

Listen, listen, listen.

This is where my strategy changes. Up to this point, I recommend a very active process. But when it comes to the project itself, just listen. Don’t try to be the smartest per-

son in the room. Let them talk so you can read between the lines. Listening gives nuance and detail that talking will never uncover.

Step 4: The Proposal

The proposal is where we make everything clear. We want to make sure that we're all on the same page about timelines, money, and scope of work. We make sure that what they want meets with what we can deliver.

A solidly designed proposal reinforces their perception of us, just like the work we did in Step 1.

Step 5: Closing the Deal

Now here we are, at the final step. By following steps 1-4 we've overcome objections before they happened. Do they know that we're professional? Check. Have we been vetted by others to prove our reliability? Check. Have they seen samples of similar projects? Check. Do they know exactly what will be delivered when, and at what price? Check.

First, ask if they need any more information from you. "Have I answered all of your questions? Who needs to sign off on this? If anyone else in your organization has questions, please contact me, or let them know that my references are available on LinkedIn, and samples of my work are on display on my website."

Remember the proposal? When submitting the proposal, tell them that you've already set aside the time for their project. "I've cleared my schedule and I'm ready to begin work on Monday, June 6, pending receipt of the signed proposal."

Always include an expiration date on the proposal. "This proposal is valid through June 3." This gives you a natural justifiable reason to follow up if they don't respond.

And now, finally, you can close the deal. "I'm looking forward to working with you. Are we ready to proceed?"

Conclusion

For the most successful freelancers, closing the deal starts with setting the stage, subtly addressing concerns and overcoming objections before they become deal breakers.

Author Contact Information

Alisa Bonsignore
Writer & Strategist
Clarifying Complex Ideas
Pleasanton, California
hello@clarifyingcomplexideas.com
+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 has been elected to serve as Director of STC (2016-2018), and is has served as Chair of the Education Advisory Panel and a session reviewer for the STC Summit. She is the former Chair of the New Educational Products Committee for the American Medical Writers Association (AMWA), and the Conference Committee for the Public Relations Society of America (PRSA) Health Academy.



COMMUNICATE MORE EFFECTIVELY IN YOUR GLOBAL VIRTUAL TEAMS: LESSONS LEARNED FROM A GLOBAL SURVEY

Pam Estes Brewer, PhD, Fellow

This proceedings article presents some results from an international survey on virtual team practices among engineering professionals. Two hundred and eighty-one participants, representing 48 nations, provided insight into their organizations and global practices. The data teach some lessons about what modern organizations are doing, not doing, and thinking regarding virtual teams as part of their business model.

“With 3 teams we designed several facilities in a refinery in 75% [of the scheduled] time, working virtually. We developed a lot of tools to help with different teams and make all documentation.”

“Other teams don’t want to work virtually, refusing all arguments about virtual meetings, sharing data; we had to drop a big project after a lot of work just because one person on the team demanded everything face-to-face and on paper.”

“During a three-year project in Germany, I transitioned from ‘face-to-face’ to ‘remote’ successfully while delivering critical functionality. This was during the downturn in the US and allowed our company to be successful while still maintaining our presence in the US.”

“Virtual teams are incredibly burdensome and difficult to maintain. Best to have a focal group in all geographical locations to coordinate efforts and synchronize between them.”

“Several years ago, while developing a system using Oracle products, we were able to establish and utilize a global team of technical problem solvers to provide solutions to problems essentially overnight. Some of the issues took days to resolve but this was better than the weeks it would have taken had we not used the global team.”

“I have great reservations about people increasingly working in isolation. It seems to me an increasing trend driven more by financial requirements than efficiency or effectiveness. Using technology to replace direct interaction, unless it cannot be avoided, does not allow a close working relationship to be developed. I prefer to have my team close to me, social interaction builds strong teams.”

The above quotations chronicle the successes and concerns of global engineering professionals regarding their experience with virtual teaming. These teams rely more heavily on technology than on face-to-face communication to work together toward common goals.

And the big irony? Many organizations put these teams at the forefront of their business strategy, often with little or no strategic planning. In this paper, Dr. Brewer shares some of the results of an international survey on virtual team practices in the field of engineering. The data provide some insight into what organizations of all sizes are doing, not doing, and thinking as they leverage virtual teams in their business models.

Methods

This online survey was conducted to examine how virtual teams are being used globally among engineering professionals. The study was approved by the Appalachian State University IRB and was piloted among 10 participants.

Career and Leadership

One goal of the pilot was to ensure that the survey was articulated clearly for a global audience including people whose first language is not English. Pilot participants included professionals from Japan, China, Iran, Belgium, and Spain. Once the survey was edited based on pilot responses, it was distributed primarily via LinkedIn engineering groups. After this launch, a snowball sampling method was used wherein recipients of the survey were encouraged to share it with other engineering professionals. The survey ran in February and March of 2013. Participants need not have been engineers to participate but were required to work closely with the engineering process in a professional role. Participants were also required to read and understand English.

Findings

This section includes some of the significant findings from the survey. Data presented here include demographics, current uses of virtual teams, and predictions for the future use of virtual teams. Of the 281 responses received, more than 67% were provided by professionals outside the U.S. Among the 281 participants, a majority reported experiencing communication problems in their virtual teams, and many of these professionals emphasized the costs of miscommunication in virtual teams.

Demographic Information

Demographics of participants varied widely providing some breadth of perspective in the data collected. Approximately 53% of participants were aged 26 to 45 years. They were highly educated; 83% of participants held undergraduate or graduate degrees. Sixty-eight percent of participants held citizenship outside the U.S., and the top five nations represented were the U.S., India, South Africa, the U.K., and Pakistan though participants represented 48 different nations in total (see Figure 1).

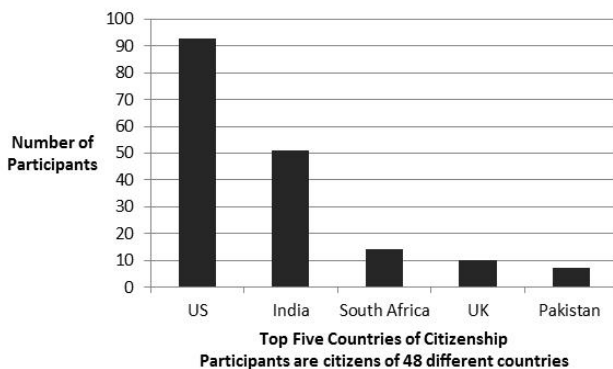


Figure 1. Citizenship of participants (266 responses)

While the survey collected much additional demographic data, one other demographic may be of special interest to STC members: organization size. Figure 2 indicates that the organizations represented by survey participants varied from small to large. Forty-eight percent of participants represented very small or very large organizations.

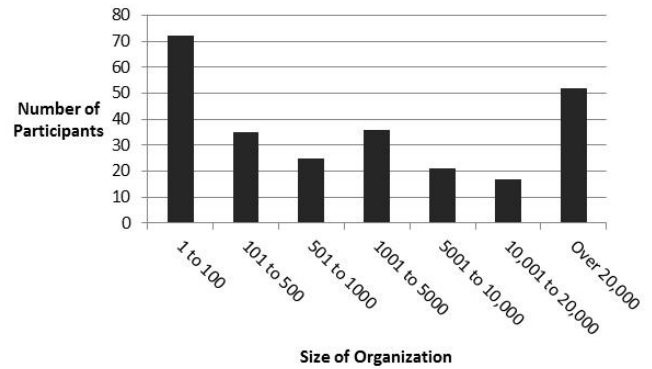


Figure 2. Sizes of organizations represented by survey participants (258 responses)

Current Uses of Virtual Teams

After gathering demographic information for these international engineering professionals, the survey began to gather information on their use of virtual teams. When asked whether or not their organizations used virtual teams for critical tasks, 69% of participants indicated that their organizations do use virtual teams significantly. Survey participants also provided a good look at how their organizations are using virtual teams to support their strategies. As Figure 3 indicates, the most often-cited reason for using virtual teams was to connect employees within an organization.

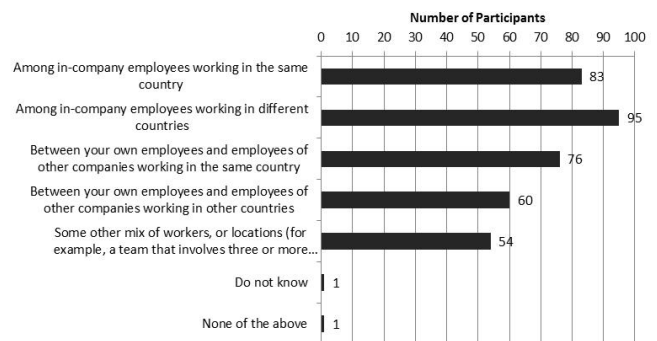


Figure 3. How organizations use virtual teams (156 responses)

Also, participants indicated that they use virtual teams in all stages of the engineering life cycle, but as Figure 4 shows, most heavily in design and development.

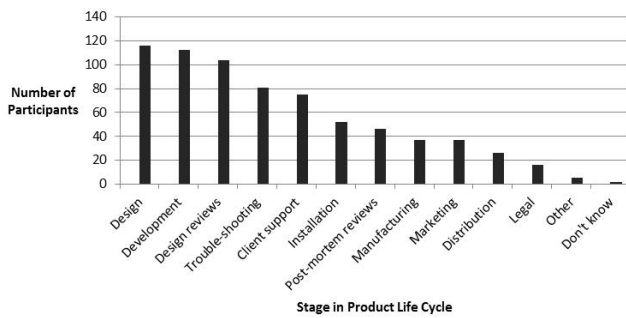


Figure 4. Participants indicated that they use virtual teams in all stages of the product life cycle (156 responses)

Participants indicated that 83% of the virtual teams they work in have international membership and went further to estimate how internationalization contributes to problems with communication in their teams. The reach of virtual teams globally is phenomenal when you consider that some countries do not have labor laws that account for virtual work (e.g., Brazil). In addition, the majority of participants said that their organizations provided no formal training to prepare people to work in virtual teams.

As Figure 5 shows, 39% of participants said that they saw no difference in the impact of miscommunication among non-international and international colleagues in their virtual teams. On the other hand, 30% of participants indicated that they experience a somewhat higher impact from problems during international communication.

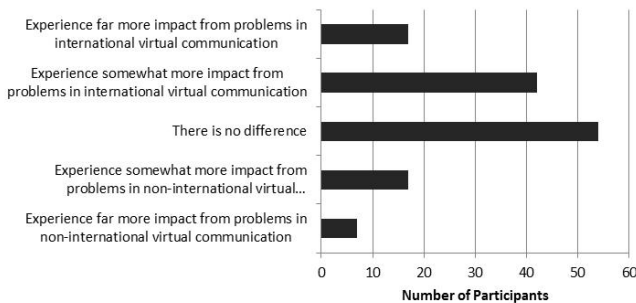


Figure 5. Participants compare how international and non-international communication impacts the problems they experience in their virtual teams (137 responses)

In order to learn more about the origins of communication problems among participants, Thomas and Bostrom’s Trigger Model (2010) was used, and participants were asked to identify the categories in which they most often experienced communication problems. See Figure 6.

As you can see in Figure 6, trust and relationship inadequacy was identified as the factor most likely to cause miscommunication in virtual teams. This factor is defined

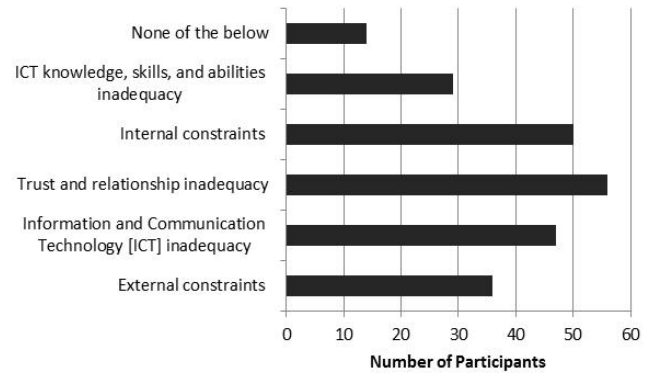


Figure 6. Factors most likely to cause communication problems in virtual teams

as “interpersonal misunderstandings and attitude problems among team members” (Thomas and Bostrom 2010, p. 130). The second most cited factor contributing to miscommunication in these teams was internal constraints which include the conditions internal to the team such as team dispersion, size, and culture (Thomas and Bostrom 2010). Thus, while the tools (both their characteristics and how people perceive them) can cause problems, the majority of communication problems have to do with the people themselves.

The survey included other questions that unpacked the causes of miscommunication, and it included questions on how these professionals sought to resolve the problems that occurred. In Figure 7, note that explicit communication is most-often cited as a method for dealing with miscommunication. The categories used in Figure 7 were first identified in an interview study conducted in 2008 and reported in several articles (e.g., Brewer 2010). They were used in the survey to help engineering professionals categorize their problem-solving strategies.

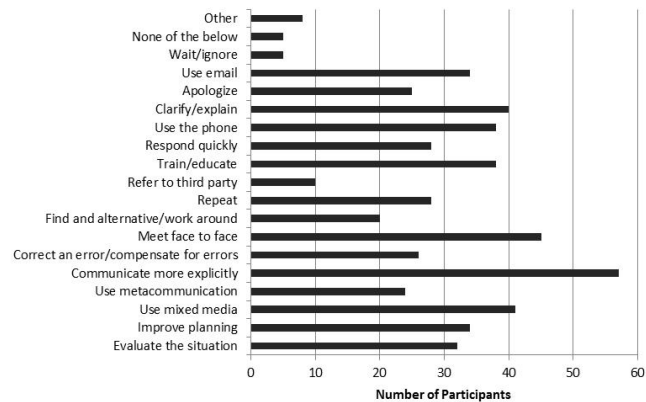


Figure 7. Participants indicated which solution strategies they most-often used in resolving communication problems within their virtual teams (145 responses)

Future Uses of Virtual Teams

The survey also presented a number of questions to gather participants' predictions for virtual team use in the future. First, participants were asked to note how important they perceived virtual teams to be to the future success of their organizations (see Figure 8).

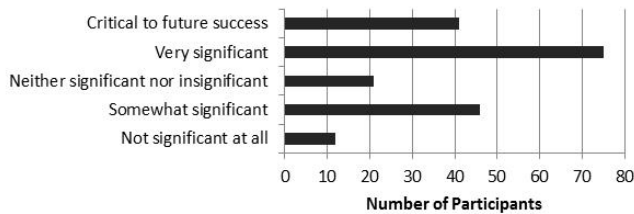


Figure 8. Participants indicated how significant they believe virtual teams are to the future success of their organizations (195 responses)

As Figure 8 indicates, 83% of professionals believe virtual teams will be important to varying degrees. Only 17% thought virtual teams were insignificant to the future of their organizations. The survey questions continued to probe for participant projections for the future. When asked what trends in virtual teaming would prove most significant in the next five years, many participants predicted that the use of virtual teaming would continue to increase and at a very rapid rate; several participants referred to virtual teaming as a mega-trend. Other popular predictions included the increase of:

- Integration with world-wide work teams
- Collaboration on mobile devices
- Virtual project management

Some other predictions (of 170 responses) are included in the list below:

- “Web conferencing will merge with social media and take on gamification rules (badging, leveling, and achievements).”
- “I think outsourcing, per se, will decrease over the next few years, and then stabilize at equilibrium. With tools such as oDesk [now UpWork], virtual teams will be alive and well over the coming years. All bets are off, however, if foreign countries try to abandon their financial competitive edge.”
- “Return to valuing f2f meetings for some tasks.”
- “Open development teams such as apps developers and even mechanical engineers. Companies can put the requirements out and have anyone bid and provide services.”
- “Longer non-traditional working hours.”

- “Virtual teams working in the cloud especially for design and development.”
- “It isn’t popular in China—unsure how this trend might change.”
- “Perhaps wishful thinking but more collaboration between government, business, and education.”
- “Increased use of BIM [Building Information Modeling].”

Conclusion

Participants provided a great deal more data including greatest successes and failures. They demonstrated that members of virtual teams come from widely varying age groups, organization sizes, countries, organization levels, and points in the product life cycle. They emphasized that the success of these teams does not lie in the tools but in the people: how they work, interact, and build relationships. They also told us that the investment in highly functioning teams is often high in the short-term but well worth it in the long-term; that trust and internal team make-up can most affect success in communication; and that explicitness is a go-to approach for problem solving.

The data from this global survey provide many messages for those of us who work in global organizations. Perhaps the most important is that organizations need to become more deliberate in establishing and maintaining their virtual teams. At the time of this survey, organizations were still “accidentally, incidentally” employing virtual teams to underpin their global markets. They put these teams at the forefront of the business model with little or no planning, training, and strategic planning. Global organizations should deliberately implement strategies (Brewer 2015) to increase the effectiveness and reliability of their virtual teams.

References

- Brewer, P. E. 2010. “Miscommunication in international virtual workspaces: A report on a multicase study.” *IEEE Transactions on Professional Communication* no. 53 (4):1-17.
- Brewer, P. E. 2015. *International Virtual Teams: Engineering Global Success*. Edited by Traci Nathans-Kelly, IEEE PCS Professional Engineering Communication Series. Piscataway, NJ: IEEE Press Wiley.
- Thomas, D. M., and R. P. Bostrom. 2010. “Vital signs for virtual teams: An empirically developed trigger

model for technology adaptation interventions.”
MIS Quarterly no. 34 (1):115-142.

Author Contact Information

Pam Estes Brewer
Associate Professor
Mercer University
School of Engineering, Department of Technical
Communication
1501 Mercer University Drive
+1 478.301.4146
brewer_pe@mercer.edu

Author Biography

Dr. Pam Estes Brewer is a technical communicator and educator. She is an STC Fellow and teaches in the Department of Technical Communication in Mercer University's School of Engineering. She also directs the Master of Science in Technical Communication Management online program. She researches virtual teaming, and her book entitled *International Virtual Teams: Engineering Global Success* was published in 2015 by Wiley. Visit her new blog at <http://virtualteamsinternational.com>.



TURNING OFFSHORE PARTNERSHIP CHALLENGES INTO SUCCESS

Barbara Giammona, Fellow

In 2009, my company signed an exclusive contract with an offshore partner in India to provide development services, including technical writing. After a 2 ½ year struggle, we abandoned the effort and brought everything back onshore. But when new senior management insisted that I try again (and insisted that I'd be evaluated based on the success of my efforts) my team and I were motivated to make it work. This is the story of how we created a successful offshore partnership that delivers technical documents on-time and to a level of quality that meets our organization's needs. I'll discuss the changes we made that led to our success and how we deal with ongoing challenges together with our partner team.

Disclaimer: This saga represents one woman's experience – an experience focused on partnering with an offshore vendor in India. Having spoken to other managers who have walked a similar journey, I think there are many generalizations presented that can apply across the board, regardless of the country to which you are outsourcing. In the end, it's all about learning continuously and becoming a more global citizen.

What we actually got was:

- A team with an average score of 55% on a basic writing skills test
- A lot of very junior people
- A group that was very separated from us functionally
- An onsite manager who didn't understand the work that needed to be done

Chapter One: Getting Started Offshore

In 2009, our company signed a contract with an offshore partner that was, at that time, the third-largest outsourcing company in India. They were engaged to provide our R&D organization with a variety of services: Development, Test, and Technical Writing. It was to be a “managed-services” agreement – we would throw work “over the wall” to them. And with their expertise in technical writing and project management, they would do the work and throw it back “over the wall” to us, completed and ready to go.

Based on the information we received in the sales process, we were led to believe we would be getting:

- English Writing Skills
- Editing Skills
- Technical Writing Skills
- Management Skills

We did an analysis after two years and determined that:

The North America team spent the same number of hours per page overseeing and reworking the work done by the offshore team as they spent in the past writing the materials themselves!

We spent nearly \$120,000 on a single group of projects for work done offshore that added NO value

It was clear that things were just not working. We decided to pull out from the partnership and call it a day.

Chapter Two: Starting Over

In early 2013, I got a new boss. As the former head of an India-based consultancy, he had full faith in the abilities of our partner. He urged me to try again. Well, urged would

Career and Leadership

be a gentle word for it. My next evaluation would be based on my ability to restart the relationship and make it work. So I was motivated.

I determined that this time around, things would need to change. From the data we gathered from the failed first try, we set up some new parameters for the reboot:

- All new writers added to the offshore team would be tested and would need to pass our test with a minimum score of 75%. (Granted, I wouldn't hire a writer in North America who didn't score at least 95% on the same test, but we had to set an achievable goal. Over time, with more careful screening of candidates, we found that the best writers we were recruiting were scoring in the 80s on this test.)
- We would set up a team structure that would create local product line "lead writers" in India to review the work of the more junior team members. As budgets allowed, these leads made short-term trips to our North America offices to train under our lead writers.
- That we would pull all the writers, supporting all of our product lines, under our direct leadership and not have any individual writers working alone on Development teams offshore. We found that the writers needed to be led and managed by other writers in order to grow and learn, and not serve as disempowered, untrained members individual of the offshore Development teams.
- Lead writers on my North America team mentored, edited the work of, and held the offshore teams accountable for their performance.

We also developed meaningful metrics to monitor the quality of the work being done offshore. It was not practical to count and report errors. That was too time consuming and would not help us improve.

Instead, we devised metrics to help us measure such things as:

- Percentage of effort spent on writing tasks (should be the majority of their time)
- Percentage of effort spent on making review fixes (should be small and go down over time)
- The dependency the writers had on their Subject Matter Experts. We wanted to demonstrate that the writers were developing meaningful relationships with their SMEs, AND were learning about the products to develop their own independence

(hence, they should spend SOME time with SMEs, but rely on them less over time)

- Percentage of time spent on training/developing product knowledge (should be steady – learning is an important uses of time)

Making It Work

There is a high value placed on rapid career advancement in India. As those of us who work in other parts of the world know, technical communication, while rewarding work, is not a place from which many of us rise quickly to the highest ranks in a company.

Thus, turnover is high in this field. Turnover is also high because it is a profession largely taken up by young women. In a culture where women are still marrying relatively young, starting families, and may be charged with the care of aging parents, we have found that our best and brightest writers are often called away to other responsibilities just as they are reaching their peak effectiveness.

These challenges, coupled with the daily challenges of life in India—a country of many contrasts – has certainly made it hard, at times, for our global company to get the most out of our India-based team.

In light of these challenges, we implemented some additional changes on our side to make the relationship work. The result is that we are getting work done on time to a level of quality that satisfies those who are requesting it.

New Roles

We needed to recognize that our North America team needed to assume new roles. Instead of being the lead writer who delivered documentation for your product line, you were the lead writer over a distant team. You no longer got to write very much, if at all. Instead, you were a mentor and an editor.

For me, as a team manager over senior people, I found I had a new role, too. I had to mentor my senior people to a new place in their careers – and deal with the fallout if they were not happy in that new role. It often meant looking for a different profile of person when I made additions to my team – people who aspired to lead and manage others and not just be writers themselves. This was a huge cultural shift for our organization.

In the end, the most important thing we learned is that the "managed services" model would not work. We couldn't

assume that the partner team knew the job already and that we could “throw work over the wall.” This had to be a partnership – one where the resources offshore augmented our own staff rather than standing alone.

Process Changes

We had to make sure that we had clear, documented processes, and that the teams on both sides understood them well. One of our greatest accomplishments was successfully moving to the partner team some of the most routine maintenance and production work. This is where we found we got the most value– giving the offshore team tasks that were important and needed to be done, but that required the least expertise about our complex products. Starting with these simple tasks gave them the best opportunity to succeed.

And Finally...

We had to devise metrics to measure our success, know when to reject work that did not meet our expectations, and make sure that we visited the offshore location so that the people on the other end of the line felt like real people to us.

It’s a Small World

Schneider Electric is a truly global company, with locations in more than 100 countries and more than 180,000 employees.

But when I began this journey, I was not such a great global citizen myself. It soon became very clear to me that I needed to make an effort to understand the culture of the country I was engaged with.

My boss, who is from India, reminds me to always to listen not just to the actual words someone uses, but to what they are trying to say. As a native speaker of English, as an American, and as a writer, I always rely heavily on the words people use to understand a message. He was quick to point out that if you don’t know the exact word for something in a language, you can still convey your message – it just may be expressed differently than the listener may expect. This is just one example of the kinds of cultural differences that can arise when working on a global team.

In addition to dealing with cultural differences, we also faced the challenges of forming single working team in a virtual world. Not only is my North America team spread

from California, to Washington, to Houston, to Boston, to Toronto, but my India team is spread from Hyderabad, to Chennai, to Kolkata. The use of effective collaboration tools for file sharing, teleconferencing, web meetings, and video conferencing are also critical to our success. And needless to say, everyone’s work day is stretched beyond the normal confines of the “eight-to-five” working hours. You simply have to be prepared to be part of the new global reality of the 24-hour working day.

Conclusion

It can work. Simple as that. You can engage an offshore partner to do technical writing. It’s not easy and not always ideal. It will stretch all of you professionally, as you learn to accept varying levels of quality for varying tasks, as you develop patience in mentoring new resources, as you devise new processes that work in a virtual reality of the 21st Century work life. The bottom line is that we are a global community and any place you work today is going to ask you to look for low-cost alternatives to support the doing of your job.

With that in mind, I encourage you to take the advice offered here and give it your best effort. It may take some refining, but you might be surprised at what you will accomplish along the way!

Resources

- Reynolds, Sana, Valentine, Deborah, *Guide to Cross Cultural Communication* (Upper Saddle River, NJ, Prentice Hall) 2011.
- Fisher, Kimball, Duncan Fisher, Maureen, *The Distance Manager: A Hands-On Guide to Managing Off-Site Employees and Virtual Teams* (New York, NY, McGraw-Hill Companies) 2001.
- Brown, M. Katharine, Heuttner, Brenda, James-Tanny, Char, *Managing Virtual Teams: Getting the Most from Wikis’ Blogs, and Other Collaborative Tools* (Wordware Publishing, Sudbury, MA) 2007.

Author Contact Information

Barbara Giammona
Director, Technical Publications and Communications
Schneider Electric Process Automation
26561 Rancho Parkway South
Lake Forest, CA 92620
+1 949.639.8354

Author Biography

Barbara Giammona has been a technical communicator for more than 30 years and a manager for more than 25. She is currently Senior Manager of Technical Publications and Communications for Schneider Electric Process Automation. Prior to that, she was the manager of IT communications for McKesson Corporation in San Francisco. She spent 10 years leading documentation efforts on Wall Street, including seven years as a vice president at Morgan Stanley in New York City.

Barbara's was awarded the Frank R. Smith Award for Distinguished Technical Communication in 2004. Her research was republished in 2010 in a compilation entitled Qualitative Research in Technical Communication. She is a frequent presenter on management topics at local and international STC events.

Barbara holds a BA degree in English from the University of California, Irvine, and a MS in the Management of Technology from the Tandon School of Engineering at New York University. Barbara is an STC Fellow and a member of the Orange County chapter.



WHERE DOES USER DOCUMENTATION FIT INTO CONTENT MARKETING?

Clinton R. Lanier, PhD

Content marketing has become a popular method of distributing information about products and brands, especially for Business-to-Business (B2B) products and services. Content marketing emphasizes an approach that replaces traditional advertising for content that people can actually use or want to consume. This content includes information products that have been in use for many years already, like white papers, as well as information products gaining in popularity, like “How To” videos. This paper discusses the role of user and technical documentation in content marketing and specifically presents ideas that technical communicators can use to transform products they already create into marketing content.

Content Marketing is billed as “a strategic marketing approach focused on creating and distributing valuable, relevant, and consistent content to attract and retain a clearly-defined audience — and, ultimately, to drive profitable customer action” (Content Marketing Institute, n.d.). Thus it is an approach to marketing where promotion is concerned with authentic or “real” content items rather than typical advertising.

This “valuable, relevant and consistent content” can include a wealth of information units, like white papers, studies or how-to videos. Importantly, not all of these documents explicitly promote the product. Instead they may help users complete a task or make a decision, and the product implicitly promoted may be a resource for completing that task or making that decision.

Such content, according to this approach, provides information that the user might be searching for that is unrelated to the product, service or brand. When users discover such information they are exposed to the brand, service or product while at the same time gaining the valuable information they were looking for.

This paper will first provide a brief history of Content Marketing, its current use and value, and relevant examples of various types of Content Marketing. Next it will discuss why and how content marketing works with today’s information consumer. Then it will discuss a brief

case that I am personally involved in for a niche, Business-to-Business (B2B) industrial product, which is marketed exclusively through technical content. Finally, it will provide ideas and recommendations for those in the technical communication fields for creating content marketing products out of existing technical documentation.

Content Marketing—History and Examples

Content Marketing, quite honestly, is nothing new. Think about any type of marketing product that is “disguised” so to speak as something else. One of my favorite examples is Rudolph the Red Nose Reindeer. Yes, believe it or not, this lovable hero of Christmas is a product of marketing.

The year was 1939 and at the time the retailer Montgomery Ward published and then gave away coloring books to children. Each coloring book was, of course, branded by the company throughout its pages and on the front and back covers. And so children, only too happy to flop down on the floor with crayons to bring the books’ stories to life, were at the same time being exposed to an implicit (that is, not overt) form of advertising.

One of the stories created for the coloring books was about a small reindeer that the writer named, “Rudolph” who saved Christmas because of his shiny red nose. They probably never expected the popularity to grow as it did, but

the story became an immediate success and the branded coloring book was in huge demand.

We could even say the story viral, as we would call it today, and was picked up in newspapers around the country. In 1949 the song was released that matched the story, and the rest (as they say) is history.

Such past examples abound, but content marketing has become increasingly popular today because of the mass consumption of digital content by online information consumers. This content is published easily through blogs or media sharing sites, and then distributed by numerous social networking platforms, either by the content's producer or by fans of the content.

One modern and poignant example I like to refer to is Volvo's Epic Splits video. If you have not seen it yet, it features actor and martial-artist, Jean-Claude Van Damme standing astride the door mirrors of what look to be two semi trucks. With Enya playing in the background and a Jean-Claude voiceover waxing on about something profound, the camera pans out and we see that the trucks are actually driving. Backwards! Without warning the trucks begin separating from each other, and so Jean-Claude must begin doing the splits to remain astride the mirrors (Kleine-Kalmer, 2016; Volvo Trucks, 2013).

He does this while looking stoically into the cameras, completely emotionless, and arms crossed. Within seconds the trucks are now far enough away from each other, still moving backwards at the same speed, that Jean-Claude is completely doing the splits between them, feet still firmly atop the mirrors.

At the time of this writing, the original one minute, sixteen second video had been viewed 82,978,420 times.

The video, while immensely entertaining, was produced by Volvo Trucks to highlight the precise steering mechanism used in their large, commercial vehicles. By all accounts, from a marketing perspective, it was a success, as the number of views, shares, likes and spawned imitations demonstrates.

But why was it a success? And why, for that matter, was Rudolph a success? The answer to both is that they simply were not advertising (advertising as people have come to know it, anyway). In this case, these two pieces of "content" were entertaining enough to make people want to engage with them (as opposed to traditional advertising that makes us want to turn the channel, the page or otherwise ignore what's being sold). And while "creating and distributing valuable, relevant, and consistent content," each

brand at the same time created a "buzz" about that content and enhanced their value.

Now while these are both great stories, these are probably outliers. In truth very few examples of content marketing occur like this, and instead, for professionals in technical industries, they are much less colorful, yet they are at the same time just as important.

The next section briefly discusses why and how content marketing works for technology products and why it is becoming much more valuable now than it ever has been.

User-Networks and Content Marketing with Technical Communication— How and Why it Works

Technology users want their questions instantly answered. They want to know immediately how to solve a potential problem, how to use something correctly, or how to come to the right solution. They have been trained to do this – to instantly find the help they need.

Fifteen years ago they would have been forced to call a product's help line, to stay on hold or to be transferred to the correct department, to pay a fee for information, and to ultimately go through a lot of trouble to solve their problems. But today they simply do a keyword search on Google. The number and type of references available to them are staggering: forums, blog articles, videos and even conversations on social networks. All of these provide potential answers to their questions.

It is important to understand that none of these resources are long, complex information products. The information products that users look for are short, easy to navigate, and quickly answer their question. They are also portable – quickly pulled up and viewed on mobile technology with little effort.

The questions that users have, and the content that answers them, represents a moment in time for the user or customer (or potential customer) (Swarts, 2015, pp. 20). The user has a specific question at a specific time and so searches for the specific answer. It may be something basic, like the price or features of a product, but it may also be something more technical, like a quick overview of a product's use or its features in operation. Or, and perhaps more frequently, it may be a question about something related to the product.

Content marketing for technology companies is successful when those companies provide resources to answer those questions.

And these resources must be different than the traditional set of documentation – user manuals, installation manuals, etc. – that have always been provided. The design of those types of information products cannot possibly conceive of every specific context of use for the products they are written about, and so really do not help users who want to know something specific at a particular moment.

And knowing this can help technology manufacturers understand the opportunity they have to enhance their brand, their reputation and their product's use by providing a wealth of content that anticipates the informational needs of users and customers.

The next section helps illustrate this by discussing a case that I am personally involved in. Although it is nothing like the dramatic examples of Jean-Claude doing epic splits or of a reindeer saving Christmas, the case illustrates that content marketing through technical documentation and related products, can work.

Case Example: the HydraSleeve

I began consulting for a small company called GeoInsight in 2010. I have had various roles as their consultant but mainly it was to help them organize their digital presence and create technical information for their product. That product, called the HydraSleeve, is a very niche product for the water sampling industry. It is, essentially, a plastic sleeve that can be lowered into a well and then brought back up, taking a water sample from the well.

Of course there is more to it than that, all of which is quite technical. For example, the water sample has to be representative of the water that is flowing through the well. The HydraSleeve also has to be able to capture any contaminants in that water. And the use of the HydraSleeve must not affect the quality of the water sample that is captured and then sent to the lab for testing.

The HydraSleeve was entering a market already dominated by other, but better funded technologies for sampling wells. GeoInsight did not have a large marketing budget, and even if they did their venue choices in which they could advertise were quite limited – primarily trade journals and advertisements at conferences, etc.

So instead of these more traditional methods, the owner of the company, Kent, decided distribute as much technical information about his product as he possibly could.

He surmised – aptly as it turned out – that the people who would be looking to buy his product would be just as interested in how and why the device worked as they would be in more consumer-centric attributes (like price, for example).

Kent began by gathering all of the technical publications that mentioned or discussed his product. He found peer-review journal articles, PowerPoint presentations, and brief columns in trade magazines. These he placed in a “technical library” section of the product's website. He directed consumers to that section through a “Don't Take Our Word For It” button on the site.

Kent also produced white papers about the device and discussed its production and use. He told people what it was made of, how it worked and why it should be used. These white papers were aimed at the people who would actually use the HydraSleeve, and so included technical language they would understand and use as well. In addition to the white papers he also produced Standard Operating Guides and brief instructions.

Finally, he created a series of very brief “how to” videos that included the HydraSleeve in actual use. The subjects ranged from simply deploying and retrieving the device to fishing it out of a well if the retrieval line were to break. These videos were hosted on both his site and YouTube.

To distribute all of this content, GeoInsight uses mass email to place a newsletter into the inbox of potential customers each month. Further, as each piece of content is indexed, it is quickly and easily found through internet searches.

This case illustrates how very boring and very technical information, can also be very valuable as marketing tools. It is important to point out here that Kent is not trying to market to a mass consumer, but instead to a very specific crowd, and so he has focused the content he creates on those specific consumers. He understands what information they are interested in – the technical information – and what language they speak, and so generates content that will appeal to their wants and needs.

Many of you are probably thinking, “we do that already.”

True, most technology companies create the white papers, they create the instructions, and they publish all of this on the web. But there are a few things I would like to point out about this case which makes it a bit unique, and which I think could serve as lessons for us.

The first item to consider is how much competition there is for information about the HydraSleeve. This is important, and it should be one of the primary reasons any com-

pany would consider using their technical documentation as content marketing in the first place.

In the case of the HydraSleeve, GeoInsight is really the only entity creating content for the device. Thus, when people search online for information about the HydraSleeve, the information that tops the search query results is information produced by GeoInsight.

This is not always true. If for example you were to enter the query “how to use photoshop” you would see a single result from Adobe, followed by pages of information from other outlets. On the other hand if you enter the same query for the HydraSleeve you see the first four are from GeoInsight followed by those from other entities.

Your content, therefore, especially for technology that is broadly used, is in competition with content made by other people or organizations. But why does this matter?

Simply put, by constantly creating content in as many topics as he can think of related to his product, Kent ensures that he is at the top of search engine query results for that product. This means that he can be assured of the second item to consider from this case, that the information customers, potential customers and users find is accurate.

One of the most important requirements in his industry is the accuracy of the water samples that a sampling device provides. Kent is very aware that, if used wrong, the sampler he manufactures will provide inaccurate results, and then, as a corresponding result, he will sell less of the product. Therefore it is essential that people learn how to use his device correctly and to be aware of the best practices when using his sampler.

If customers and users (and potential users) find and follow incorrect information it could harm his reputation and his sales. Thus Kent tries to anticipate what his customers will do wrong and then will design videos and documentation to provide the best method for solving the potential issue the user may have.

The result of all of this effort has been tremendous sales growth and adoption by professionals and organizations in the industry. The reputation of the HydraSleeve and its manufacturer are very good and professionals who use the product are ready to sing its praises and to tell others about its use. And technical documentation and information, used in the role of content marketing, is one of the foremost reasons.

In Conclusion: Lessons for Technical Communicators

So now we have defined content marketing, briefly looked at its history, relevance, and why it works, and we have also quickly looked at a case of a technical product marketed through its technical documentation and information products. Now we will look generally at content marketing and how technical communicators can contribute to it through the documentation products they produce.

First, as stated previously, when consumers search for information about a product, especially a technology product, they are trying to find out something very specific rather than general information. Thus, the pieces produced should be very small, very specific pieces of information. In our industry, modular documentation is nothing new, however in the past we have used it only as an organizational tactic – producing information modularly allowed us to pick and choose from pieces of information we needed to assemble a final manual or help system.

But now I think it best to find a way to publish the modules themselves, rather than the thing assembled with the modules. As people have a specific question or want to know specific information that corresponds with the information in a specific module, they will find it (via an online search) and then use it.

Another approach is to create brief “how to” guides about as many situations and circumstances as you can find. These should be brief and web-accessible articles that show users how to perform a specific small function with your product. A great example is Microsoft’s Office documentation, which features numerous articles demonstrating how to carry out small tasks with their various platforms (like combining the contents of two columns in Excel).

An important point about the documentation “modules” or the “how to” guides is that these should be posted as web pages – like blog articles – and not as help files. Remember that over 50% of internet users now view the web through mobile devices, and so the format should be responsive and easily read on these mobile devices (including both tablets and smart phones, among others).

Videos are also extremely effective methods of content marketing. Approach these like the above: explain to users how to carry out a multitude of operations and functions with the product. A great example is Adobe’s extensive network of videos created for their different products. Like GeoInsight, Adobe hosts their video tutorials on YouTube, ensuring that they show up in the results of internet

searches. Also like GeoInsight, Adobe offers all of these videos through its website that serves as a video library.

Any and all of this content can then be distributed and broadcast via social media channels, like those belonging to the brand or product. Especially popular right now are “hacks” or methods of using products to solve some daily or seemingly mundane life problems.

This content can also be shared with users on forums and when contacting support. You can think of it as evergreen content that never gets old and will always be needed. For example in the aforementioned Microsoft Office example, I have both Excel 2010 and Excel 2003 (on different computers, of course), both of which I use often. Luckily, Microsoft provides guidance for its older products as well as its new, and so I can find solutions for either version.

Ultimately, content marketing is a great fit for technical communications, and one that I think will only grow and become more important in the future. There will be some needed organizational changes, of course, like a closer relationship between technical communication and marketing. But in the end, turning technical information and documentation into content used to market the product to existing and potential customers will be profitable strategy.

References

“What is Content Marketing.” Content Marketing Institute. (n.d.) <http://contentmarketinginstitute.com/what-is-content-marketing/>.

Kleine-Kalmer, Barbara. Brand Page Attachment: An Empirical Study on Facebook Users’ Attachment to Brand Pages. (Wiesbaden, Germany: Springer Gabler), 2016.

Volvo Trucks. “The Epic Split feat. Van Damme (Live Test).” YouTube.com (13 November, 2013). <https://www.youtube.com/watch?v=M7Fivfx5J10>.

Swarts, J. “What User Forums Teach Us about Documentation and the Value Added by Technical Communicators.” *Technical Communication* 62.1 (February, 2015): 19-28.

Author Contact Information

Clinton R. Lanier
College Assistant Professor, Rhetoric and Professional Communication
New Mexico State University
P.O Box 30001 MSC3E
Las Cruces, NM 88003
+1 575.496.6807

Author Biography

Clinton R. (Clint) Lanier has been a professional in technical communication since 2000 when he received an MA in Technical Communication from New Mexico State University. From there he worked as a Software Engineer/ Technical Writer for IBM in Tucson, AZ, and then as a Technical Editor for the Army Research Laboratory at White Sands, NM. After receiving his Ph.D. in Rhetoric and Professional Communication in 2006 he was an Assistant Professor of Professional Communication at the University of Memphis, and then an Assistant Professor of Technical Communication at New Mexico Tech in Socorro, NM.

While at New Mexico Tech Clint was also the Interim Director of Web Communication from 2008-2012. In this role he designed and built the school’s Content Management System, oversaw the contribution of over 90 distributed authors, and planned, produced and carried out the school’s digital marketing campaign including social media marketing. He also developed the school’s mass email marketing mechanism and a video content campaign for recruiting. Largely because of his efforts New Mexico Tech saw record freshman enrollment from 2010–2013.

Clint is now a College Assistant Professor of Rhetoric and Professional Communication at New Mexico State University. He teaches classes and conducts research in topics including digital and new media, technical communication theory, and document design. He is also an active industry consultant.



YOUR THREE HATS

Rich Maggiani, Fellow

All independents and freelancers wear three hats for their business: an Expertise Hat, a Marketing Hat, and a Financial Hat. All three hats must work in concert with each other, and the benefit of each must be made clear to prospects and clients. Each hat governs a different area of their business. Understanding what each hat represents, and fulfilling its role, leads to a successful business.

To best succeed as an independent consultant (never contractor), you must master your three hats. All freelancers and independent consultants wear three hats: one for their expertise, one for marketing, and one for financial sense.

Let's look at this from the metaphor of a luxury liner. Your Expertise Hat is the luxury liner itself; your Marketing Hat is the ability to get passengers to ride and pay your rates; your Financial Hat is your ability to keep the liner afloat and moving forward. All three hats move your liner to its destination. Both you and your passengers benefit.

Let's examine these hats individually.

Your Expertise Hat

Most independents focus on their Expertise Hat, thinking that knowledge, experience, and professionalism will carry the day. I have news for you. Prospects and clients expect, from the very start, that you are competent. To what level that they appreciate your competence is up to you. Your Expertise Hat is most effective when properly paired with your other two hats.

Here is your Expertise Hat at a high level. First, stop calling yourself a contractor. You are a consultant. Your clients should be relying on you for your communication expertise, no matter what form that takes, and not simply seeing you as a plug to fill a hole. Plugs are generic; experts are specific. Plugs are unbranded; experts are branded.

To be an expert, it is incumbent upon you to create your brand, promote it assiduously and arduously, and to pro-

tect it all all costs. Employ your Marketing Hat to create your brand; that is how you promote your Expertise Hat.

Your Marketing Hat

What sets you apart from everyone else? This is the fundamental question you must answer—and answer well—before you can make any meaningful progress wearing your Marketing Hat to create your brand.

You must ensure that prospects and clients fully understand and appreciate your brand—what you alone can bring to the discussion. Any misconceptions must be righted, immediately. This is a tireless, never-ending venture. You are not a national or international brand, nor will you ever be. Resign yourself to this task, for it is fundamental to your professional life and success.

You need to focus on your Marketing Hat to position yourself in the marketplace, to get work, and to keep work. You must define who you are, your brand. Most independents rely on their clients to understand and appreciate their work, and their contribution. Big mistake. First, you need to perform above expectations. You need to don your Marketing Hat and tell your clients about the importance of your contribution. In other words, you can't rely on your clients to appreciate what you do. When you do, you allow your clients to brand you. That's abdication of your Marketing Hat. Instead, you need to frame the message, and communicate that message to them. You need to tell them! For that to ring true, you also must perform.

Your Financial Hat

Success in business centers around one, and only one, element: money. Let me say this a different way: everything you do in business ultimately boils down to how much money your clients are willing to pay. You have succeeded in business when your prospects and clients happily accept and pay for your services, at the fee you specify, without flinching. That happiness is what your Financial Hat is all about.

Here is the foundation of your Financial Hat: set your hourly rate at a level commensurate with what you should be getting paid as an independent consultant. You can set your hourly rate at any amount, so long as it is at least three figures. Did you gasp when you read that? Did you perhaps think that you are not worth at least that amount? If so, refer back to your Expertise Hat and make it so.

Are your current clients unwilling to pay that amount? Does a prospect or client hesitate to pay your fee, or—yikes—have the audacity to negotiate, or set the fee they are willing to pay? In essence, they are clearly demonstrating, at a minimum, reduced respect for you. Always remember: in business, money is the sign of respect.

When these situations occur, do not consider your prospect or client the pariah. Rather, look in the mirror: you have not succeeded in solidifying the value of your brand with them. If so, refer back to your Marketing Hat. Either educate this prospect or client, or find new prospects and turn them into new clients who appreciate you and pay you.

There are many other aspects to your three hats, such as how to excel in your chosen craft, how to reach and land prospects, and how to manage money and cash flow, as well as many, many other factors. All of these, however, follow the fundamental concepts presented here for your three hats.

Wear them well and you shall succeed.

Resources

Jantsch, John. *Duct Tape Marketing*, second edition (Nashville, Tennessee: Thomas Nelson Publishing), 2011.

Hayden, C.J. *Get Clients Now*, third edition (San Francisco, California: American Management Association), 2013.

Meerman Scott, David. *The New Rules of Marketing and PR*, third edition (Hoboken, New Jersey: John Wiley & Sons), 2011

Walters, Jamie. *Big Vision, Small Business* (San Francisco, California: Ivy Sea Publishing), 2001.

Frick, Elizabeth. *Business Matters* (Laguna Hills, California: XML Press), 2013.

Author Contact Information

Rich Maggiani
Communication Consultant
Solari Communication
137 Lost Nation, Suite 14
Essex, Vermont 05452
+1 802.879.9330
+1 802.999.7440
rich.maggiani@solari.net
www.solari.net

Author Biography

Rich is a Fellow and a 20-year member of the Society for Technical Communication. He is a former two-term member of the Board of Directors, and has been the Marketing Committee Chair and Competitions Committee Chair. He is also a co-founder of the defunct Vermont STC chapter.

Rich founded Page Designs Inc in 1988. He was President of the company's former dba PDI Creative, a full-service marketing agency, and is now the President of Solari Communication. He teaches and speaks throughout the country on communication and business topics. This is his twenty-first presentation at the international STC Summit.



A PRACTITIONER'S GUIDE TO TEACHING TECHNICAL COMMUNICATION

Michael R. Opsteegh

This presentation explores the variety of academic positions that are available to the well suited practitioner and explains a variety of methods for breaking into teaching technical communication. This presentation examines how a practitioner's skill set relates to teaching, explains the variety of educational settings and academic positions, and offers advice on how to apply and prepare for a teaching position. It doesn't cover the breadth and depth of pedagogy in the tech comm classroom. This presentation draws upon my own experience and informal conversations I've had with a handful of practitioners who teach part-time or have transitioned into teaching full-time. A version of this paper was published in the October 2015 issue of Intercom.

Teaching technical communication can be a fun and rewarding experience. It's an important and worthy endeavor. By teaching tech comm, you not only help advance the profession by training those who wish to enter the field of tech comm, but you help those who wish to enter other fields understand and appreciate the value of tech comm.

While teaching tech comm is a noble cause, it's also a lot of work. It requires many hours outside of the classroom, preparation, and patience. Class planning, office hours, late-night grading, mentoring, and responding to student questions takes a lot of time and isn't for the faint of heart. Weighing the tangible cost of time against the intangible reward of helping students become better writers, why would anyone teach tech comm? How does one begin teaching tech comm? And what are the challenges tech comm practitioners face in the classroom?

Assessing Your Interest

Why do you want to teach tech comm? This is the first question I ask when my tech comm friends approach me for advice. Knowing why someone wants to teach tech comm helps me assess their level of interest and gives me insight into what they know about the academic world. You might be interested in teaching tech comm for many reasons:

- You want to make a career change.
- You need more secure retirement benefits.
- You want to supplement your income.
- You genuinely desire to share your knowledge and experience with students.

Of the practitioners I've spoken to, their interest in teaching traverses some or all of these reasons. Let's consider each of them individually to help assess your interest.

You want to make a career change. While this is certainly a valid reason, you should ask yourself why you're drawn to teaching rather than instructional design, training, project management, or other areas related to tech comm. Are you trying to escape some unpleasantness in your current job, or are you really compelled to teach?

You need more secure retirement benefits. While most full-time and some part-time professor appointments offer excellent retirement benefits, they aren't as plentiful as they once were. Many professors contribute a sizable portion of their paychecks to their pensions and contribute additional funds to a 403(b), similar to a 401(k) in the US, to supplement their retirement incomes.

You want to supplement your income. Teaching part time will generate additional income that you can use to pay bills or stash in your retirement accounts. Teaching

Career and Leadership

part time, or adjuncting, however, doesn't pay a lot for the amount of time required. The national average pay for adjuncts in the US in 2010 was \$2,700 per three-unit course ("A Portrait of Part-Time Faculty Members"). This sounds like a lot of money, but when consider the time spent in and out of the classroom, it's about \$10 per hour (Wessler).

You genuinely desire to share your knowledge and experience with students. This is by far the best reason for considering teaching tech comm, and some might say it's the only reason. That passion for fostering the growth of new writers and editors will help sustain your patience and perseverance through late-night grading sessions, bureaucratic idiosyncrasies, and even apathetic students. Teaching is challenging work, and your tenacity and skills are the key ingredients to your success.

If you're still interested after this sobering assessment, let's discuss the various settings and types of teaching appointments that are available.

Breaking into Teaching

The path you take, depends on where you are in your career, your education level, and your experiences. Depending on these variables, you might want to teach in any of the following settings:

- Universities and colleges
- University extensions and continuing education
- Vocational Schools

Universities and Colleges

Universities and colleges are the traditional academic setting. Technical communication classes are often offered as electives in business, engineering, mathematics, or other degree programs, or they're offered within a dedicated technical communication program. The courses may be offered by English, business, or engineering departments and are taught by tenure-track professors or adjunct faculty, often called lecturers or instructors.

Tenure-track faculty. Tenure-track positions are full-time appointments. They generally require a PhD in the field you want to teach. Tenure-track positions are becoming more and more rare as colleges and universities try to cut costs by relying on lower-paid, expendable adjuncts. With so few tenure-track positions open, you have to be willing to relocate to the states where the jobs are available. Candidates with academic backgrounds are favored, but it

is possible to pick up a PhD later in life and transition into teaching. The tenure track positions may also require that you teach some general education classes and will require you to participate in committees and student advising. Depending upon the school, you may be required to "publish or perish," meaning that you're continually conducting research and writing articles and books to burnish the university's reputation.

Adjunct (contingent) faculty. Adjunct faculty positions generally require a master's degree. Adjuncts are often called contingent faculty, because they're hired on a part-time, as-needed basis. You might want to adjunct if you want to keep your day job and teach a class a semester in the evenings. Adjuncts are paid significantly less than their tenure-track counterparts, but adjuncts aren't obliged to sit on committees, participate in governance, manage programs, or advise students. They're often obliged, however, to hold unpaid office hours. Because the work is contingent, and you have no control over when classes are offered, you may need to work with your employer to accommodate your teaching schedule.

University Extensions and Continuing Education

Some tech comm courses and programs may be offered through a university's extension. These course are designed for nontraditional students who have already started their careers or are looking to start a second career. Classes are sometimes offered for credits, but are often offered for continuing education units (CEUs), which aren't applicable to degree programs. Special courses may be designed for corporate clients and delivered on the work site. Instructors of extension courses are nonsalaried, and are hired on a per class basis. A master's degree is often required but may be substituted with advanced professional attainments. The pay varies greatly by institution.

Vocational Colleges

Vocational schools may offer tech comm or business communication courses. These courses are for vocational degrees, and the credits often don't transfer to other colleges and universities. The level of education and experience required vary greatly by school; some require a bachelor's degree while others require a master's or PhD. The compensation for teaching varies dramatically from school to school, and vocational schools typically don't pay as well as the colleges and universities.

Gaining Experience

Now that you have an idea of the setting in which you'd like to teach and the level of education required to get there, you need to start building professional experiences that demonstrate engagement with the profession, a knowledge of contemporary issues, and the ability to present information to an audience. Many of these suggestions are advisable for all practitioners even if they aren't interested in teaching.

Present at conferences. Developing and presenting engaging and contemporary topics at regional or international conferences is a great way to develop your skills as a researcher and presenter. Don't forget to write the paper for the conference proceedings so you have a published record of your topic. A proceedings paper is an article that covers the information presented at the conference and documents any research you conducted.

Present to STC chapters. You don't have to save all your great presentation ideas for conferences. Presenting to local STC chapters is great way to build your repertoire and sharpen your ability to answer challenging questions extemporaneously and think on your feet when your example doesn't resonate with the audience. Even the best planned presentations can go awry, so it's a good idea to learn how to adapt.

Present to other professional organizations. Flexing your presentation skills outside of the friendly confines of STC will help you learn to adapt to unfamiliar situations, people with varying attitudes toward tech comm, and different levels of familiarity with the topic. Presenting to other groups also helps strengthen your credibility as a presenter.

Contact the local university and offer to guest lecture. This is a great way to introduce yourself to the faculty at your local college or university. You can often find the faculty names in the schedule of classes and their email addresses in the campus directory online. Send the professor an email a few weeks prior to beginning of the semester, when he or she is likely to be putting the syllabus together. Name the course you're offering to speak at (you got the course title from the schedule of classes), list three to five topics you could speak on, and ask if there's any room in the class schedule for you to guest lecture. If you don't receive a response, don't feel bad—professors have a lot to do with few resources. Try again next semester.

Conduct workshops. Conducting a half-day or full-day workshop is the closest many practitioners get to teaching: you have a presentation that must be sustained over several hours, you have activities to reinforce the concepts you're teaching, and you often have a learning assess-

ment at the end. You could conduct workshops for your employer, an STC chapter, or as part of a preconference event.

Publish articles. For a tenure-track position, you'd want to publish articles in a peer-reviewed journal like *Technical Communication*. For adjunct positions, publishing articles in professional magazines like *Intercom* certainly helps bolster your credibility. Like presentations, publishing articles demonstrates a certain level of engagement with the profession, your desire to continue to learn, and your desire to share your knowledge.

If you're presenting to an STC chapter or to another organization, contact the local tech comm faculty and ask them to promote the event to their students. This will not only help give you more exposure, but it would benefit the students and chapter by building those connections.

Although, I placed heavy emphasis on giving presentations, this is not to say that presenting is the same as teaching. They're related in the same way a sprint is related to a triathlon. The presentation is one short part of the longer, sustained, and varied endeavor of teaching.

Projecting Yourself as an Instructor

Landing an academic job is very different than landing any other job, but there are some similarities. Don't forget the basics.

Develop a curriculum vitae. The curriculum vitae (CV), like your résumé, outlines your contact information, educational background, your work experiences, and accomplishments. In addition, your CV should also include details about your areas of interest, grants, honors, awards, publications, presentations, and professional memberships ("Curricula Vitae (CVs) versus Resumes"). While you should distill your résumé down to two pages, the details included in your CV may span several pages. For information on assembling a CV, check out the resources available through the OWL at Purdue.

Start cultivating references. Networking and references are important to every job search, and academic positions are no different. Keep in touch with conference organizers, chapter leaders, and faculty for whom you've guest lectured.

Develop a portfolio of targeted writing samples. Some academic applications require writing samples. Literature professors would include research papers or journal articles. A tech comm professor might also include those items, but it might be more appropriate to include

technical documents that show proficiency in the field. Check with the department chair to find out what types of writing samples the committee would consider.

Finding Teaching Positions

Like tech comm jobs, teaching positions aren't all listed on LinkedIn, CareerBuilder, or Indeed; it takes some searching to find the good ones.

Tenure-track positions. If you're looking for a tenure-track position in the United States, you must be willing to search nationwide. Your best bet is HigherEdJobs (higherjobs.com). HigherEdJobs has been around for several years and is one of the most comprehensive sites for finding faculty positions today. If you're searching in the U.S. or internationally, you can try a relatively new site from the *Chronical of Higher Education*, called Vitae (chroniclevitae.com). Both sites allow you to create a profile and apply for jobs, but Vitae also allows you to browse a database of syllabi.

Adjunct positions. If you're looking for a part-time, adjunct position, you probably want to stay local. While you can try the above resources, you're probably better off checking the human resources pages of your local universities, colleges, and vocational schools. Applications may be accepted all year or during specific times. Deadlines will be given for specific semesters or quarters. Sometimes departments will announce openings in the adjunct pool on the department's web page. Bookmark these pages, and keep checking.

Getting Your Foot in The Door

One of the standard requirements on every faculty application is previous teaching experience. How do you gain experience if no one will hire you without experience? This is an age-old catch-22. Your path to teaching tech comm depends upon where you are in your career and education, your publications and presentations, and how well the faculty know you. Here are some suggestions to help you get started.

Teach classes as a grad student. If you're enrolled as a graduate or PhD student, the best way to gain teaching experience (and reduce your tuition) is to become a teaching associate (TA). TAs teach lower-division classes under the supervision of a faculty member. While the syllabus and course materials are often prescribed to them, TAs lead the class discussion, grade the assignments and tests,

and record final grades. There's limited funding for TAs, so it can be competitive.

Volunteer as a member of a tech comm program's advisory board. This one surprised me, but I know a few people who started teaching this way. Many tech comm programs have advisory boards that are comprised of industry leaders and practitioners who meet with the faculty regularly to explain what they expect students to know when they graduate and describe the current trends in the market. The faculty might explain the limitations of the program or state and institutional requirements to adjust the practitioners' expectations. Through this dialog, the curriculum is regularly revised to address the needs of the market. Also through this dialog, faculty and practitioners connect. When a class comes up that needs an instructor, the faculty may invite a promising, qualified advisory board member to teach it.

Reach out to your alma mater. If you made it through grad school and never had the opportunity to be a TA, reach out to some of your former professors. If you graduated within the last five years and they remember you, they might be able help you figure out how deep the adjunct pool is (whether they have enough classes for their current adjuncts, or whether they have too many).

Apply everywhere. If all else fails, play it by the numbers and apply everywhere you see an open position. This is particularly true if you're looking for a tenure-track position. Tenure-track searches are nation wide.

Apply to one place over and over again. This is a long shot, but it worked for me. If you apply to the adjunct pool every semester for several years, you might catch the department chair when a class suddenly becomes available.

If your teaching experience is a little thin, your associations with faculty members become even more critical. Establish those connections early by inviting faculty to presentations, offering to guest lecture, and volunteering to sit on the advisory board.

Preparing for Classes

Long before you ever set foot in the tech comm classroom, you need to prepare. Reading the literature on tech comm, gathering resources, and planning classes takes time, but it'll help you prepare to demonstrate lessons for interview panels and develop your classes when you when you're hired. The following suggestions will help you get started.

Read journal articles. STC's *Technical Communication*, IEEE's *Transactions on Professional Communication*, and

ATTW's *Technical Communication Quarterly* are all dedicated to technical communication research. Subscriptions to these journals require that you become a member or pay a nominal fee. If you want to comb through past articles, they often reside in databases that are locked behind a paywall. Find out if your local public or university library has a subscription to these databases so you don't have to pull out your credit card each time you want to read an article.

Read tech comm textbooks. Pearson Higher Education, Oxford University Press, and Baywood Publishing Company each offer a wide range of tech comm textbooks. Since textbooks can be expensive, check your public or university library or find used copies online. Review texts to determine which seem to offer the best advice, are the best organized, and the most up to date. When you're evaluating textbooks for courses you're teaching, keep price and availability in mind for the sake of your students.

Gather examples of good and bad tech comm and add notes. You'll need real-life examples to analyze and discuss in class. Collecting and evaluating these pieces takes time, so start early. Add notes or create a spreadsheet to catalog each piece and record your own thoughts so you can remember why you selected them.

Look at course syllabi online. Plug popular course names into your favorite search engine and check out the syllabi that come up. Course requirements may vary drastically from school to school, but you can find such commonalities among them as which textbooks and articles seem to be the most popular, which assignments seem to work, and which topics are the most prevalent.

Develop in-class projects. No one wants to sit through a three-hour lecture. No one can absorb all that information, nor should they sit still that long. In-class activities break up the time, engage the students, and reinforce lessons by giving students the opportunity to apply what they just heard or read. Developing in-class activities takes creativity and thoughtfulness, and your first ideas aren't usually your best. You might get ideas from colleagues or from books and articles.

Take a couple of classes. They could be tech comm classes, but they don't have to be. Preferably, take classes from different professors. What does each professor do well, and what could they improve on? Try to do well in the class, so you have an understanding of the struggles students face. Study the classroom dynamic as much as or more than the subject of the class. As a current student you'll have access to the university library and access to some of those textbooks and journal databases I mentioned earlier (well worth the price of tuition!).

Yes, I just assigned you a lot of homework here. You might try some of these suggestions while you're trying to get your foot in the door, but you'll try all of them at some point throughout your teaching career.

Conclusion

You've probably heard the pejorative expression, "Those who can't do teach," which infers that people pursue academic careers because they lack the skills to become practitioners. This statement couldn't be further from the truth. Teaching requires analytical and critical thinking, organization, and the ability to communicate difficult and abstract concepts. These skills overlap greatly with the skills required of practitioners. However, teaching also requires management, advanced presentation, and interpersonal skills to effectively manage a classroom, adapt to situations, and accommodate a wide range of learning abilities.

Practitioners need to understand that teaching a class is not the same as presenting at a conference. Teaching requires sustained long-term planning and preparation, specific learning objectives, adherence to institutional standards, and fair and detailed grading criteria.

Whether you want to pursue a tenure-track position or adjunct on top of your regular job, your skills as a tech comm practitioner, presenter, and author will lend themselves to gathering information and teaching students the skills they need to be successful in our profession. Your desire to share your knowledge with the next generation of tech comm professionals will compel you to continually revise your lessons, practice patience, and persevere through the difficult task of teaching. Cultivate your associations with tech comm faculty, not only for the benefit of your career, but to foster that symbiotic relationship between academia and the profession for the betterment of technical communication.

Resources

Here are a few of the resources mentioned throughout this article.

Curriculum Vitae

Online Writing Lab (OWL) at Purdue University: <https://owl.english.purdue.edu/owl/resource/641/1>

Career and Leadership

Job Postings

HigherEdJobs: <https://www.higheredjobs.com>

Vitae, Chronical of Higher Education: <https://chroniclevitae.com>

Journals

Technical Communication. Society for Technical Communication (STC): <http://www.stc.org/publications/technical-communication>

Technical Communication Quarterly. Association of Teachers of Technical Writing (ATTW): <http://www.attw.org/publications/tcq>

Transactions on Professional Communication. Institute of Electrical and Electronics Engineers (IEEE): <http://pcs.ieee.org/transactions-of-professional-communication>

Textbooks

Baywood Publishing Company: <https://baywood.com/books/BooksBySeries.asp?series=7>

Oxford University Press: <https://global.oup.com/ushe/disciplines/english/business-and-technical-writing-courses>

Pearson Higher Education: <http://www.pearsonhighered.com/educator/discipline/Technical-Communication/91000068.page>

References

- “Curricula Vitae (CVs) versus Resumes.” University of North Carolina at Chapel Hill. 2014. <<http://writingcenter.unc.edu/handouts/curricula-vitae-cvs-versus-resumes/>>.
- “A Portrait of Part-Time Faculty Members.” Coalition on the Academic Workforce. 14 June 2012. <http://www.academicworkforce.org/CAW_portrait_2012.pdf>.
- Wessler, Seth F. “Your College Professor Could Be On Public Assistance.” NBC News. 6 April 2015. <<http://www.nbcnews.com/feature/in-plain-sight/poverty-u-many-adjunct-professors-food-stamps-n336596>>.

Author Contact Information

Michael R. Opsteegh
Eyefinity
220 Commerce, Suite 200
Irvine, California 92602
USA
+1 949.390.8320
michael.opsteegh@eyefinity.com
michael.opsteegh@csulb.edu

Author Biography

Michael Opsteegh has been a technical writer in the software and financial services industries since 2004. He is currently a senior technical writer for Eyefinity, which supports eyecare professionals with industry-leading software and services, and he is a lecturer in the technical and professional communication program at California State University, Long Beach (Go Beach!). Michael specializes in professional and technical communications that include user guides, websites, policies, procedures, and illustrations, and he adds value through research, insight, and enthusiasm. He holds a master’s degree in English, rhetoric, and composition and a certificate in technical and professional communication. He has been a member of STC’s Orange County chapter since 2002 and is currently the chapter president. He is among the first in the profession to be designated as a Certified Technical Professional Communicator (CPTC).



LESSONS MY CAT TAUGHT ME ABOUT INSTRUCTIONAL DESIGN

Jamye Sagan

Sometimes, technical communicators may be required to design training materials and programs for their subject matter – even if they have never formally studied instructional design. The author will share various anecdotes of her beloved cat Gimli in discussing five lessons she has learned over the years about developing training courses and related materials.

Technical communication and instructional design share many similarities. The Society for Technical Communication defines technical communication as “any form of communication that exhibits one or more of the following characteristics: communicating about technical or specialized topics, communicating by using technology, and providing instructions about how to do something”.

Instructional design—“the entire process of analysis of learning needs and goals and the development of a delivery system to meet those needs” – incorporates the above tenets of technical communication, especially the last point.

Here are five lessons I have learned over the years about developing training programs and materials, all told through the lens of stories about my cat Gimli.

Lesson 1: Keep Your Eye on the Goal

As a kitten, Gimli would do anything to reach the dangling plants on the bookshelf. At first, he would try to leap directly from the floor to the top shelf, but never could reach the plants. Over time, Gimli learned to use the furniture around him. He would first leap up on a chair, then stretch his tiny paws on the shelving unit, and finally hoist himself at his desired destination, where he proceeded to play with the plant.

Just as Gimli kept his eyes on his goal, the plant, so should you when developing training. First of all, identify what your learners need to know. Once you have done so, then

begin writing learning objectives. Learning objectives are simply what you want your learners to know. Make sure these objectives are specific, measurable and tangible – that they can be seen, felt, or heard.

Also, when defining learning objectives, refer to Bloom’s Taxonomy for examples of measurable verbs to use, depending on whether you want your learners to recite drug information from memory, have them use a glucometer to measure blood sugar, or troubleshoot issues with a drug dispensing machine.

Lesson 2: Cats Don’t Like to Dress Up

Most cats do not enjoy wearing costumes; Gimli was no exception. His human companions, though, loved seeing him wear cute hats, so they would risk getting scratched to capture the perfect photo. Taking these photos was a two-person job. One person had the camera ready to shoot, while the other person (usually I) had the unlucky job of grabbing Gimli, placing the hat on his head, and later nursing her wounds with peroxide.

Training development is the same way. In your training, keep things simple. Cover only what your learners need to know. If the information is “nice to know” vs. “need to know”, cut it from the training. You can always provide this information as reference material. For example, when developing a training video on using a glucometer to measure blood sugar levels, I focused solely on using the glucometer. As for information about diabetes (related to, but not the focus of, the training), I posted links and

documents on our intranet site as reference. I mentioned the presence of reference materials in the video, and did not incorporate it into the main training.

Plus, having clear-cut goals and objectives (covered in Lesson 1) will help you keep the training on track since you have already defined what you want your learners to know.

Lesson 3: Take Small Bites and Don't Gulp Down Your Food

When Gimli ate too much food at one time, he coughed it back up a few minutes later. A few minutes later, he would return to his food dish as if nothing happened, and eat more slowly. As a result, he didn't regurgitate his food, which had a chance to nourish his body and give him energy to run around the house while everyone else slept.

If you present too much information to your learners at once, they may get overwhelmed and start tuning out information. A good rule of thumb is 15-20 minutes per module or activity. When developing training, break it up into smaller chunks – also known as chunking – to make it easier to digest. If developing e-learning courses, break up the content into several modules. For instructor-led courses, mix up activities within the given classroom time. For instance, when I helped develop training on how to use the various devices we use for our health screenings, I focused on one piece of equipment at a time. First, I presented a couple of slides about the device, and the disease state associated with it. Then I demonstrated how to use the device. Once the learners saw what to do, I then gave them time to practice with it.

Chunking information goes hand-in-hand with making sure you present only relevant information, covered in Lessons 1 and 2.

Chunking training content also makes course updates much easier; you only have to update the small chunk vs. the entire course.

When chunking your training, consider these points:

- Think of how steps are broken down.
- Think of the topics you are covering. If you are covering multiple topics, group related ones together
- Break down larger objectives into smaller ones.

Lesson 4: Praise, but Don't be Afraid to Use the Squirt Gun

Scratching posts are the ideal objects for cats to scratch their nails, but many prefer to file their claws on the sofa. When Gimli began scratching his claws on the sofa, I immediately squirted him with the water bottle so he would stop doing so. Gimli hated getting wet (unless he wanted to get wet), so he quickly learned to associate scratching the sofa with getting squirted. On the same token, whenever Gimli successfully used the scratching post, I immediately praised him and gave him his favorite special treats. Over time, Gimli stopped scratching the sofa and started scratching exclusively on the scratching post.

In training, provide immediate feedback, to help reinforce desired behavior and reduce unwanted behavior. Sometimes, it's not feasible to offer immediate feedback, such as when presenting a scenario, but offer it as soon as you are able.

Feedback also needs to be specific. Instead of just saying "Good job!" or "Wrong, please try again", give a reason. Explain why.

When I develop e-learning courses, I incorporate several "Test Your Knowledge" questions throughout the course. I base these questions on the defined objectives, and provide . When the learner answered the question, they would immediately know if they answered correctly or not, and more importantly, why.

Meaningful feedback helps the learner know not only what they did correctly, but also what their areas of opportunity are.

Lesson 5: Practice, Practice, Practice!

When Gimli first was diagnosed with diabetes, I had to learn how to inject him with insulin. I had never before used a needle and syringe, and was afraid to hurt Gimli. To get me accustomed to the injection process, the veterinarian had me practice injecting water into an orange, since an orange best simulates skin. Once I became comfortable with pushing the syringe plunger, the veterinarian then showed me how to restrain Gimli so he could receive the insulin shot. It was quite nerve-wracking at first; I would have to wrap Gimli up in a blanket burrito, so he could not move. Over time, we both became comfortable with the process, even to the point where I no longer had to hold Gimli down. Within months, giving insulin to

Gimli became second-nature; I could simply inject him while he was at his food bowl.

Just as the veterinarian had me practice injecting an orange with water before injecting Gimli with insulin, incorporating practice opportunities are vital to effective training programs. They enable users to practice their skills in a safe environment, without harming others or affecting live production environments. When developing practice scenarios or hands-on training, focus on the most important and/or complex topics first.

How can learners practice what they have just learned? In e-learning courses, you can create software simulations or branching scenarios with Captivate. In instructor-led courses, you can use a testing environment with predefined exercises and scenarios, or use software simulations. Depending on what the learning objectives are, you can incorporate hands-on training. For example, in the health screening class described throughout this paper, I built in a large amount of time for hands-on practice during which the students pair up and perform the screenings on each other. To reinforce further practice outside the classroom setting, I helped produce assessment checklists and cheat sheets that learners could refer to, to make sure they followed all steps correctly.

Conclusion

These above five lessons will help you create effective training courses and related materials for your learners. Although Gimli passed away in July 2014 at the age of 12, his memory lives on – not only in the years I have spent with him, but also how various aspects of his life helped me reflect upon my journey as a home-grown instructional designer, which continues today.

Resources

- Crowder, Laura. “Instructional Objectives.” The Online Learning Curve (18 October 2011). <http://www.learninghouse.com/blog/publishing/instructional-objectives>
- Dalto, Jeffrey. “How to Chunk Training Materials.” Convergence Training (31 October 2014). <http://blog.convergencetraining.com/chunk-training-materials>
- Hopper, Keith and Sun, Wei. “You May Be an Instructional Designer.” Intercom (May 2011). <http://intercom.stc.org/2011/05/you-may-be-an-instructional-designer/>

- Kuhlmann, Tom. “How Long Should My E-Learning Course Be?” The Rapid E-Learning Blog (17 March 2015). <http://blogs.articulate.com/rapid-e-learning/how-long-e-learning-course/>
- Malamed, Connie. “Chunking Information for Instructional Design.” The eLearning Coach (unknown date). http://theelearningcoach.com/elearning_design/chunking-information/
- Simmons, Kerri. “Incorporating Real-World Experiences in E-Learning.” Training Industry (21 January 2015). <http://www.trainingindustry.com/e-learning/articles/incorporating-real-world-experiences-in-e-learning.aspx>
- Staff. “Three Tips for Writing Measurable Objectives.” The Online Learning Curve – Best Practices in Online Higher Education (2 November 2011). <http://www.learninghouse.com/blog/publishing/3-tips-for-writing-measurable-objectives#sthash.PNRBdacM.dpuf>
- Stenger, Marianne. “Five Research-Based Tips for Providing Students with Meaningful Feedback.” Edutopia (6 August 2014). <http://www.edutopia.org/blog/tips-providing-students-meaningful-feedback-marianne-stenger>
- Tucker, Christy. “Intrinsic and Instructional Feedback in Learning Scenarios.” Experiencing E-Learning (19 March 2015). <https://christytucker.wordpress.com/2015/03/19/intrinsic-and-instructional-feedback-in-learning-scenarios/>
- Vipond, Sharon. “Research Spotlight: Writing Assessments to Validate the Impact of Learning.” Learning Solutions Magazine (17 February 2016). <http://www.learningsolutionsmag.com/articles/1904/research-spotlight-writing-assessments-to-validate-the-impact-of-learning/pageall>
- Definition of technical communication: <http://www.stc.org/about-stc/the-profession-all-about-technical-communication/defining-tc>
- Definition of instructional design: <http://www.umich.edu/~ed626/define.html>
- Definition of Bloom’s taxonomy: <http://www.nwlink.com/~donclark/hrd/bloom.html>

Author Contact Information

Jamye Sagan
Pharmacy Communication Advisor
H-E-B
646 South Flores Street
San Antonio, TX 78204
+1 210.938.8526

Author Biography

As part of her role as Pharmacy Communication Advisor for H-E-B, Jamye helps design training programs and materials for various projects and initiatives in the pharmacy department, including use of their proprietary prescription dispensing software and health screening programs. She also manages communications between the corporate office and the store pharmacies.

A Senior Member of STC, Jamye volunteers with the Instructional Design and Learning SIG as the Social Media and Surveys Manager. Jamye also belongs to the Policies & Procedures SIG and the Technical Editing SIG. She also volunteers at the Society level, as a member of the Community Affairs Committee and a member of the Community Achievement Award and Pacesetter Award committees.



AN INTROVERT'S JOURNEY TO LEADERSHIP

Ben Woelk

Many of us might agree that Western society lauds extroverted leaders and their accomplishments. However, introverts make great contributions and can be effective leaders too. Many of us are introverts who may feel unsuited or unequipped for leadership, but are not sure how to take that next step to increase influence and improve visibility. Have you wondered how to empower yourself and leverage your innate skills to become a leader? This paper shares my personal insights and my leadership journey from shy team member to recognized leader. I'll also share key steps I took and experiences that have helped me become a successful leader.

I am an Introverted Leader, despite outward appearances. My introversion informs my approach to leadership. I believe that self-understanding and leveraging introverts' strengths have made me a better leader.

What is an Introvert?

NOTE: Please regard this section as a generalization. Individuals will differ.

Extraverts focus on the outer world of people and things. They tend to be active and have a wide breadth of interests. They understand things through experience. They may be reward seekers and desire fame. They are energized by contact and activities undertaken with others.

Introverts, on the other hand, have a rich inner thought life of ideas. They tend to have a depth of interest, preferring specialization to a breadth of knowledge. They may mull over thoughts and concepts, but not express those thoughts verbally or externally. Introverts recharge themselves by withdrawing from the hubbub to places of quiet and solitude.

How Introvert Strengths Apply to Leadership

Think of introverted leaders. Examples include Albert Einstein, Steve Wozniak, and Abraham Lincoln. What

made them good leaders? Those are the characteristics introverts may want to emulate. Einstein is known for his depth and clarity of thought (and his genius). Although Steve Jobs is the most well known leader of Apple, Wozniak was responsible for many of the innovations. Abraham Lincoln was not gregarious, yet was able to think and act strategically and provide leadership during what may have been the most trying times for the United States.

How I've Made it Work for Me

My strengths include my ability to identify gaps, my desire to make a difference, practicing a servant leadership model, and pursuing excellence. I'm also competitive. (That competitiveness can be both a strength and a weakness. I can push myself and others towards goals. However, I have an innate desire to win at whatever I'm engaged in.)

By no means do I consider myself to have "arrived," but I am surprised by how far I've come. Most of my professional growth has come in the last 20 years. I never envisioned myself as a leader and certainly didn't believe that I'd be able to stand in front of a crowd and still be able to speak articulately. Although there are many formative steps I could look back on, building on small successes, reading about and better understanding introversion, understanding how I communicate best, and understand-

Career and Leadership

ing my personality type (MBTI) have probably helped the most.

Growth opportunities include:

- Overcoming conflict avoidance
- Overcoming reticence
- Not over-analyzing
- Harnessing competitiveness

Strategies

An introvert can employ a number of different strategies:

- Network: Stop avoiding events
- Understand how you work best. Get to know yourself. Take a personality assessment.
- Let others know how you process
- Leverage social media
- Communicate in multiple formats
- Recharge!

Resources

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

Kahnweiler, Jennifer B. *The Introverted Leader: Building on Your Quiet Strength* (Oakland, CA: Berrett-Koehler Publishers), 2013.

Keirsey, David. *Please Understand Me II: Temperament, Character, Intelligence* (Delmar, CA: Prometheus Nemesis Book Company), 1998.

Laney, Marti Olsen. *The Introvert Advantage: How to Thrive in an Extrovert World*, (New York, NY: Workman Publishing Company), 2002.

Petrilli, Lisa. *The Introvert's Guide to Success in Business and Leadership* Kindle e-Book (Chicago, IL: C-Level Strategies), 2011.

_____, Jung Typology Test, www.humanmetrics.com. Accessed 4/11/2016.

Author Contact Information

Ben Woelk
ISO Program Manager
Rochester Institute of Technology
151 Lomb Memorial Dr.
ROS A201
+1 585.475.4122
ben.woelk@gmail.com

Author Biography

Ben Woelk
Former Director, Society for Technical Communication; ISO Program Manager; Information Security Office, Rochester Institute of Technology; Security Guru; Introvert; INTJ; CISSP; Author of *Shockproofing Your Use of Social Media: Staying Safe Online* (Kindle).

DESIGN AND TESTING

Usability Testing	35
<i>David Bachmann and Mandy Morgan</i>	
An Extreme User-Centric Approach to Kiosk Writing	39
<i>Yu-kwong Chiu, PhD</i>	
From Fred Flintstone to George Jetson: Creating Tension in Training Increases Adoption	45
<i>Viqui Dill</i>	
Help Begins with the GUI: TC Meets R&D	49
<i>Leah Guren, Fellow</i>	



USABILITY TESTING

David Bachmann and Mandy Morgan

Employing usability testing for your documentation offering is far more than asking people if they can get through a procedure successfully or if they feel the documentation is sufficient. This presentation discusses not only how to implement usability testing standards in-house, how to write different tests, and also how to use the results to enhance your documentation offering in a variety of ways. You'll also learn how to gain buy-in for conducting usability tests and how to gather a pool of usability test candidates. The concepts we present will also help explain the value and the ins and outs of usability testing from gathering internal and external participants and fielding their feedback to writing tests focusing on key areas of the documentation.

Among the most basic measures an organization can take to assure the quality of its services are the creation and execution of tests specifically designed to expose that service's weakest components. Documentation can reasonably be categorized as a service that is eligible for such tests in order to determine whether or not that content is fulfilling its purpose of accommodating its audience's needs. If usability testing on technical documentation isn't already part of an organization's core quality assurance regimen, then the need for testing will become apparent once that organization's writers become interested in exploring avenues for professional self-improvement. In other words, when writers are compelled to ask themselves, "Is my technical documentation still good?" it might be time to create a suite of usability tests.

Rationale

Whether it's via a company requirement or a writer's individual initiative, there is immense value in gaging the utility of a set of instructions. In short, it's constructive to know if they work before someone attempts to use them, especially if the instructions' purpose is complex enough to provide any amount of peril to the individuals performing them. Consider the United States military's need to adopt usability testing in the early 1940's after army recruits suffered injuries or died misusing new technologies on the eve of World War II, when both mechanical and human operational health would become paramount. Such mishaps encouraged efforts to understand how peo-

ple read and then apply knowledge to technologies. Thus, the window was opened for technical communicators to influence technology use, as well as the transfer of knowledge about technologies. In this case, any excuse on behalf of inadequate documentation was discredited.

For those technical writers who are burdened by the knowledge that other peoples' lives depend upon the soundness of their content, the incentives for usability testing, or a simulation thereof, are obvious. For writers of less important content, the impetus may lie in accountability, perfectionism, insecurity, relentless customer service, unwillingness to be satisfied with the status quo, willingness to recognize one's own fallibility and seeing that fallibility as an opportunity to learn how to improve as a writer. Or it may be driven by a lack of trust that the source of one's content, whether it's a product development or client services department, is adequate and is informing the tech writer of every change in the product or every new or different way in which the client is using the existing functionality. In other words, if those who have the capability to eliminate the writer's ignorance don't convey that which will eliminate the writer's ignorance, then the writer remains ignorant and any number of those who will consume the writer's content do as well. This is one iteration of the cycle that produces the need for usability testing.

Adoption

As technical writers for the litigation support software company kCura, some of our own motivation for adopting usability testing was to get closer to the client experience so that we could learn how to predict potential sticking points and avoid them accordingly when publishing our documentation. We did this in part to embrace our role in the Customer Success department of the organization and because we weren't comfortable with the possibility that flawed documentation was an obstacle to our clients being able to use a product for which they paid an exorbitant amount of money. We also did it because we suspected that our documentation was broken. This was evident by the high volume of support tickets we received that contained suggestions for improving the content and navigability of our documentation.

Once we recognized the need, we were able to address the best approach for getting people to quickly and clearly tell us where, how, and why our content failed. The rules of our tests dictated that we should focus on a specific goal. Ask not whether the subject thinks the page they're viewing is, say, easy to read; ask them if the page allows them to achieve the precise procedure it enumerates. Inherent in specificity is smallness, in that the tasks we provide our participants should be small and concise. Not unlike the way in which we expect our documentation to be laid out, we shouldn't expect our test takers to remember a complex or circuitous instruction, especially if they are unfamiliar with our documentation offering. The classic forest-tree analogy is applicable here, in that we'd generally like our users to find a specific tree and not just circumnavigate the forest. Another general rule is that we should build empathy into our tests and come to a genuine understanding of the circumstances under which the user would ever need to access the information we're asking them to access. Then, we wanted to dig even deeper by asking repeatedly where they need to be after they consume the documentation we've provided.

In the case of an installation procedure, for example, we never want them to arrive to a point where the component they're installing presents them with an error message or a dead-end in which an unforeseen screen is awaiting an input that they didn't know they needed to provide. It's easy to empathize with that sense of failure because almost anyone who's ever used a computer has encountered it to some degree and thus know that they don't want to have to encounter them again. One of the most important rules, in that it often determines the test takers' comfort level, and therefore the willingness to embrace the process, is to never test the user, but rather test the content's capacity to guide the user in completing the procedure. Put your

writing on the spot, not the one using it. And make sure to make this clear to them as they start your tests.

Best practices

It's important to understand how easy it is to unwittingly sabotage your own testing initiative by maintaining the wrong intentions and/or asking the wrong people to take your tests. Unfortunately, you might not realize that you've made either of these errors in judgment until you're in the middle of the tests or have already completed them. Having conducted usability tests on our documentation for over two years, we've learned that diversity in experience is crucial when forming a pool of test takers.

Participants

In terms of test takers inside your organization, you may need both expert consumers of documentation and novices in order to get meaningful, actionable results. For example, having known for certain that John X referred to our installation content on a daily basis while on support calls from certain clients, we recognized John X as an obvious candidate to take the tests we created to gage the completeness of the installation content. This was because John X had a clear and potentially automatic understanding of exactly where the content failed in relation to the ease with which the client installed the software. If John X has noticed repeatedly that the client stumbles at step 5 of the procedure because step 5 doesn't provide a thorough enough explanation of what makes a valid input for that part of the installation, then John X, out of rote familiarity, will most likely speak up loudly and clearly regarding step 5 once he himself is being tested on it.

Meanwhile, Jane P has never before had to look at the installation content while taking support calls from clients because the clients she serves aren't in charge of installing the software at their companies. However, it's still worth our while to test her on the installation content in the same way we tested John X because she may be able to notice purely logical flaws in those procedures that he wasn't able to due to his familiarity with that content. For example, John X is so used to providing clients with the DLLs and other files they need to install new versions of the software that he doesn't notice that the documentation fails to state where clients can get those files. During her usability test, Jane P notices this omission immediately and points it out accordingly by stating the obvious, which is something like, "I can't start installing the files because I don't know where or how to get them." This then provides us with a clear and easy action item, courtesy of Jane P:

provide the location at which and/or the process by which clients can get the files they need to install. John X also provided an action item that Jane P wasn't able to, namely a technical insight that step 5 isn't complete because it doesn't tell the user that their installation will fail because their input wasn't in the correct format. In this way, both types of internal participant (novice and expert) provided feedback that will lead to opportunities to improve our content.

Regarding external users, there may be less of an incentive to deliberately test a novice on certain areas of the documentation. This is simply to avoid wasting their valuable time asking them for input on content that they never have to consume, seeing that external users are most often entrusted to be experts on one and only one area of the software on behalf of their company. Thus, the goal in testing external users is usually to leverage their expertise and gain insights on workflows that we didn't previously know were practiced out in the real world and, as with internal users, to tell us when an existing procedure stops making sense or in any way leads them astray. Furthermore, we always want to convey to our external users our rationale for seeking them out as participants, so that they know we are actively seeking to enhance our documentation to further serve them.

Results

When we attempted to put our self-improvement into practice, we had mixed results in every facet of the testing experience. Some test takers were more vocal than others in voicing their misgivings about the content they were being tested on. We quickly learned that the most useful results often came in the form of the harshest criticisms, in that they were far more actionable than a user's observation of, for example, "This could be slightly clearer" or "This might make a little more sense if it came at the end of these steps." When, as observers, we hear such feedback as the following, it provides us with a clear roadmap for improvement: "It makes no sense for the steps for configuring agents to appear before the steps for setting up a server. Nobody would ever do this in the order that you've documented it. You should switch these groups of steps." In other words, they've provided us with clearer instructions than we've provided them. Thus, we've been given something we can use, as opposed to something we have permission to ignore based on its lack of severity or immediacy.

In rare cases, we've witnessed a test taker become so frustrated by what they're looking at that they go off on a rant about it. As with isolated critiques, such candidness pro-

vides all of the observers with the option of taking offense to the criticism or jotting it down as something to leverage in the near future. One such rant resulted in a list of action items that ended up completely changing the organization of a page on our documentation site and making it far more intuitive than it would have been had someone not felt the need to lambaste it.

One of the worst pieces of feedback we can hear is the following: "At this point, I'd probably just call support for help." As an observer, and possibly the writer who created the very content, it's deflating to hear that the user, after only a few minutes of either searching for the procedure you wrote or even using it to achieve a goal, would give up completely on that content and opt to make a phone call, possibly be put on hold, and have a support representative actively perform the steps with you. This is especially discouraging when you consider how theoretically simple technical documentation can be. While this sort of reaction is always dependent upon the disposition of the test taker, it does serve as a reminder that the documentation that you took pains to create is easily forsaken for a phone call.

Conclusion

In this way, we see that the success or failure of a usability test has varying degrees of obviousness. Sometimes a user's assertion that a page of content is "horrible" can serve as a kind of moment of clarity and thus motivate the writer to improve it. Likewise, the observation that a page is "pretty good" can lead to a false sense of security by convincing the writer that there is little room for improvement at the moment, when in actuality the user is neglecting to mention that a sub-procedure is missing entirely from the page. Perhaps, they've done this because that particular test didn't invite them to comment upon something that didn't exist on the page. Perhaps, the user simply didn't have the time or energy to explain that which we'd left out. Perhaps, they merely hinted at the omission and claimed that they would work with their support person to provide details on it. In either case, the content creators aren't leaving that usability test with something on which they can take action, although their feelings may remain intact.

In addition to accruing a list of "quick wins" for improving our content, instituting usability testing encouraged us to adopt a healthy skepticism about our documentation that has helped to keep it current. Finding a flaw in a documented procedure is no longer a cause for lamentation but an opportunity to eliminate that flaw and give the next reader a purer page. And this may be the true value of

Design and Testing

maintaining the impetus behind a usability testing effort, namely that it promotes professional vigilance and prevents the technical writer from becoming complacent with their content or their strategy for drafting it, which in turn helps prevent it from becoming stale in the face of shifting client business needs and subtle changes in the ways in which clients use the products we document.

Author Contact Information

Dave Bachmann
Senior Technical Writer
kCura
231 S. La Salle St
8th Floor
Chicago, IL 60604
+1 312.784.9863

Mandy Morgan
Technical Writer
kCura
231 S. La Salle St
8th Floor
Chicago, IL 60604
+1 312.529.4625

Dave Bachmann Biography

Dave Bachmann has been a technical writer for six years and currently works at kCura, a litigation support software company in Chicago. Having had no formal training in tech writing, he's had to learn on how to document a complex software system on the job and continues to learn every day.

Mandy Morgan Biography

Mandy works as a technical writer for an ediscovery software company in Chicago, Illinois. She's received both a bachelor's and master's degree in technical writing from Missouri State University. Her master's thesis focused on document usability particularly for patients diagnosed with Inflammatory Bowel Disease (IBD). She is also a member of STC Chicago and is currently working on conducting a global audit in-house at kCura.



AN EXTREME USER-CENTRIC APPROACH TO KIOSK WRITING

Yu-kwong Chiu, PhD

More and more businesses use self-service kiosk applications as a cost-cutting means. However, many kiosk applications today were designed without a deep understanding of users. Kiosk users have needs that are more demanding than users in other channels. I'm proposing an extreme writing approach that aims at not only raising the awareness of kiosk users' characteristics, but also demanding highly disciplined writing in the kiosk environment.

More and more businesses use self-service kiosk applications as an effective means of cutting operating costs. Kiosk applications that achieve this benefit must help users complete their transactions successfully and easily. However, many kiosk applications on the market today were designed without a deep understanding of the intended users. We cannot over-emphasize that understanding users is crucial to successful writing. This is especially true in writing for kiosk users, who have needs that are more demanding than users in other channels such as desktop and mobile. Content developers or user interface designers must clearly understand their needs in order to write successfully.

This article covers an extreme writing approach that emphasizes kiosk users' unique information requirements. This extreme approach aims at not only raising the awareness of kiosk users' characteristics, but also demanding highly disciplined writing in the kiosk environment.

Kiosk Users

Kiosk users exhibit characteristics that are much stronger than users in other channels, such as web or mobile, do.

Kiosk Users Are Goal-Oriented

Applications are designed to help users achieve their goals. Kiosk users coming to a kiosk have specific goals in mind to complete in the transaction. For example, users arriving

at a movie rental kiosk would want to rent or return a movie. Or ATM users would want to withdraw funds or deposit checks. This characteristic is quite prominent in kiosk users. While users of other channels (web or mobile users) may also be goal-oriented, they may or may not have specific goals before interacting with the applications. For example, users visiting Amazon or eBay may not have specific goals in mind. However, kiosk users have predetermined goals. If they fail to accomplish them, they would feel greatly disappointed and frustrated. So to develop a successful kiosk application, content developers must identify the specific goals that users want to achieve.

Kiosk Users Are Task-Oriented

As described previously, kiosk users have specific goals to accomplish, so they expect kiosk applications to guide them through by providing simple, clear, easy-to-follow instructions to help them complete their goals. Do not provide extraneous information. If they read in a hurry, they may get confused. They want the kiosk to provide them with simple instructions to complete their tasks. They expect instructional information.

Kiosk Users Are Not Learners

Kiosk users are in a hurry to complete their transactions. They do not have time to learn new tricks, so it is not a good time to teach them how to use the self-service kiosks. Users would rarely use online help if they run into issues.

Design and Testing

They would simply abandon the transaction instead of learning how to correct their mistakes. All they want is “tell me what to do to get what I want.”

Kiosk Users Are Distracted

Kiosks are usually set up in public places, so users do not have the luxury of selecting a quiet location. They may be distracted from the surrounding. For example, a movie rental machine may be placed outside a retail store. Users may be subject to the noise of the traffic or people entering or leaving the store. So it is all the more important to help users to complete their transactions as quickly as possible.

Understanding the characteristics of kiosk users serves as guiding principles for designing effective user interfaces for kiosk applications and providing the best user experience. In addition, all these characteristics point to a writing approach that uses simple and direct style. The information on the screens must be extremely clear and concise. Instructions must be easy to understand and follow. Make it simple. Remember that in most cases, users are pressed for time. Many do not read the text carefully. If they do, they simply scan the text rather than reading it.

Kiosk Writing Techniques

The following writing guidelines help information developers create user-friendly content for kiosk users.

Don't Make Me Think

The message on the screen must be absolutely clear and unambiguous, so that users can follow the instructions immediately without pausing and pondering what to do

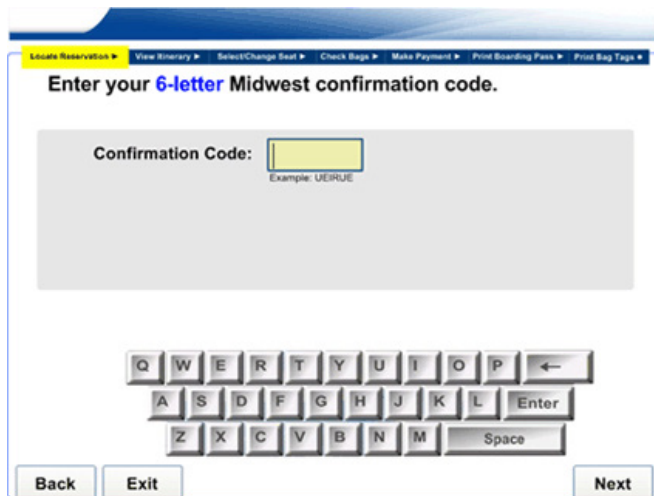


Figure 1. Instructional text that needs to be more specific.

next. In a kiosk environment, if users have doubts of what to do to start or continue, they will most likely abandon the transaction.

Be as specific as possible so there will be no questions asked. Users can interpret specific information more quickly than general information. In kiosk applications, content developers must strive to provide information as specific as possible so that users can act on it immediately. The screen pictured in Figure 1 is used in a self-service check-in application at an airport.

Users are asked to enter their reservation confirmation number. This number may not be clear or familiar to some passengers who do not travel frequently. The original text was “Enter your confirmation number.” To make it more specific, we added “6-letter” and the airline name (Midwest) to tell the user that the code has six letters and is given by the airline (not their travel agent). In addition, visually we set the width of the input field to about six characters long in case users may find a code longer than six characters in their email communication. These changes help them locate the code more quickly and accurately

Content developers may be tempted to add a Help button on the screen to help users find the confirmation code. In a kiosk application, however, adding a Help button is least helpful. During usability testing on this application, the users didn't select the Help button even though they were not sure about the requested information.

Don't think it's clear to you, it's clear to the user. The screen in Figure 2 is also used in an airport check-in kiosk application. Users are asked if they have any hazardous materials in their bags. The screen provides details on what hazardous items they cannot carry in their bags. Users select “Yes” or “No” accordingly. However, many

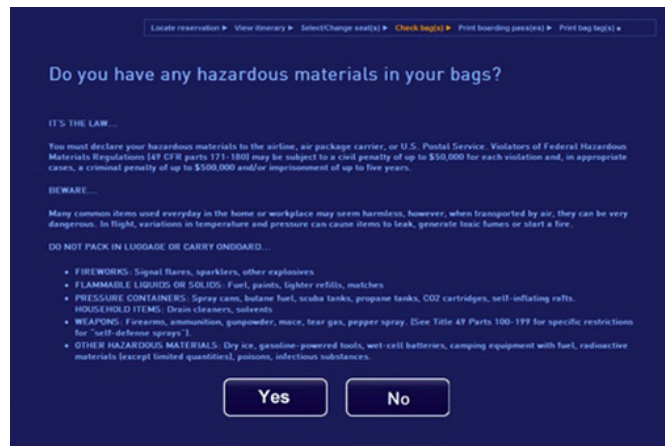


Figure 2. Users select Yes or No to the hazardous materials question; however, a large number of users mistakenly select Yes.

users select the Yes button by mistake even though they do not have any hazardous materials. The reason is that throughout the application users select the Yes option to go to the next screen.

To let users have specific information, we make the button labels more descriptive and clear. We changed the Yes button to “Yes, Cancel Check-In,” and the No button to “No, Continue Check-In”. This change makes the buttons more specific.

To take an “extreme” approach, we replaced the Yes and No buttons with the “Continue” button. Users are informed about what hazardous materials they cannot carry. Then they select Continue to go to the next screen.

Think globally. The world is getting smaller. A travel kiosk application may be used from people all over the world. In Figure 3, the symbol # in “Confirmation #” is used to stand for “number”.

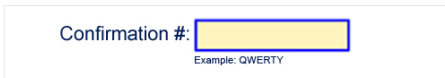


Figure 3. The symbol # is used instead of “number.”

This symbol is commonly used in the United States to mean “number”, but it is not as popular in other parts of the world. Even in this country, this symbol may represent other meanings, such as the hashtag (in Twitter) and pound sign. So if we have to use a symbol, we need to make sure it is a universal symbol, such as temperature degrees Fahrenheit (°F) and degrees Celsius (°C).

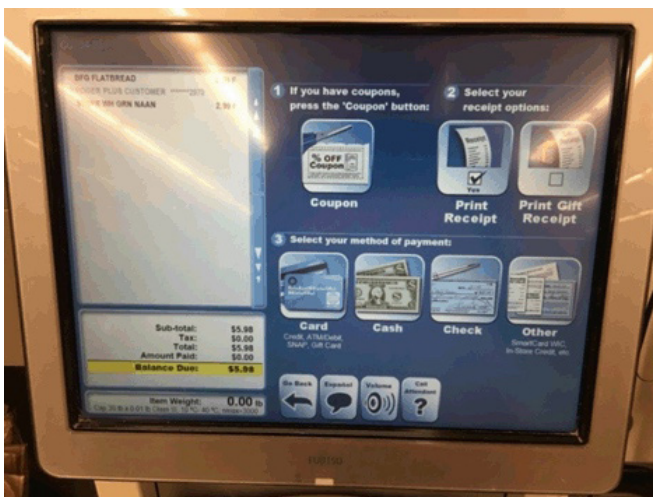


Figure 4. This screen is overly crowded with instructions. Even though the three steps are numbered, there is still too much text for the user to read and understand.

Don’t Make Me Read

The text on the screen must be concise. Most users do not read carefully. Users expect to complete their transactions quickly. For example, users may be at a self-serve check-out machine at a Walmart store. There may be many people lining up to check out, and they may feel pressured to complete their check-out as soon as possible in order not to hold up the line.

Use the least amount of screen text. Figure 4 shows a check-out screen at a grocery store that displays a large amount of information in a kiosk setting. Most users don’t read all the text on the screen. If they do, they simply scan.

One way to make the screen easier to read is to display each of the three steps on a separate screen. This way, users can focus on one step on each screen. Although the number of screens is increased, users will find it easier to use and follow.

Use instructional buttons. As mentioned previously, a kiosk application allows users to perform very specific tasks. We may not need to have introductory information on a screen. Figure 5 shows a screen from a movie rental kiosk that does not display an introductory sentence, such as “Please select one of the following options”. Button labels themselves can serve as instructional text.

Make every word count. The payment screen in Figure 6 is used in a grocery check-out application. The instructional text at the top right “Select your method of payment from the choices shown below” can be reduced to “Select payment method.”

Use simplified sentences. For the sake of brevity, content developers can use simplified sentences. Figure 7 shows an example of a simplified sentence: “Withdrawal request

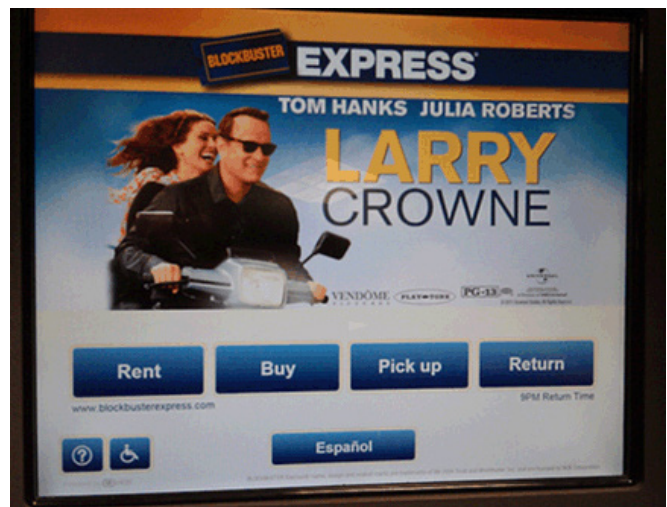


Figure 5. Button labels can be used as instructional text.

Design and Testing

from which account?” It is shorter than using a complete sentence, such as “From which account do you want to make your withdrawal?” Telegraphic style can be used in kiosk and mobile writing.

Be Organized

In some cases, there is a need to display detailed information on the screen. The screen in Figure 8 is used in a self-service application at a post office. The displayed information includes the weight, delivery method, sender's location, scheduled delivery date and time, insurance cost, and postage fee of a packet the user is mailing.

There is quite a lot of information for users to digest. The information can be organized in an easy-to-read manner. One way to help users is to display the information in a simple table, as shown in the bottom right of Figure 8.

Using a tabular format or a bulleted list helps users read and understand the information more easily and quickly.

Let's Go Graphical

Content developers need to try their best to use graphics to support the content. In the hazardous materials example used previously, we can use images to represent the hazardous items. Also, we display a label for each item in case users have trouble understanding the image as shown in Figure 9.

If possible, use only graphics (without text) to convey the message. For instance, to ask users to insert a credit card to make a payment, we can use an animated graphic simulating the action of inserting a credit card into the slot as shown in Figure 10.

Another benefit of using graphics without text is a reduction in translating costs because there is no need to translate any text. The savings would add up quickly if we need to support a score of languages in a global application.

Conclusion

Clear, concise, and organized writing is fundamental to the success of a kiosk application. The audience analysis and writing techniques discussed in this article can be applied to kiosk writing and writing in other channels, such as web, tablet, mobile, or even wearables. Jakob Nielsen remarks that “it's best to be concise when writing for the web. Mobile simply reinforces this point and stretches it to the limit. Short is too long for mobile. Ultra-

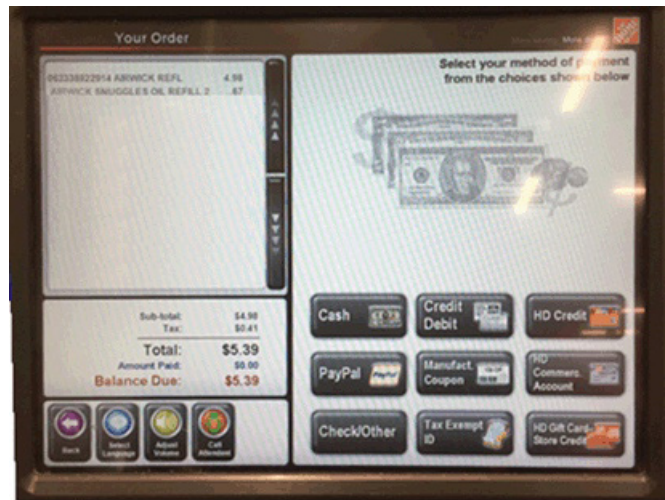


Figure 6. Instructional text “Select your method of payment from the choices shown below” at the top-right can be shortened to “Select payment method.”

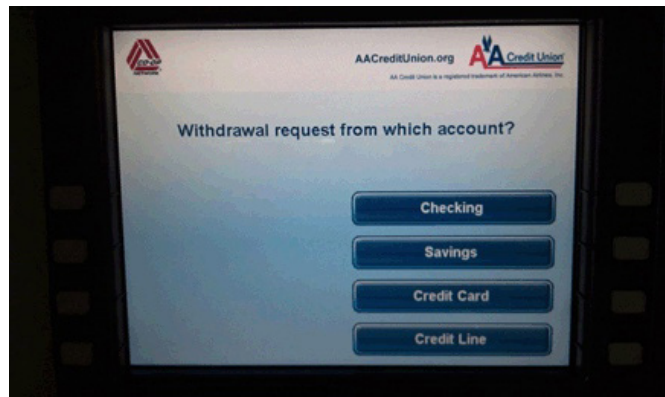


Figure 7. Simplified sentence style is appropriate in kiosk writing. A sentence fragment “Withdrawal request from which account is used instead of a full sentence.



Figure 8. Detailed information can be organized to help users understand faster and better. See the edited text at the bottom right.

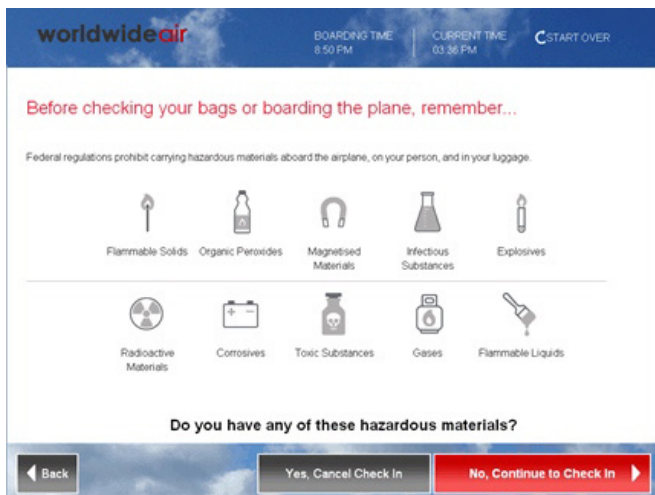


Figure 9. Use images appropriately to help users process the message faster. Each hazardous item is represented by a small icon, which helps users understand the content more quickly.



Figure 10. A good picture can be self-explanatory. This animated image shows users how to insert a credit card into a slot. Text is not needed.

short rules the day.” In kiosk writing, “ultra-short” is not enough. Being “extreme” reigns supreme. Content developers need to ensure that the writing is geared toward kiosk users in their particular challenging circumstances. They need to take a creative and extreme approach to provide the most usable information to meet kiosk users’ highly demanding needs and provide the best user experience possible.

References

- Nielsen, Jakob. “Defer Secondary Content When Writing for Mobile Users.” (1 August 2011). <https://www.nngroup.com/articles/defer-secondary-content-for-mobile/>
- Nielsen, Jakob. “How Users Read on the Web.” (1 October 1997). <https://www.nngroup.com/articles/how-users-read-on-the-web/>
- Urgo, Raymond E. “Telegraphic Style – An Often Unknown, Yet Useful Writing Technique.” Urgo & Associates. http://www.urgoconsulting.com/resources/telegraphic_style.php

- “Wearable Writing Style Guidelines.” Tizen Developer. <https://developer.tizen.org/design/wearable/styles>
- “Write Clearly and Concisely” IEEE. <http://sites.ieee.org/pcs/communication-resources-for-engineers/style/write-clearly-and-concisely/>

Author Contact Information

Yu-kwong Chiu, PhD
 Senior User Experience Designer
 Sabre
 3150 Sabre Drive
 Southlake, TX 76092
 +1 682 605 4707

Author Biography

For more than two decades, Yu-kwong Chiu has been working as a practitioner in the field of technical communication and user experience, specializing in user interface design, information architecture, usability, and HTML/CSS development. He has extensive experience in designing applications in various platforms, including web, mobile, desktop, and self-service kiosk. Currently, he works as a senior user experience designer for Sabre Corporation, a leading innovative technology company in the travel industry. Chiu earned a PhD in technical communication at Oklahoma State University. His dissertation focuses on using visual metaphor in increasing usability of a hypertext system.



FROM FRED FLINTSTONE TO GEORGE JETSON: CREATING TENSION IN TRAINING INCREASES ADOPTION

Viqui Dill

The objective of a good training program is adoption and excellent field execution. This presentation is about how to use a combination of traditional training deliverables and old school psychology to gain user buy-in and achieve a successful launch. We'll talk about how my company uses cartoons and countdowns to ensure that users seek out training and have a stake in adoption and field execution excellence. Whether we create video, user assistance, classroom training, or documentation, what we really want is a group of folks who use the product to do an excellent job with little or no effort and make no mistakes. Creating good training is less about the deliverable and more about building the right relationship.

The Challenge

American Woodmark is a company that manufactures cabinets for kitchens and baths and is headquartered in Winchester, VA. Our company has a number of locations including nine manufacturing plants, fourteen builder service centers, four corporate office buildings and over 200 remote home offices for our sales representatives.

Vision

The project scope was to move the timekeeping software for all employees from one third party software program

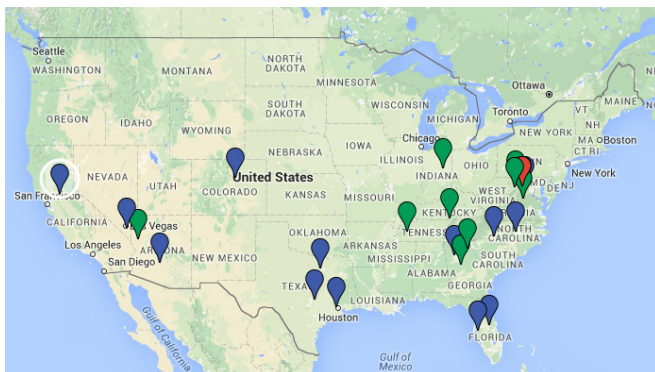


Figure 1. Map of American Woodmark locations

running on premises to a new version on the cloud. The user interface was changing drastically as well as the log in process. Any mistakes made by the employee or the employee's supervisor could potentially show up in the employee's paycheck. Tension on this project was high.

Current Reality

All employees had to move all at once. A phased rollout was not feasible. In addition to upgrading the timekeeping software, all employees needed to install a different version of Java all at the same time. Updating too early would have locked the employee out of the existing system. Updating too late would mean the employee could not use the new system. This restriction on the Java version made it difficult for training to be hands on because demonstration computers were scarce.

No money had been budgeted for travel.

The Team

The first priority was to organize a team of experts. This team would need a wide range of skills. Team members were pulled from Human Resources, Payroll, IT, and from

Design and Testing

each location. Frequent communications ensured that each employee knew the point of contact for the transition.

We organized and trained the team to be trainers and evangelists, equipping them with online help, step by step handouts, a training plan, coaching, and support.

The Process

We used every channel possible for training and communication. Each channel was effective but some were more effective than others.

We rated each channel for effectiveness and engagement. They are listed below from low to high.

1. Online Help (least engaging)
2. How-to Instructions
3. Books for the Training Team
4. Training Materials for the Training Team
5. Person-to-person Training by the Training Team
6. Weekly Email to All Employees
7. Posters (most engaging)

Online Help

The least engaging of the training material turned out to be the online help with only 129 engaged users on launch day.

The online help was robust, searchable, and findable but not as engaging as the other channels.

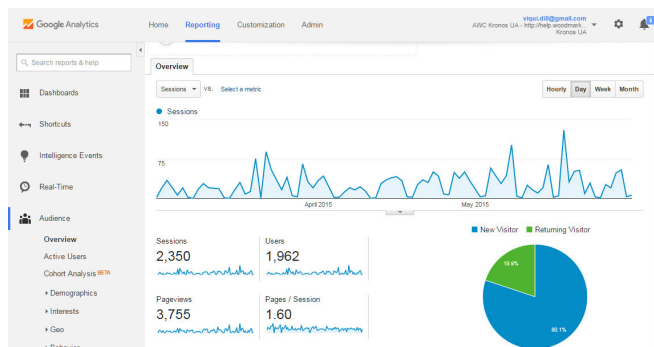


Figure 2. Google analytics on the online help

Google analytics, pictured above, shows that the engagement in the 60 days prior to the rollout was similar to that of the actual rollout period. The data does show a small spike in activity (129 users) on the first day of the launch.

This spike of 129 engaged users is significant but only represents 5% of the total user population.

Our online help is better for just-in-time learning of a specific task than it is for getting-started learning. It contains our own internal processes which were not changing during the transition.

How-to Instructions

Slightly more engaging than the online help was our library of how-to instructions. The library contained a suite of eight different Welcome guides based on the eight different persona profiles for software users. Each how-to guide was designed to be printable and specific to the common tasks of the persona. Topics like “What’s new” and “Pro tips” are highlighted in the guide.

Users read these guides during the training session but did not need to retrieve them at product launch. The guides were not needed because the other training channels had been so effective and the new user interface proved to be easy to understand and intuitive.

Books for the Training Team

We printed and compiled notebooks of training material for each member of the training team. The notebooks contained the how-to guides for each of the eight different persona profiles. We added pencils, pens, sticky notes, and note pads to the package and gave them a colorful cover. Each of the 31 trainers used the material to prepare for the training sessions and had them available as reference during the session.

Training Materials for the Training Team

In addition to the hardcopy training books, the training team was equipped with a training program using PowerPoint and hands-on demo.

The program opened up with an excellent icebreaker quiz about test pilots. Icebreakers get the attendees engaged early and encourage them to participate in the training. The test pilot theme introduced the idea that all employees were going somewhere we had never been before, but that we have the right stuff and will be a success as we take our timekeeping software to the cloud.

The PowerPoint provided structure on the background of the transition, including the what, when, and why.

A hands-on demo of the new software interface gave the attendees a chance to explore the new software and have success in the training environment. Attendees would walk away from the training with confidence that they could use the new software on their own after the launch.

Person-to-person Training by the Training Team

Once the training team was trained and equipped, each trainer scheduled person to person training sessions at each location. The person to person training format gave each employee a point of contact for questions and coaching. We found in our training that employees preferred this kind of relational training and would choose it over more formal methods such as how-to guides or online help.

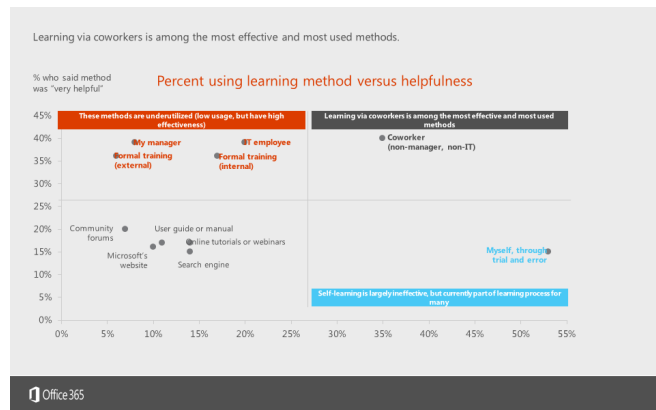


Figure 3. Why are champions important?

Weekly Email to All Employees

The communications team sent weekly emails to all employees to keep them informed of the coming transition. Each email included a countdown of the number of days to launch as well as the most important things to know or to do during the week.

Because we kept the messages short and targeted, employees actually read these emails and kept them as reference. Knowing what to do for the week ensured that employees were not overwhelmed by a long list of coming milestones.

Posters

The most engaging channel of learning material turned out to be the oldest format, colorful printed posters. Each week a new poster was distributed to all employees and posted around each facility. Each poster contained the

daily countdown, a few things to know for the week, and a colorful cartoon of either Flintstones or Jetsons characters.

One engaging place to post turned out to be on the restroom entrance door. Employees would actually stop and read the poster before entering the restroom. And every employee visited that location several times during the week.

After the launch date, the featured cartoon changed from Flintstones to Jetsons, reinforcing the theme that we were going from the Stone Age to the Jet Age with the transition (see Figures 4 and 5 on the following page).

The Results

The team had prepared for a week of troubleshooting and problem solving during the week after the launch. We reserved a conference room to be the central war room for the team. We blocked out the calendars of the IT and HR team for the entire week of the launch.

And nothing happened. There were no problems or issues after the launch. The phone did not ring. All employees had a successful transition to the new software without any issues.

The Party

The last phase of the project was a party to celebrate the successful launch. Project executives presented certificates to each member of the team along with commemorative stress balls shaped like clouds. The CEO of the company joined us that day.

References

- American Woodmark Corporation website <http://americanwoodmark.com/>
- Google Analytics <https://www.google.com/analytics>
- Dill, Viqui, “From Fred Flintstone to George Jetson: Creating Tension in Training Increases Adoption,” Slideshare http://www.slideshare.net/viqui_dill/from-fred-flintstone-to-george-jetson-creating-tension-in-training-increases-adoption
- Microsoft Corporation, “Resource Guide: Change Management, User Training & Communications” <http://fasttrack.office.com/resources>



Figure 4. Weekly announcement poster before the launch

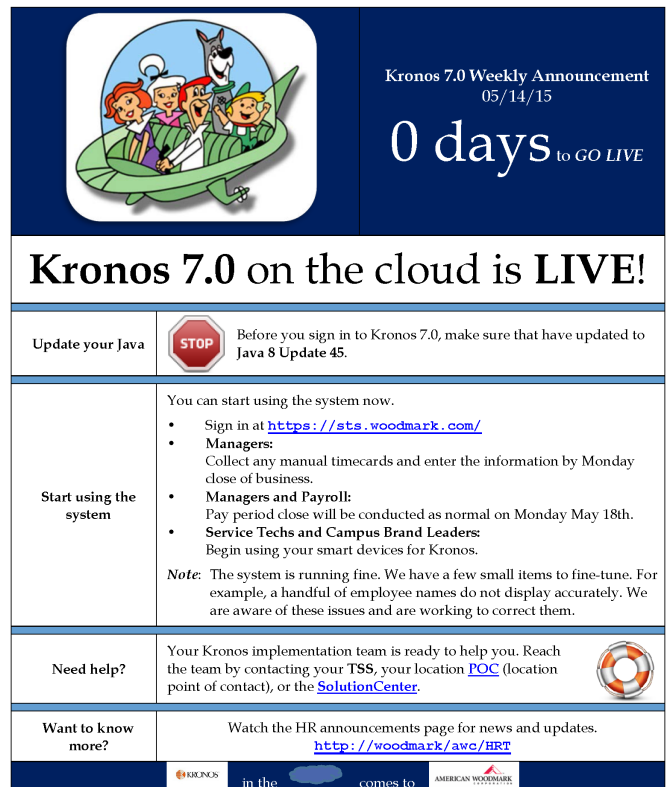


Figure 5. Weekly announcement poster after the launch

Author Contact Information

Viqui Dill
 Technical Communication Leader
 American Woodmark Corporation
 National Technical Center
 131 Dawson Drive
 Winchester, VA 22602
 +1 540.535.7757

Author Biography

As a technical writer for American Woodmark in Winchester, VA, Viqui is passionate about continued education and lifelong learning for those in our field. Sharing the tips and tricks she has gathered in her 18 years as a writer, she is actively promoting networking with other writers to help us all learn and stay connected.

Viqui describes herself as “Technical writer, wife and mom, bass player, worship leader, I’m happiest when folks sing along with me.”



HELP BEGINS WITH THE GUI: TC MEETS R&D

Leah Guren, Fellow

Traditionally, technical communicators start documenting software applications after the first prototype is stable. In some companies, documentation doesn't begin until after the feature freeze. At this point, it becomes extremely difficult to get R&D to make any changes to the user interface. This means that TCs are forced to document screens that are poorly labeled, have confusing text, or even blatant mistakes. This paper explores the ways TCs can get involved earlier in the development cycle and have a greater influence on the usability of UI text.

The Role of the UI in UX

Whether you call it UI (user interface) or GUI (graphical user interface), this is the only side of the product that the user sees. The user doesn't distinguish the dialog boxes and screens from the code behind them. The user considers the documentation to be part of the product (and rightly so!). Even content on the company website, from FAQs to user forums, all become part of the overall UX (user experience). The UX is positive or negative based on how clear the UI is, how the software application responds based on the users' expectations, how well it matches the needs of the user for specific tasks, and the accuracy and helpfulness of information.

Negative UX happens when cognitive friction is present. Alan Cooper describes this as "...the resistance encountered by human intellect when it engages with a complex system of rules that change as the problem permutes." (Cooper) In other words, one dialog box functions differently than another, or one concept is written several different ways.

So what is the problem with products that have high cognitive friction?

- Users experience more frustration.
- There is a higher rate of task failure.
- Sloppy typos and inconsistencies create a subconscious impression that the product is buggy and the company unprofessional.

- Users abandon the official documentation and search for solutions outside the company (such as Google or YouTube searches).
- Ultimately, many users may even abandon the product.

From our perspective, a high-cognitive friction product means that we are left to perfume the pig; that is, write lengthy explanations in the online Help and other documentation to try to explain a bad UI.

Why Does This Happen?

Cognitive friction almost always occurs when the UI is created by developers who are emotionally and intellectually cut off from the users. They are only able to think like developers, and have difficulty understanding why their technical explanations leave users confused and frustrated. Further, developers do not understand linguistic nuances the way we do. Therefore, they often fail to notice vague and ambiguous writing, inconsistencies of style and punctuation, and more. In most cases, developers:

- don't really understand the user or the user goals
- do not understand linguistic subtleties
- do not understand L10N (localization) issues and how they effect the UI

It leads to sloppy, embarrassing, and confusing UI text that we have all seen too many times.

Design and Testing

Since we are the people who have exactly those skills, so why are we not getting involved sooner in the software development cycle? Sadly, in many software companies, we are only asked to proofread the UI for typos (in some companies, the TCs are not even “allowed” to do that). Our goal, therefore, should be to convince the right people that we have much to offer, including richer edits of text (for example, for linguistic consistency and for actual usefulness and clarity), and even usability review of the total project design.

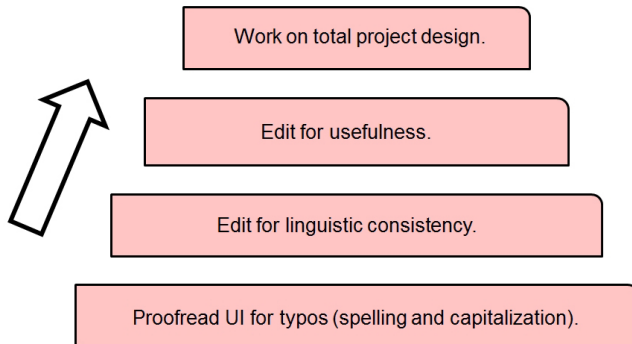


Figure 1. The role of the TC in most software organizations, from low involvement to total involvement

Starting Strategies

If we want to become more involved in the process and stop perfunctory, we need to be competent in several areas:

- Basics of UX: Understanding the users, their motivations, their goals, and how to perform basic usability testing. For details, see Resources, below.
- Basics of L10N: Understanding how to write labels to simplify the translation process and avoid the biggest pitfalls. For details, see Resources, below.
- Rudimentary programming: Be able to read an RC file and understand how to identify text strings.
- UI text standards: Learn the accepted standards for each applicable platform (for example, Windows, Apple, Android, etc.). For details, see UI Text Standards, below.
- Best practices for effective error messages: Learn how to write error messages that users can read, understand, and actually use to solve a problem. For details, see Effective Error Messages, below.

Above all, we need to be willing to step outside of our comfort zone and think of the overall product, not just the documentation.

UI Text Standards

Every platform has its own terminology and guidelines. Some, such as 32-bit and 64-bit Windows environments, became codified, whereas others, such as iOS, have been very fluid and somewhat inconsistent. However, all have style guides stating preferred rules for things such as:

- capitalization
- syntax styles
- punctuation rules
- recommended label length

It is simply a matter of understanding the style guides and using an editor’s attention to detail to spot deviations from the rules.

In addition to these rules, there are conceptual guidelines that apply to a good UI text label. It must be:

- Concise: as short as possible. Avoid unnecessary modifiers. Don’t try to explain everything. Instead, rely on the context when possible. For example, “Emergency Water Pressure Shut-off Valve” is far too wordy; “Emergency Shut-off” is better.
- Clear: provide just enough info to trigger recall. We don’t always know what something is when we see it for the first time (for example, think about the paintbrush icon used for format painting), but once we know what it is, we can successfully recall the function with minimal labeling. In some cases, we may need to add a few words to trigger recall. For example, “Hide errors” may be too vague, but “Hide spelling errors in this document” provides critical detail that is not available within the context of the UI.
- Unique: distinct from other labels. Never use synonyms and avoid similarly-sounding labels. For example, “Design” and “Style” could be interpreted to have the same general function. They are not unique for labels.
- Consistent: the same function, the same name. Use the same menu command and dialog box (screen) title. Be consistent with linguistic structure. For example, if a UI has an icon for adding a picture and one for adding a chart, the labels

“Create Picture” and “New Chart” are not linguistically aligned.

- L10N-compatible: easy to translate. The UI must also be adaptive to language bloat that occurs when translating from English to some other languages (for example, Romance and Germanic languages). If the UI elements are too close together or text fields are fixed size, L10N will be difficult and expensive.
- Enduring: must survive the user learning curve. Users go through three phases: immediacy, learning curve, and permanency. If UI labels are written solely for novice users (the immediacy stage), they will quickly become annoying and burdensome to users. We need to use our Help content to fill the gaps and support the needs of different users throughout their learning curve.

UI Messages

One problem is the blurring of purpose between system message and error messages. System messages:

- should provide information about what is going on internally
- should provide visual feedback or confirmation during a process
- should not be remnants of troubleshooting feedback during the development stage
- should not be lazy design (pushing responsibility onto users instead of the application)

On the other hand, error messages:

- should be corrective issues that can't be handled internally by the application
- should be feedback to let the user know that there is a problem
- should provide immediate information to the user solve a problem
- should not be info that is only helpful or meaningful to SMEs



Figure 2. Cryptic and overly technical UI message

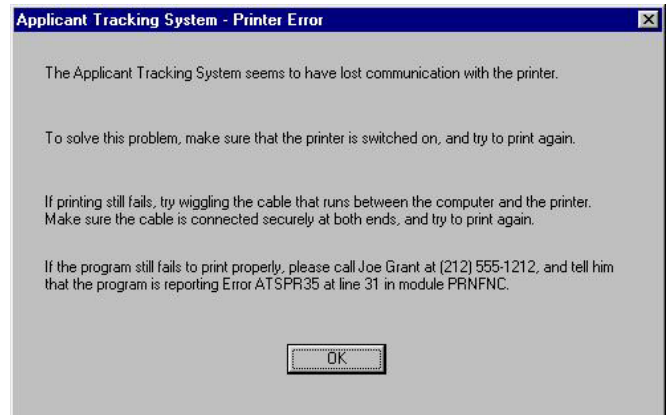


Figure 3. Well written UI message with useful, clean user instructions

Conclusion

So how do we get from the low-level involvement of just checking for typos to full, rich editing and providing usability recommendations?

1. Review your audience analysis data and, if necessary, update your personas.
2. Brush up on your domain and product knowledge.
3. Learn coding basics. You don't need to become a programmer, but take a few tutorials on some of the top high-level programming languages. All share some common concepts. Learn to read an RC file. Make it easier to talk to and work with the SMEs.
4. Learn more about design (see Resources and References, below). In addition, look at existing style guides for your application's platforms (Microsoft, Apple, Sun and other UNIX OSs, etc.).
5. Build internal alliances. Identify the stakeholders and their specific interests and concerns. Then figure out how your skills can address their specific concerns. For example, if mid-level managers

Design and Testing

only care about ROI, don't talk to them about UX. Explain how what you do can improve ROI.

6. Take baby steps. Start by reviewing the prototype UI. Then move on to editing the labels. The work you did in learning code and design basics will pay off and allow you to explain things in a common language with the SMEs. Eventually, get involved in the development of the UI.

Resources

Johnson, Jeff. GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers (Morgan Kaufmann Publishers), 2000.

Hackos, JoAnn T. and Redish, Janice C. User and Task Analysis for Interface Design (Wiley Computer Publishing), 1998.

Kohl, John R. The Global English Style Guide: Writing Clear, Translatable Documentation for a Global Market (SAS Institute Inc.), 2008.

References

Cooper, Alan. The Inmates Are Running the Asylum (Sams Publishing), 2004.

Norman, Donald A. The Design of Everyday Things (Currency Doubleday), 1988.

Author Contact Information

Leah Guren
Owner
Cow TC
leah@cowtc.com
www.cowtc.com

Author Biography

Leah Guren is the owner/operator of Cow TC. She has been active in the field of technical communication since 1980 as a writer, manager, Help author, trainer, usability specialist, and consultant. She now devotes her time to consulting and teaching courses in technical communication, including several of STC's online certificate courses. Her clients include some of the top high-tech and biotech companies internationally, including Intel, IBM, Microsoft, Johnson & Johnson, RAD, Cisco, General Physics, Given Imaging, and Rambam Medical Center. Leah is a published author and an internationally-recognized speaker in the field of technical communication. She is Fellow in STC.

INDEPENDENT CONSULTING

Understanding and Answering a Request for Proposal

55

Brian J. Lindgren

The Client Said Yes: Now What?

59

Teresa S. Stover



UNDERSTANDING AND ANSWERING A REQUEST FOR PROPOSAL

Brian J. Lindgren

Government contractors seeking support for proposal development look for proficiency in English and grammar. While these are critical skills to have, understanding all the nuances of Government Requests for Proposal (RFPs) transcend basic writing skills and require an attentiveness to detail that can make the difference between a great writer and a great proposal developer.

RFPs exist in the commercial world, but with few exceptions RFPs are required for Government agencies seeking contractor support. The typical reaction from a contractor desiring the work would be to determine whether or not the RFP scope is within the organization's capabilities. While this is an important criterion, it is definitely not the only one and at times might be secondary; a big part of deciding whether to respond to an RFP is determining if the likelihood of success. The Government might have a particular contractor or group of contractors in mind. Whether you are an independent consultant or an employee of a contracting company, you will ultimately need to get approval to move forward with the proposal.

Most contractors have some form of gate review process to determine whether to pursue the work. In its simplest terms, a gate review is a business case; as a proposal manager or capture manager, you are trying to convince stakeholders to invest the resources, which in addition to cash could include contracting officers, finance personnel, technical Subject Matter Experts (SMEs) who might need to write content. Very often there are several levels of gate review, particularly for larger proposals. Getting through the gate review process requires persuasion, and there are some indicators that can help determine the chances of success. Even as an independent consultant, be aware of the gate review process; smaller companies might not be as savvy in the pursuit of contracts and helping to ensure their success will exponentially increase your chances of being asked to support future proposals.

Can the Company Win the Work?

Two key data points in a gate review are Pgo and Pwin, where "P" is probability. The Pgo is the probability that the opportunity will come out. If you're working with an actual RFP as opposed to a draft or pre-RFP (more on that later), the Pgo would be 100%. The Pwin is the likely percentage that your company can win. Before determining if you can win, you need to determine if you can bid. Some solicitations are restricted to certain types of businesses, such as a Small Business Set-Aside (SBSA) and at times the restrictions include subsets of small businesses. Generally for Government solicitations the business size is determined by the North American Industry Classification System (NAICS), and depending on which NAICS code applies, business size might either be determined by the number of employees in a company or the company's average revenues over a three-year period.

If the formal RFP is the first time you're hearing about an opportunity, the Pwin is going to be low. If you've been tracking an opportunity pre-solicitation, such as attending an industry day, following advanced notifications, draft or pre-RFP, draft Statement Of Work (SOW)/Performance Work Statement (PWS), the Pwin will increase. Very often organizations will have capture managers, whose role is to track and shape opportunities generally up to 18 months before formal RFP release. In sales terminology the slang "bluebird" is defined as an opportunity (or actual sale) that presents itself to the salesperson or the selling organization without having made much direct effort in securing it. As a rule, the larger a contractor is, the less likely the

Independent Consulting

approval to go after a bluebird, though it can happen if the cost (financial and human resources) is low.

Some things to watch for:

- Will the evaluation be best value or Lowest-Price Technically Acceptable (LPTA)?
LPTA often attracts more bidders and is a challenge because technical excellence is not as critical as the lowest price
- Is this a new opportunity or a re-compete of, or follow-on to, existing work?
- If it is a re-compete, who is the incumbent? Does the Government like the incumbent? Can the incumbent bid? (Sometimes the size standards change; an incumbent small business might “graduate” to large and can no longer bid, or other times the Government decides to reissue the follow-on solicitation as a SBSA and the large business prime can no longer bid.)
- Is the opportunity “wired” for someone – i.e., does it appear the Government is attempting to get to a particular company or individual? Any capabilities cited that only certain companies have?
Are there any personnel or resume requirements that indicate the Government wants to get to a specific individual?
- Is the opportunity backwards-looking or solution-driven? In other words, does it rely heavily on previous work your company has done, or is it more focused on the solution being presented?

Even as a proposal lead rather than a capture lead, try to keep abreast of forthcoming opportunities and review draft RFPs, if available. One of the best resources is Federal Business Opportunities (aka Fed Biz Opps), which presents solicitations, pre-solicitations, and announcements across a wide array of Government agencies. Fed Biz Opps does not require a user ID to browse, but does if you want to subscribe to updates for certain events. Another good website is DefenseLink, which shows all Government/military contracts awarded on a daily basis and includes great synopses of contracts and archives. This requires neither user ID nor subscription. Other search engines, such as GovWin IQ by Deltek, carry a substantial subscription fee.

What to Watch for When the RFP is Released

When the RFP is released, check to see if it is LPTA or best value. Determine if it will be a firm fixed price (FFP) bid

or cost plus fixed fee (CPFF); FFP generally benefits the buyer as it locks the bidder into a price. If you price effectively, FFP can work to your advantage but it definitely carries more risk of underbidding to win the work (and if you bid too high to ensure your costs will be covered, you might lose the bid). Even with best value proposals, pricing is essential to success; though this is often handled by contracts/pricing personnel, the proposal manager should be aware of the proposed cost.

Check Sections L and M. Section L is “Instructions, Conditions, and Notices to Bidders” and Section M is “Evaluation Factors for Award.” Very often a Government procurement will have a broad SOW or PWS to ensure that all its needs will be met, but the requirements for answering the RFP will be streamlined. Speaking of the SOW, if there was a draft RFP for the effort, compare how closely the actual RFP matches it. Substantial content changes could offset the Pwin. Strict compliance to Sections L and M is necessary. Read all details carefully including electronic and hard-copy submission requirements; if the Government is not asking for a Table of Contents or List of Acronyms, do not offer anything extra as it can create new opportunities to introduce errors and be perceived as non-compliance.

When documenting achievements, avoid passive voice; credit for achievements. A remark such as “the network was secured with the latest security updates,” does not convey who actually did the work. Compare that to “Our engineers applied the latest security patches to properly secure the network.”

Ultimately, keep in mind that although technical proposals are firmly within the realm of technical communication, a proposal manager role is basically a project manager; a proposal fits the description of a project (“a temporary effort to create an outcome”). For proposals, editing is arguably more important than writing (particularly when integrating the writing of multiple SMEs and wanting to apply a common voice). When leading the proposal, managing the schedule and costs (not only the bid price, but the use of internal resources to complete the tasking) will determine not only success or failure, but whether you will be brought back to manage future proposals.

Resources

DefenseLink: www.defense.gov. Daily listing of Government contract awards.

Federal Business Opportunities (FedBizOps): <https://www.fbo.gov/>. Searchable clipper service that shows RFPs, pre-solicitations, and allows advanced searches by custom criteria. No cost to subscribe.

GovWin IQ: <https://iq.govwin.com/login/loginpage.cfm>.

A paid service that also offers resources to help companies research specific opportunities.

NAICS organization: <http://www.naics.com/>.

References

CapturePlanning.com article, “Roles People Play in Developing a Proposal.” <http://www.captureplanning.com/articles/11149.cfm>

Sales Dictionary, definition of “Bluebird.” <http://salesdictionary.com/index.php?term=b&tab=%3F>

Author Contact Information

Brian J. Lindgren, PMP
PO Box 43
Sylva, NC 28779
+1 843.696.1190

Author Biography

Brian J. Lindgren currently works for Engility Corporation, a Government contractor, in the dual role of program manager and director of business development. Over his career Brian has been a technical illustrator, technical writer, editor, and web developer. His background includes technical project work at various levels from software testing and documentation to project management to business development and capture management. He considers himself primarily a writer/editor, and the skills associated with these disciplines are critical to technical proposal management and development.

Brian is a Fellow and the current Chair of the Associate Fellows Nominating Committee. He was previously Chair of the International Technical Communications Competitions (now known as the Summit Awards International Competitions) and has been a member of STC since 1991. Brian is also a certified Project Management Professional (PMP) and Secretary of South Carolina’s Charleston Defense Contractors Association (CDCA).



THE CLIENT SAID YES: NOW WHAT?

Teresa S. Stover

Contract work is an ever-growing workforce trend and a great way to work in an atmosphere of flexibility and variety. As a technical communicator, you're well-positioned to be part of this trend. It's one challenge to find the work and get the contract signed. It's a bigger challenge to have the client give you repeat business over time. You start on this road by establishing a good contract with your new client, starting the project on the right foot, working it professionally, and closing it out with grace. See how you can establish an excellent working relationship with your clients and ensure they always call you back for more, and also refer you to others as one of the best in the business.

Congratulations! You've landed a new client and the future is bright. Your first step is to establish a good contract to legalize the relationship, and then work the project with professionalism and success so they'll be calling you back for more.

Set the Stage for Success

As a technical communications contractor or consultant, your degree of success with a particular client can be predicted by the knowledge gathered and preparation done at the proposal stage, even before the contract is signed. Such due diligence includes:

- Knowing the scope of work.
- Knowing the project parameters (audience, deadline, tools, team, technical support, editorial support, graphics support, process).
- Determining whether the scope and parameters match the contract terms and whether the expectations are realistic.
- Executing a signed contract before work begins. The contract should include the scope of work, contract term, requirements, payment, ownership, nondisclosure, independent contractor clauses, and more.
- Taking advantage of your "orientation period" in the first few days (or weeks) of your new contract and asking as many "dumb questions" as possible.

Work Like a Pro

You build a stellar reputation as a professional technical communicator by doing high-quality work, meeting your deadlines, and transforming into a trusted partner who helps meet your client's goals. Key elements to success on the project are:

- Knowing your work, your subject matter, the tools, and the process.
- Adhering to high levels of quality, including accuracy, completeness, standards, and style.
- Fulfilling commitments and meeting your deadlines.
- Balancing the need for strong independent work, being a flexible team player, and being a leader.
- Decoding the most effective communication styles and methods for a variety of team members.
- Developing productive working relationships with team members who have diverse work styles, locations, and viewpoints.
- Keeping the right people informed.
- Anticipating and dealing with problems, and developing workable solutions.
- Discerning when to suggest changes and when to just do it.

Independent Consulting

- Making your client look good, being sensitive to your client's business goals.
- Becoming as indispensable as possible.

Stay On Top of Requirements

Even during the heat and crush of an intense project, be sure to keep up with the business of business. This not only ensures your client's needs are met, but also that you are paid fairly for your work.

Staying on top of requirements means:

- Tracking your tasks and time diligently and accurately.
- Reporting status.
- Billing the client as stipulated in your contract.
- Determining when the project might go beyond the original scope and presenting a contract addendum as necessary.

Close Out Your Contract

Use the short wind-down period as the contract comes to a close to establish continuity in relationships and to market for future projects. While closing out your contract, be sure that you are:

- Tying up loose ends.
- Conducting a final meeting with your client for a debrief, evaluation, and a discussion of future projects.
- Obtaining permission to use items for your portfolio or mentions on your website.
- Asking for referrals or references.
- Distributing thank-you cards and business cards.

Get Repeat Business

After you've moved on to the next projects with other clients, be sure to keep the channels open for future opportunities with previous clients. Do this by:

Staying in touch—the best alternative to cold calls.

Keeping a calendar of when projects from previous clients are scheduled for staffing up.

Following up on the next project.

Repeat business and referrals are typically the only performance review a technical communications contractor or consultant gets.

Repeat business is rewarding because you're developing long-term relationships in a profession that's often characterized by quick-turnaround tight-deadline requirements. With repeat business, your clients are a known quantity and it becomes a pleasure periodically returning to the same companies and the same people. As you gain more and more business from referrals, you can reach the point where you are spending less time marketing and more time on billable work.

With these practices, you can build your brand and reputation as a technical communications consultant or contractor. Your clients will think of you as a bargain at any price as well as a vital asset to the company. What's more, they'll enthusiastically look forward to the next opportunity to work with you.

Resources

Bly, Robert W. *Secrets of a Freelance Writer: How to Make \$100,000 a Year or More* (Montville, NJ: Holt Paperbacks), 2006.

Edwards, Paul, and Edwards, Sarah Anne (1998). *Getting Business to Come to You*. Los Angeles, CA: Tarcher/Penguin.

Author Contact Information

Teresa Stover
Technical Writing Consultant and Book Author
Stover Writing Services
1026 NW Prospect Avenue
Grants Pass, Oregon 97526
+1 541.471.2931
teresa@stoverwriting.com

Author Biography

Teresa Stover, owner of Stover Writing Services, has worked more than 20 years providing excellent technical writing and project management services for software, manufacturing, education, and nonprofit organizations. She is the author of 15 books, the most recent being "Your Project Management Coach" and "Microsoft Project Inside Out." She has conducted many seminars on business, project management, technical writing, and software.

Ms. Stover has a BA degree in English Literature from UC Santa Cruz and a Certificate in Technical Writing from San Jose State. Teresa is an STC Senior Member, having been affiliated with the Puget Sound (Washington), Mid-Valley (Oregon), and Willamette (Oregon) chapters. She's been a member of the Consulting & Independent Contracting SIG since 1990. Learn more at www.stover-writing.com.

STRATEGIES AND TRENDS

Beyond Creating Content: We Can Influence the User's Experience	63
<i>Michelle M. Gardner</i>	
Saving Institutional Knowledge Before Baby Boomers Retire	69
<i>Betsy Kent</i>	
Preparing Legacy Help Projects for Responsive Design	73
<i>Neil Perlin, Fellow</i>	
Providing a Great Search Experience	77
<i>Jeff Stein</i>	



BEYOND CREATING CONTENT: WE CAN INFLUENCE THE USER’S EXPERIENCE

Michelle M. Gardner

You have created a content strategy targeted at specific users and their unique needs. You have helped improve your products through usability testing. So what more can you do to make a difference for your product, team, or business? You can influence the overall user experience of your product. You can use your position and knowledge about the user’s needs to help ensure the success of new features and changes to existing ones. The tools are simple: user profiles, journey maps, usability tests, your knowledge, and your confidence. You can become an advocate for the user.

As technical communicators, we have the basic skills to become user advocates who influence the user’s experience for our products. We think strategically to organize content according to the user’s needs. We ensure that the user has the right information at the right time. We are not invested in the internal functionality of the product or how the code is written. Rather, we view and often use the product from the user’s perspective. To become a user advocate we simply need a good understanding of the methods for improving user experience, as well as the confidence to use our skills and knowledge at the right stages of the design process.

User Experience is More Than Usability

To the newcomer, the concepts of usability and user experience might seem synonymous. However, usability is a component of the overall user experience. Usability expert Jakob Nielsen defines usability as a quality attribute that assesses how easy user interfaces are to use (Nielsen, 2009). To assess usability, we look at task-based interactions: what the user is doing and how the user does the task. We want to reduce the steps in a task or ensure that the tasks can be performed easily and intuitively. User experience focuses on the user’s emotional connection to the task. We ask: Why is the task meaningful for the user? How does the user feel when interacting with the product? If the user performs a required task every day to get a job done and nothing more, then the user likely sees the task as routine

and boring. If that same task is easy to perform, it might have good usability but very poor user experience.

In addition to usability, user experience includes the following elements:

- Branding, which defines the targeted users.
- Functionality, such as the product architecture, design, and interface. This element ensures that the product does what it is designed to do.
- Content, which includes your content strategies, interface labeling, and messaging.

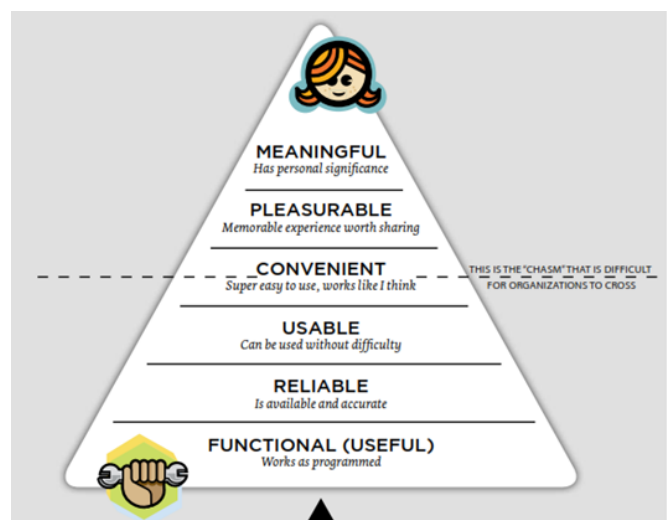


Figure 1. Hierarchy of User Experience Needs (Anderson, 2011)

Figure 1 provides a great example of the user experience elements according to Stephen P. Anderson. At the bottom are the basic requirements that your product should meet. The demarcation line running through “Convenient” shows that this need is both a basic need and an element that can elevate the user experience.

User Experience Requires Knowing Your Users

It seems obvious to state that the most important element in user experience is the user. (Yet, I stated it.) Before the development team starts writing code, we should start thinking and acting on the user’s behalf. Thus, we need a strong understanding of our users. Google states this concept best in their belief “focus on the user and all else will follow” (Ten Things, n.d., para. 2). To start, we gather information about our users and delve beneath the surface of their answers. Then we learn about their needs, pain points, and behaviors. Once we know these facts about our users, we can create user profiles and user personas to help our development team focus on each user.

The best ways to learn about your users are personal interviews and observing users at work. However, content writers often do not have direct access to users. In these cases, we can interview subject matter experts (SMEs) who do have regular contact with users. You can gather feedback from technical support, product managers, field sales, field engineers, and in-house sales staff. Depending on your product, you might want to get demographics about your users, such as age, experience, job tasks, job titles, and goals. During the interview, ask open-ended questions that allow the interviewee to expand on the topic. Schlomo Golz (2014) provides a great list of questions, such as

- How do you do [a certain task]?
- How long does this task typically take?
- Where would you start?
- What would you do next?
- What are the most difficult, challenging, annoying, or frustrating aspects of your job?

You can also review web analytics to determine areas where users might pause, ask for help, or even abandon a difficult task. These types of analytics can help you identify user pain points, which represent those frustrating or difficult aspects of the user’s tasks or the activities for which the user does not have a useful tool. At the same time, question all assumptions: yours, your team’s, and even those of the user. When you receive an answer such as “we

always do it this way,” try asking either of the following questions:

- Why did we choose to solve the problem in this way?
- Could we have done it in a different, better, or more remarkable way?

At the very least you need to know what users do, how they do it, and why they do it.

Once you have learned as much as you can about your users, you can create user profiles and personas. A user profile represents a group of users with common characteristics, goals, and behaviors. It provides the facts and figures about this group. Basically, the user profile explains what the user does and how the user does it. Depending on the complexity of your product, you might have a single profile or several. To ensure you have identified the right user groups, look at each task, feature, and solution that your product provides; then determine which profile applies to that activity. Next, create a user persona for each user profile. 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. The persona breathes life into the facts and figures of the user profile.

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. Empathy enables your team to become that user. Then you can add journey maps and usability testing to help your user advocacy program.

Map The User’s Journey

To create a great user experience, walk in your user’s shoes. A user persona encourages your team to say, “I’m going to log in as Andrew the Administrator and try to complete this task his way.” To help your team in this endeavor, map the user’s journey. A journey map help us understand the user’s experience for a single process or a very complex one. For example, Figure 2 shows the process for purchasing a cup of coffee (Journey, 2013), starting with where to get the coffee and ending with the first sip.

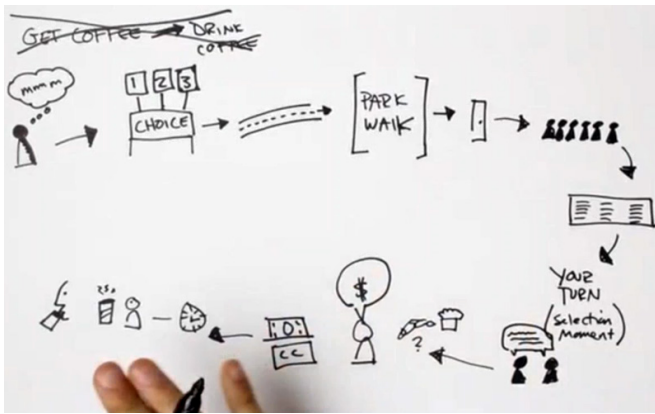


Figure 2. Sample Journey Map

Based on scenarios or tasks, the map allows you to 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.

To create a journey map, select the feature, task, or solution that your product provides for a particular persona. Walk through the activity stage by stage. For each stage, describe the user's actions, thoughts, feelings, and experience. You should identify the outcome or result that the user expects for that stage. You might also want to note whether a stage can be skipped, when and why.

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.

Also, you do not need a special tool or software to create the journey map. You can use 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. Because the journey map describes the user's thoughts, feelings, and expectations, it helps you empathize with that user, which helps improve user experience.

Perform Usability Tests

You can improve the user's experience with usability tests that do not require a large budget or professional usability experts. These simple, do-it-yourself usability tests have a common set of rules (Nielsen, 2012) where test participants:

- Represent a persona
- Perform representative tasks with the design
- Do all the talking

You should perform the tests early in the development process so you can identify the most serious issues before your team has gotten too far along in the code. For testing, you can use wireframes, existing code, or paper prototypes. In *Rocket Surgery Made Easy* (2009), Steve Krug outlines the methodology for performing DIY usability tests. With a little practice anyone can use Krug's method. You need a test moderator, three test participants, and a note taker. In a separate room, place all the stakeholders for the project so they can observe the tests. Each participant performs one or several tasks, but no more than can be performed in about 45 minutes. The moderator encourages the participant to speak aloud to ensure that the observers and note taker capture the user's thoughts and feelings throughout the test. After the third participant completes the test, gather the stakeholders in a room (or conference call) for a debriefing. Identify the main pain points that the participants encountered. As Krug recommends, focus ruthlessly on a small number of the most important problems. You do not need to discuss the methods for resolving the issues or why the issues occurred. Save those discussions for another time rather than allowing the debriefing to get sidetracked.

These DIY usability tests can help you improve the user interface, identify the most serious usability issues early enough to make changes, and begin user-centered design practices. When the stakeholders observe users struggling, the DIY tests help the entire team realize that user experience is everyone's responsibility.

Become a User Advocate

Figure 3 is based on the design process model created by *Effective UI*. At the Empathize stage, you create or update user profiles and personas, share stories about those users,

Strategy and Trends

and encourage your team to think like those users. The Ideate stage allows the team to plan methods for satisfying customer's needs and resolving pain points. This is where you apply what you learned from the journey maps. If you have multiple solutions, perform quick usability tests to see which might be the best course. During the Prototype stage, you can review the user interface and organization of the user activities. Finally, in the Test stage, perform the DIY usability tests.

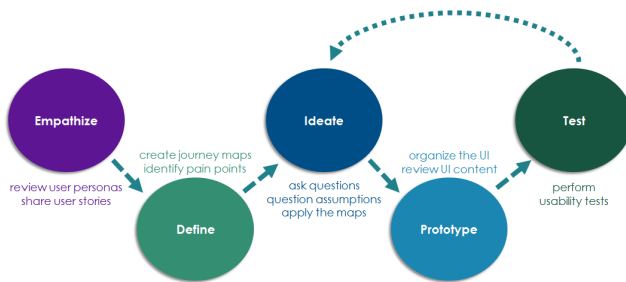


Figure 3. Applying UX Methods to Stages of Development

The model shows that you can influence the user experience at all stages of the design and development process - even when you don't have a formal budget for user experience. Remember, your advocacy for user experience starts with sharing stories about users so the team will view the persona as real person. In doing so, you help your team make the product meaningful and pleasurable to use.

Resources

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/>

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>

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

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

References

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>

Ten Things We Know To Be True. (n.d.). Google Company > What We Believe: <https://www.google.com/about/company/philosophy/>

Ilama, Eeva. "Creating Personas." *UX Booth* (9 June 2015). <http://www.uxbooth.com/articles/creating-personas/>

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>

Journey Map (4 November 2013). Stanford d.school: <https://vimeo.com/78554759>

Nielsen, Jakob. "Usability 101: Introduction to Usability." Nielsen Norman Group (4 January 2012). <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

Krug, Steve. *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems* (New Riders) 2009. <http://www.sensible.com/rsme.html>

Author Contact Information

Michelle M. Gardner
Information Developer
Micro Focus
515 Post Oak Blvd, Suite 1200
Houston, TX 77027
+1 713.418.5180

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 tradeshow. She started as a technical writer at the University of Houston then 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 content to meet those needs.

In 2008, Michelle found her way back to technical writing, but with a twist: As an 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 also mentors colleagues for DIY usability testing and runs a monthly UX Working Group for individuals across the company's product teams to share ideas and methods for improving overall user experience.

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



SAVING INSTITUTIONAL KNOWLEDGE BEFORE BABY BOOMERS RETIRE

Betsy Kent

Today, baby boomers account for 31% of jobs in the US. However, every day, many are reaching full retirement age, and leaving the workplace, taking decades of experience, expertise, and history with them. The same trend is happening in many countries. What issues are going to arise from this departure? And how can we identify and capture the information before the baby boomers depart to finish that novel, plant that garden, or travel to remote parts of the world? This session discusses the potential issues from different points of view (manager, retiree, and coworker) and ideas and best practices on addressing the problem.

In the US, the first Baby Boomers reached full retirement age in 2011. Many of them are leaving the workplace, taking decades of experience and history with them. Today, baby boomers account for 24.3% of the US population. Only 19% of workers over 65 are working. Few employers have analyzed the demographics of their organizations; they don't even realize that they have a pending problem. Similarly, some STC Chapters struggle to recruit new members, especially members who actively work to support Chapter activities. The leadership worries what will happen when older active members stop participating.

But People Leave All the Time, What's the Big Deal?

In a good economy, people change jobs when something interesting comes along. It causes some brief dismay, duties are shuffled around, and with luck, a replacement is hired quickly. Why is this different?

It's the sheer volume of the people retiring. In the United States, roughly 10,000 people a day reach full retirement age, taking with them vast amounts of expertise and knowledge. By 2029, all Baby Boomers will be 65 or older.

For a start, you are:

- Losing a lot of history in a short time

- Losing a lot of accumulated knowledge
- Losing a lot of experience and expertise

Brain Drain Issues

Issues vary depending whether you're a manager, retiring worker, or a colleague.

Issues for management

Brain drain can slow the work process when employees lose that smart person who knows everything. Operations slow while the remaining staff figures out where to find information they need, and the right way to handle it. To

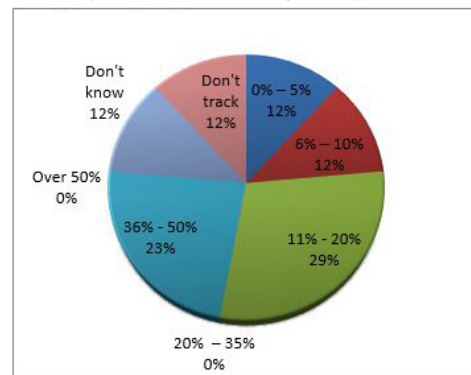


Figure 1. Survey of Human Resource executives on the number of workers over 55

Strategy and Trends

help slow the loss of knowledge, skills, and experience in the U.S. government, Congress and President Obama approved the creation of a phased-retirement program that took effect in November 2014. The program includes a mentorship component to help transfer vital skills and knowledge. However, to date, only one agency has offered the program to its employees.

In a corporate environment, ask Human Resources for a report on which employees are nearing retirement, if such a report is available. Find out what options are available for offering reduced work hours or staged retirement for keeping valued employees.

Consider the following approaches:

- Analyze the prospective retirees and their skills and knowledge.
- Reduce workload of new projects so the employees have time to document what they know and train replacements.
- Enable older workers to work fewer hours while they train and mentor younger workers.
- Encourage the departing staff to talk aloud and explain the reason for doing something. One approach might be to assign junior members of the team to practice interviewing subject matter experts by interviewing senior members of the team about procedures and processes.
- Identify people who enjoy being the only one who knows certain information or processes. Find ways to extract key information from them.
- Ramp up mentoring. This could be traditional mentoring, peer edits, or other approaches.
- Prepare senior and emerging leaders. Many Millennials are interested in becoming leaders, and pick their jobs for leadership opportunities. They are also interested in learning about technologies. More junior members of the team might be looking forward to having more responsibility, or perhaps switching projects. Do some strategic grooming to help them to succeed.

Remember that as older workers are getting nearer retirement, they might have more appointments and commitments outside the offices, for family or health reasons. For example, they might be taking care of parents or grandchildren (or both), or have to deal with illness or a death in the family.

Issues for a Person with an Eye on the Retirement Clock

At some point, even very dedicated people are going to find a countdown clock. And even the best might develop “short-timers disease.” This group needs to consider the good of the profession and the company, and make sure that information and wisdom they have spent a career accumulating is available for others to use.

Remember that your health or family obligations might not cooperate with your long-range plans to work until age 72, or until the mortgage is paid off, or the retirement fund reaches a particular number. And you might just reach a point where you are ready to move on.

Well in advance of your planned departure date, start debriefing yourself. This could take a number of forms:

- Audit the department procedures. This might include migrating the content from an old FrameMaker file to the corporate wiki. Analyze where steps can be simplified or removed. For example, does the procedure include steps that were needed for a long-defunct version of the build software?
- Document historical reasons for the way things are done. In some cases, you might add notes in files using the <draft-comment> element in DITA, or a comparable feature in another technology directly in the source files.
- Update the list of tips and tricks, and again, remove the tips for software that your group no longer uses.
- Check your stash of cheat sheets and copy the substance to the appropriate procedure, wiki, or blog.
- Check emails for useful history or tricks (you’ll be wanting to clear that out anyway)
- Ramp up mentoring. This could be traditional mentoring, peer edits, or other approaches.
- Make notes on communicating with SMEs in general in writing, and advice on specific people verbally. Who’s who information can be critical. Help newer coworkers by explaining who used to be who. It can be helpful to know that Person A has been working on Product C for a long time, but before that, was an expert on Product B.

Issues for Workers with Years to Go

The team members who still have years left on their retirement clocks are probably aware which of their colleagues are counting down the days. This group needs to make opportunities to watch and learn, and position themselves to take on more interesting opportunities.

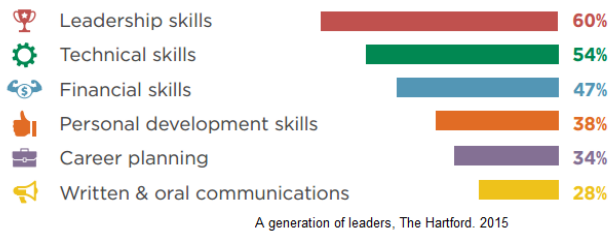


Figure 2. Types of training that Millennials want from their employers

Many of the Millennials may already be making plans to move ahead into leadership roles, as well as preparing themselves for their futures in the field.

As your Boomer colleagues update procedures, help to document the history. Analyze procedures and find out if there are new ways to do the time-consuming tasks.

Observe things that older colleagues do from sheer experience and instinct. Ask how they know the things they know.

Practice interview techniques on older colleagues to capture knowledge. If you are interested in doing video interviews as part of your documentation, experiment with recording your question and answer sessions. They could serve as a proof of concept in a subsequent topic.

Transferring Knowledge

Clearly, capturing information is important. Traditional methods like interviews and audits are a start, but new technologies allow different approaches. Be flexible in your approach.

In 2015, Millennials became the largest demographic in the workforce, outnumbering Gen X and the rapidly-departing Baby Boomers. Training needs to balance the techniques and technology that Boomers are comfortable with and what Gen X and Millennials will be willing to use. Fortunately, senior technical communications practitioners are likely to be comfortable with many of the tools that make sharing information easier.

Blogs and Wikis

Blogs and wikis are natural options for collecting information. These options are appealing to Millennials, and might well be a standard tool for the entire team. The Boomer practitioner might already be creating or editing blogs, so creating a separate training blog is a natural progression. The team can review the accumulated knowledge in smaller, focused chunks, over time, instead of in a rushed brain dump.

Mentoring

According to some sources, Millennials want mentors and prefer Boomers to Gen X colleagues. Determine what type of mentoring relationship will be most productive. Does the younger co-worker want general career help, or specific help with tools?

Consider Video

Video offers many options for capturing knowledge. Some people might make their own videos, which would need to be reviewed for completeness and clarity (and possibly attitude). In other cases, a question and answer interview might make the video more interesting and complete.

Maybe Agile Can Help

Many companies are using Agile in their development process. Teams meet regularly, usually daily, to discuss how the projects are going. Most companies use tracking software to monitor the progress of stories and issues. If stories and tasks are detailed enough, they can provide additional historical support. If the team does not use a template to set up standard routines, such as release processes, encourage them to do so.

For example, the team probably has a routine to get a product out the door. However, when the web site gets overhauled, it's likely that the Documentation person is the one who remembers that the Web team is going to freeze the site content for several weeks. If the freeze coincides with a planned release, the Agile release story can include the necessary details.

Agile also can create some institutional memory. Daily standups include discussions of problems, and the people with longer memories can bring up reasons why things were done in a particular way. Scrum discussion might generate ideas and workarounds to long time problems.

And then There's STC...

All of these issues apply to STC chapters and volunteers. It's always a struggle to inspire new members to volunteer for responsible positions. Encouraging people to stay active and involved is hard, and some of the active members burn out. It's important for senior members to document what they do for the chapter so future volunteers have a baseline to work from.

Conclusion

The departure of Baby Boomers will leave an enormous gap in the workplace. Managers should be worried because specialized knowledge and skills will be walking out the door. Younger workers are looking forward to increased opportunities, though perhaps with some anxiety because their mentors will no longer be around to consult. By working together and planning ahead, we can reduce the impact of the departure.

Resources

- Boulton, Chris. "Millennials force CIOs to rethink tech, training processes." CIO (15 Apr 2016). <http://www.cio.com/article/3056785/cio-role/millennials-force-cios-to-rethink-tech-training-processes.html>.
- Bureau of Labor Statistics. "Monthly Labor Report." Monthly Labor Report. December 2013. <http://www.bls.gov/opub/mlr/2013/article/labor-force-projections-to-2022-the-labor-force-participation-rate-continues-to-fall.htm>
- Challenger, Gray & Christmas, Inc. "Survey: 53% of Employers Concerned About Retirement Plans, Brain Drain." 2013. <https://www.challengergray.com/press/press-releases/survey-53-employers-concerned-about-retirement-plans-brain-drain>.
- Pena, Andrew M. "Institutional Knowledge: When employees leave, what do we lose?" HigherEdJobs. 18 November 2013. <https://www.higheredjobs.com/articles/articleDisplay.cfm?ID=468> <https://www.higheredjobs.com/articles/articleDisplay.cfm?ID=468>
- Psychologistmimi. "Knowing where the bodies are buried: The Psychology of Institutional Memory." Psychologistmimi 03 December 2012.

References

- Lindegreen, Rebecca. "Baby Boomer Brain Drain [Infographic]." MBA@UNC. 28 April 2015. <https://onlinemba.unc.edu/blog/baby-boomer-brain-drain-infographic/>
- Mullins, Ken. "BPM - A Cure for Institutional Memory Loss." BPMInstitute.org 2007 <http://www.bpminstitute.org/resources/articles/bpm-cure-institutional-memory-loss>
- Office of Personnel Management. "Phased Retirement." <https://www.opm.gov/retirement-services/phased-retirement/>.
- The Hartford. "A Generation of Leaders." The Hartford. <http://www.thehartford.com/sites/thehartford/files/millennial-leadership-2015.pdf>.

Author Contact Information

Betsy Kent
Principal Technical Writer
Progress Software Corporation
3005 Carrington Mill Blvd.
Morrisville, NC 27560
+1 919.461.4386

Author Biography

Betsy Kent's first writing job was in medical manufacturing, where she wrote procedures by hand, and had them typed by a secretary. She moved on to writing hardware and software documentation using a variety of computers, and for the last 19 years has been happily writing API documentation. Betsy tells newcomers to the field that being a technical writer is like being a detective. You gather masses of information, sort out what's relevant and important, and arrange that information so the meaning is clear. When not writing API documentation, Betsy looks for someone to take over running the STC Carolina Competition, and writes letters to North Carolina politicians, explaining how they could do their jobs more effectively.



PREPARING LEGACY HELP PROJECTS FOR RESPONSIVE DESIGN

Neil Perlin, Fellow

This paper begins by explaining what responsive design is and, briefly, how it works. The paper then looks at two types of problems that appear when preparing legacy projects for responsive design, legacy problems caused by poor or non-standard coding practices and new problems caused by the nature of responsive design.

Like many companies, yours probably has legacy online help or online manuals. The tools and formats used to create those legacy projects may be outdated but the content itself is still valid, so you'd like to put it into responsive design form because of the increased distribution flexibility. But you have a few questions first.

- What is responsive design?
- How does it work?
- Most important, what do I have to worry about to recast legacy content for responsive design?

Let's take a look at each of these questions.

What Is Responsive Design?

Responsive design means creating one web site or online output that detects the properties of the device on which it's being displayed and automatically reformat itself.

For example, say you've been creating online help to be viewed on desktop monitors only to be told to create output for tablets as well. So you create a tablet-oriented output. Then you're told to create output for phones too. No problem; just add a phone-optimized output. Creating these multiple outputs is easy, but the work can pile up. And you have to maintain them. The solution is responsive design.

The idea of responsive design was developed by Ethan Marcotte in 2010. (The original article is available at <http://alistapart.com/article/responsive-web-design/>. It's interest-

ing to read because it's one of the rare cases where we can identify the birth of a technology.)

Here's an example of responsive design in action from the Children's Museum of Pittsburgh.

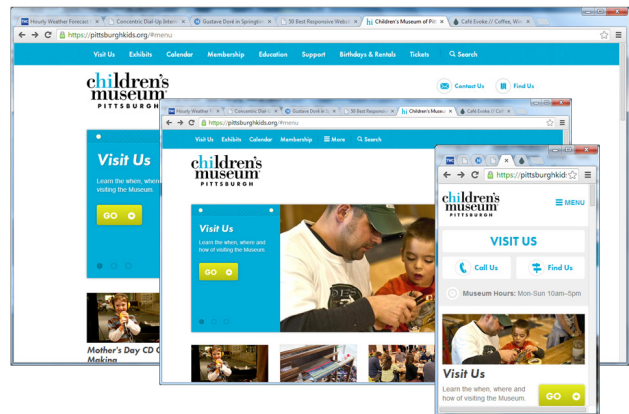


Figure 1. Responsive design in action.

The largest pane is the desktop monitor output. It contains the complete page. The middle pane is the tablet output. It resembles the desktop monitor size but the menu is smaller, with less-important items stored in the three-line flyout icon (sometimes called the "hamburger"), second from the right and labelled More. Finally, the small pane is the smartphone output. It's clearly different, recast to focus on crucial information.

Figure 2 shows another example, a topic in an online help system created using MadCap Flare.

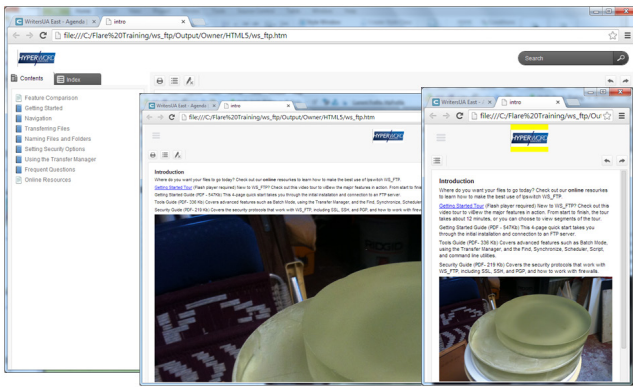


Figure 2. Responsive design from Flare.

The largest pane is the desktop monitor output. It contains the complete page in a typical tripane format. The middle pane is the tablet output. The left-hand navigation pane is replaced by the “hamburger” in the upper left corner, and the “Hyper/Word” icon has moved from the upper left corner to the center. Finally, the small pane is the smart-phone output, with the color bars above and below the icon to indicate a format change.

The crucial point about both examples is that each one is a single output that’s automatically configuring itself for the display device. Why is this important?

It’s important because it lets technical communicators create content to run on any device (within reason) that readers may have. This is the biggest change in technical communication since online help shifted from RTF to HTML in 1997 and started merging technical communication and the web.

But nothing is ever perfect, and responsive design has two speed bumps:

- Converting legacy projects can be messy.
- Converting legacy projects can raise complex new issues.

How Responsive Design Works

Responsive design uses three primary technologies that come out of CSS (Cascading Style Sheets). In brief:

- Relative size units—Uses browser-based “relative” size units like %, em, and rem instead of the more traditional points for resizing flexibility.
- Media queries—Formulas like @media screen and (min-width: 320px) { } test whether the display is a screen (a testable property) and whether its width is 320px or more. The content’s format-

ting changes depending on whether the answer to the query is yes or no. These yes/no points are “breakpoints” that indicate when the formatting should change.

- Fluid grids—Allow content to shift between horizontal and vertical format as the screen size changes. This is a common issue with graphics, as I’ll discuss below.

With that, let’s take a look at the common problems, starting with...

Legacy Issues

Legacy issues are those that are due to bad past development practices or good past development practices that now aren’t supported well or at all.

Local Formatting

Many legacy projects made extensive use of local formatting, where text properties were specified for the text rather than as part of a style. For example, you might see something like:

```
<p class="abc" style="margin-left: 12px; text-align: left;">
```

rather than the cleaner and more up-to-date <p>.

This is not a responsive design issue per se. But it is bad coding practice, inefficient, and overrides the styles in your CSS.

Fixing this requires:

- Replacing the old code with styles and may require editing the CSS.
- Switching from points to one of the relative size units mentioned above.

Image Issues

Images are often sized in inches or pixels and positioned via hidden tables. This causes two problems. The images can’t resize, and the table positioning prevents them from dynamically repositioning themselves as the screen narrows, which produces horizontal scrolling.

Images are also often styled locally and individually. This means that property changes to images must be made one at a time, which is very inefficient.

Fixing this requires:

- Removing the images from tables and positioning them using the `img` “float” property.
- Switching to a relative size unit.
- Setting desired properties for `img` tag, or creating `img` sub-classes to apply to the images.

Table Issues

Fixed table sizes, like fixed image sizes, won’t resize and will force a horizontal scroll bar.

Fixing this requires:

- Using any “auto-size to screen width” option.

As more legacy projects move to responsive design, we’ll probably encounter more technical issues (perhaps the subject of a presentation next year).

New Issues

The new issues are those that are due to new responsive design requirements or options.

More Image Issues

In addition to the issues discussed above, we now have to worry about displaying images on devices that use retina displays. (“Retina display” is Apple’s term for devices whose pixel density is higher than that of “standard” devices. See http://en.wikipedia.org/wiki/Retina_Display.)

How to deal with this? An `img` property called `srcset` lets us include several versions of an image at different resolutions and lets the browser decide which one to use for a given device. (See <http://html5hub.com/srcset-attribute-solving-responsive-image-dilemma/>.)

We also have the opportunity to resize our images but at some point they’ll become too small to read.

How to deal with this? If an image becomes too small to read, we can set the `img` tag’s `display` property to `none` at a breakpoint in order to hide the images, or create a sub-class of `img` and apply it to selected images in order to hide them. However, this may require working in unfamiliar areas of CSS. And bear in mind that hiding an image may affect text that refers to it. Writing “...as shown in the image below” no longer works if the image is hidden.

Do you use image maps, images with multiple hot spots on their surface, in your projects. The hot spots will lose their anchor and float around the image when it’s resized. You can fix this problem by adding hot spots on an image in SVG format, but image maps are an old feature and I’m not aware of image map editors that support SVG format. So you may have to remove your image maps.

Movie Issues

If you’re using “movies” created in Captivate, Camtasia, Mimic, etc., you’ll have to convert them from SWF to HTML5 to let them run on iOS devices. (Your movie authoring tool has to support responsive design or else the movies won’t scale as the screen narrows and you’ll get horizontal scrolling.)

If the movies do scale and contain text caption boxes, make sure the caption boxes scale. Check to see if text in the caption boxes gets too small to read as the boxes scale. If it does, you must replace the text captions with voice-over narration and control which version of the movie gets used, the text caption version or the voice-over version. You’ll also have to increase the time in your project schedule since you’ll be creating two versions of the same movie.

CSS Optimization Issues

The cleaner your CSS, the smaller the file and the faster it loads. CSS files typically use familiar CSS codes but the CSS standard does offer a “shorthand” that can reduce the code. For example, the shorthand lets you combine values for related properties in order to replace this:

```
p {margin-top: 10px;
margin-right: 20px;
margin-bottom: 10px;
margin-left: 20px;}
```

With this:

```
p {margin: 10px 20px 10px 20px;}
```

Google “CSS optimization” for more information.

Excess Content Issues

We create lots of content because we have the screen space for it and have gotten into the habit. But content now has to display on anything from a 22” monitor to a phone, so it’s important to write tightly and edit legacy content. The problem with editing is that at some point, the content

Strategy and Trends

style will change. As an extreme example, consider editing the first line of a masterpiece of English literature, Alfred Noyes' *The Highwayman*, to better fit it for responsive design. We go from:

“The wind was a torrent of darkness among the gusty trees.”

To, with apologies to Alfred: “It was windy.”

Even with more mundane technical content, editing can be difficult and time-consuming.

Summary

Responsive design is part of the future of technical communication, allowing us to efficiently create content for use on almost any device that users may have. But making legacy content work with responsive design may be a complex task that covers the range of project work from content creation to coding and may take us into unfamiliar areas of the CSS standard.

Author Contact Information

Neil Perlin
President
Hyper/Word Services
101 Emily Rd.
Tewksbury, MA 01876
+1 978.657.5464

Author Biography

Neil is president of Hyper/Word Services (www.hyperword.com). He has decades of experience in technical writing, with 31 years in training, consulting, and development for online formats and outputs ranging from WinHelp to HTML5 to mobile apps and tools ranging from RoboHelp and Doc-To-Help to Flare and ViziApps. He has also been working in “mobile” since 1998, CSS since around 1999, and responsive design since it appeared.

Neil is MadCap-certified in Flare and Mimic, Adobe-certified for RoboHelp, and Viziapps-certified for the ViziApps Studio app development platform. He is an STC Fellow, founded and managed the Bleeding Edge stem at the STC summit from 1999 to 2014, and has been a columnist for STC Intercom, IEEE, and various other publications. Contact him at nperlin@nperlin.cnc.net.



PROVIDING A GREAT SEARCH EXPERIENCE

Jeff Stein

Customers rely on search technology to find the information they need in your documentation. Therefore, it's important to keep search in mind as you develop content. Information that you may have created with great attention to clarity and accuracy is useless if people cannot locate it.

Search Features

To get a sense of the current landscape, here is a list of common search features:

- Keyword. The user searches for a single word.
- Phrase. The user enters two or more words in quotation marks.
- Boolean operators. More technical users might want to use operators such as AND, OR, and NOT.
- Wildcards. Users can include a special character, such as an asterisk, to represent zero or more characters. For example, the search query `config*` will match both “configure” and “configuration”.
- Ranked results. The search results appear in order from most relevant to least relevant.
- Filtering. The user is given the opportunity to narrow the scope of the search. Assume that your company has documentation for 10 products. You can let the user restrict the search to a subset of the products.
- Result excerpts. Each search result includes a small piece of text from the topic. This excerpt could be a certain number of initial characters, or an area of the topic in which the search term appears.
- Federated search. Information that a user might be interested in can appear in different types of content: user documentation, knowledge base articles, developer forums, and so on. Federated search lets the user search across all of these types.

Authoring Tools

The search features that you can provide depend a great deal on your authoring tool. The following subsections cover a few of the better-known tools.

MadCap Flare 11

The available search features vary depending on the output type. I looked primarily at the HTML5 output type, which MadCap recommends for online content. Filtering is done with “search filter sets,” a feature that requires you to insert markers into related topics. Synonyms let you provide results for search terms that are similar to words in your content. The search database is divided into “chunk files” for performance reasons. MadCap Pulse is a separate product that shows what users are searching for.

Adobe RoboHelp (2015 Release)

By default, search results include the first 60 characters of each topic. As with MadCap Flare, RoboHelp lets you define synonyms. You can update the stop list file, which contains common words (such as “a” and “the”) that are ignored. With external content search, you can display content from URLs based on terms that you expect users to search. “Dynamic content filters” is not a search feature per se, but it can affect whether content is eligible to appear in search results. A separate product called RoboHelp Server keeps track of searches that occur frequently.

Atlassian Confluence 5.9

Confluence is a wiki-style authoring platform. The search functionality is based on Apache Lucene, a search engine library written in Java. The filtering options are a bit different than that of the previous authoring tools. For example, you can filter by contributor or by when a page was last modified. A convenient help window has examples of searching with Boolean operators, wildcards, and so on.

Case Study: CA Technologies

At CA Technologies, our authoring environment consists of Confluence and various add-ons.

Most of our documentation is public. As a result, it can be searched from Google and other search engines, as well as from the search functionality provided by Confluence.

Our writing standards include best practices for search engine optimization. For example:

- Include keywords in the page titles and the initial 140 characters of a page.
- Avoid one-word page titles, such as “Overview” and “Introduction.”
- Do not start a page with a list of links.

The default page URLs were not search engine-friendly, so we used a Confluence add-on to improve the format. An example of a default page URL is <https://docops.ca.com/display/DTS91/Using+CA+Service+Virtualization>. The new version of this URL is <https://docops.ca.com/devtest-solutions/9-1/en/using/using-ca-service-virtualization>.

Conclusion

After researching and thinking about the current state of documentation search, I offer the following suggestions. Understand and make full use of the search features provided by your authoring tool. Push vendors to fix any problems and add missing functionality. Find out or try to predict the search terms that your customers are using, and ensure that those terms lead to a relevant set of search results. If possible, make some or all of your documentation public so that the content is open to search engines. And be aware that while search technology is a key part of findability, it's not the only part. For example, a documentation set that is poorly organized can make it harder for people to find information.

Resources

- Baker, Mark. “Search is Not Enough: Why We Need Multimodal Navigation.” Every Page is Page One (October 7, 2014). <http://everypageispageone.com/2014/10/07/search-is-not-enough-why-we-need-multimodal-navigation/>.
- Johnson, Tom. “How to Search Engine Optimize (SEO) Your Help Documentation.” I'd Rather Be Writing (September 22, 2013). <http://idratherbewriting.com/2013/09/22/how-to-search-engine-optimize-your-help-content-or-documentation/>.

Author Contact Information

Jeff Stein
Senior Information Services Engineer
CA Technologies
2040 Main St. Suite 400
Irvine, CA 92614
+1 626.372.6910

Author Biography

Jeff Stein is a senior information services engineer at CA Technologies, where he works on the documentation for a product suite called DevTest Solutions. Since becoming a technical writer in the 1990s, Jeff has worked primarily in the software industry. He earned a BA in communication studies from UCLA and an MS in journalism from Northwestern University.

TECHNOLOGY AND DEVELOPMENT

Collaborating and Contributing in GitHub	81
<i>Nicky Bleiel, Fellow</i>	
Editing Tons of Text? RegEx to the Rescue!	85
<i>Eric Cressey</i>	
Using Word to Automate Writers' Processes	89
<i>Robert Delwood</i>	
The Sound of Help—Improve Your Voiceovers	93
<i>Bryna Ann Gleich</i>	
Release Notes Automation	103
<i>Tanya Ivanova and Barry Grenon</i>	
Programming Language Storytime	111
<i>Sarah Kiniry</i>	
REST API Documentation Best Practices	117
<i>Marta Rauch, Fellow</i>	
Getting Creative with Technical Communications	121
<i>Liz Roscovius</i>	
Let The Force of APIs Be with You: Building the Developer Experience Right	125
<i>Nabayan Roy</i>	
The Making of a Learning Platform: Designing Addictive, User-Centric Content	129
<i>Pavi Sandhu</i>	
Power to the People: Targeted Content through Dynamic Filters	135
<i>Dustin Vaughn</i>	
DevOps Docs: Fast, Great Content for the Cloud	139
<i>Vanessa Wilburn</i>	



COLLABORATING AND CONTRIBUTING IN GITHUB

Nicky Bleiel, Fellow

GitHub (<https://github.com/>) is a web-based repository for software projects and is reportedly the world's largest open source community, hosting over 31 million repositories that include both code and the documentation for that code. GitHub includes issue tracking (bugs and feature requests), notifications, diffs, and status dashboards, as well as social features. There are three options for documenting a project in GitHub: Readmes, a wiki, and GitHub Pages. All of these options can be authored in Markdown syntax. Because docs are stored in the same repository as code, they can follow the same workflow, which is especially useful in an Agile environment.

It All Starts with Git

GitHub uses Git as a repository. Git (initially released in 2005) is a distributed version control system for software development that was developed by Linus Torvalds, creator of Linux. According to a survey released in May 2014 by the Eclipse Foundation, Git is the #1 code management tool, with one third of software developers using it. When using Git, documentation can have the same workflow as code, because docs can be stored in the same repository and reviewed along with the code. This visibility is positive in general, but especially useful in an Agile software development environment, where docs are part of the “definition of done.”

Why GitHub?

But this isn't the only reason GitHub has become so popular. The main reason is because it has simplified the process of contributing to open source projects. According to Gregg Pollack of Code School (in the article “What Exactly Is GitHub Anyway?”), before GitHub, contributing to a project meant a developer had to download the source code, make changes, create a patch, then email that patch for consideration. With GitHub, you can create a “fork” of the source code, make your changes, and submit a “pull” request for consideration. And there is a public record of all of this, so you can build a reputation in the community.

GitHub could be considered a “one-stop-development-shop” because it includes issue tracking (bugs and feature requests), notifications, diffs, and status dashboards. It also has social features. You can “follow” other contributors and GitHub will automatically alert you of their activities. You can also “watch” specific projects — you will receive notices about those also. If you just want to tag a repository, but skip the notifications, you can “star” it. You can then go to your “stars page” to catch up on those projects, as well as take a look at the “stars” of your friends.

Organizations can post projects on the public version of GitHub for free, as well as purchase GitHub Enterprise for their internal use behind a firewall. Individuals can create public repositories and contribute to projects with a free account, but must pay a fee to create private repositories that are accessible only to them and their invited collaborators.

GitHub Terminology

Selected GitHub terminology (from the [GitHub Glossary](#)):

- Repository (also referred to as a “repo”): A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history.

Technology and Development

- Fork: A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original.
- Pull request: Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators.
- Merge: Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another.



Figure 1. The Octocat is the official mascot of GitHub. This variation from the *Octodex*, the Ordered Listocat, is by [Cameron McEfee](#).

GitHub Documentation Options

GitHub provides several ways to document a project: Readmes, GitHub Pages, and a Wiki.

Readmes

When you create a project repository in GitHub, a README file is created automatically. It is edited in Markdown, and has the file extension of .md. Markdown is a lightweight markup language that can be converted to HTML easily. If you have used wiki syntax, it will look very familiar. Markdown was originally developed in 2004 by John Gruber, and has splintered into different variations. GitHub uses “GitHub Flavored Markdown” which has extra features like syntax highlighting, task lists, tables, and @mentions (a quick way to notify someone that you need them to take a look at something). GitHub provides a quick guide to mastering Markdown that includes a downloadable quick reference guide. Markdown is also used for GitHub Wikis and GitHub Pages, for comment-

ing, and when creating sharable GitHub reusable snippets, called Gists.

Because the README for a GitHub repository is included with the code, members of the community may suggest changes to the README that you need to review and merge. You can also review the READMEs in other public repositories and suggest changes to them.

Wikis

You can expand upon the information in the README by adding wiki pages to your project repository. Wiki pages are easy to create, just click the Wiki button in your repository. You can edit in Markdown, or choose one of the other eight edit modes. Wiki pages are stored in Git repositories like all other content. By default, anyone can edit your wiki, but you can change the settings and make your wiki read-only. If your wiki is public, you should monitor contributions from the community.

GitHub Pages

Another way to create documentation for your project repository is to use GitHub Pages. These are webpages hosted and published on GitHub. They are authored in Markdown, and you can use GitHub-provided themes to create a custom look. You can even add your Google Analytics tracking ID to each of your Pages. After you are satisfied with your content and the theme, you can publish. By default, the URL of your GitHub pages will be: `http://[accountName].github.io/[repoName]`. GitHub Pages are always public, even if your repository is private. You can also create separate GitHub Pages for you or your organization.

Integrations with Other Tools

Over 70 productivity tools can be integrated with GitHub, including Slack and Travis CI. See the list at <https://github.com/integrations>. GitBook can be used to host and write books (outputs include PDF, ePub, mobi or a website), see <https://github.com/integrations/gitbook>.

Other Uses

GitHub's origins are in software collaboration, but it can also be used to collaborate on other types of projects. According to the article “From Collaborative Coding to Wedding Invitations: GitHub Is Going Mainstream,” it

is also being used to collaborate on projects as diverse as Gregorian chants, licensing agreements, and wedding invitations.

Strategies/DITA North America), and LavaCon; and has been published in STC's Intercom, tcworld magazine, ISTC Communicator, and more. Learn more about her at nickybleiel.com.

Resources and References

- Finley, Klint. "What Exactly Is GitHub Anyway?" TechCrunch (14 July, 2012). <http://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/>
- "Mastering Markdown" GitHub Guides. <https://guides.github.com/features/mastering-markdown/>
- "The Basics of Git and GitHub" GitHub Training and Guides. <https://www.youtube.com/watch?v=U8GBXvdmHT4>
- GitHub Glossary. GitHub Help. <https://help.github.com/articles/github-glossary/>
- GitHub training (try the free 15 minute "Basics of Git" exercise). <https://training.github.com/>
- McMillan, R. "From Collaborative Coding to Wedding Invitations: GitHub Is Going Mainstream" Wired (2 September, 2013). <http://www.wired.com/2013/09/github-for-anything/>
- Bell, Peter and Beer, Brent. Introducing GitHub (Sebastopol, CA: O'Reilly Media), 2014. <http://shop.oreilly.com/product/0636920033059.do>
- Chacon, Scott and Straub, Ben. Pro Git (Apress), 2014. <https://git-scm.com/book/en/v2>
- GitHub Octodex (Original Octocat plus variations). <https://octodex.github.com/>
- Andersson, Arvid. Emoji Cheat Sheet <http://www.emoji-cheat-sheet.com/>

Author Contact Information

Nicky Bleiel
 Watson Information Developer
 IBM
 1710 Murray Avenue, Suite 300
 Pittsburgh, PA 15217
 +1 412.315.2148

Author Biography

Nicky Bleiel is a Watson Information Developer at IBM. She is a Past President and Associate Fellow of the Society for Technical Communication and has over 20 years of experience writing and designing content for software products in a variety of industries. She is a popular speaker at many conferences, including the STC Summit, WritersUA, tcworld, CIDM (Content Management



EDITING TONS OF TEXT? REGEX TO THE RESCUE!

Eric Cressey

Regular expressions can help reduce the time technical communicators spend on repetitive text editing tasks. This paper discusses what regular expressions are, why technical writers should consider using them, and the most common patterns that technical writers should know.

As technical communicators, we're often responsible for repetitive text editing tasks if they're even peripherally related to content we own. As content sets grow, so does scope of these maintenance tasks. Regular expressions are an underutilized but valuable tool technical communicators can use to increase text editing efficiency and reduce the time it takes to complete these tasks.

Regular expressions generally have a bad reputation from the development community. If a technical communicator has heard of regular expressions or seen one, they've likely seen a monstrous, inscrutable regular expression concocted by some developer. Even developers generally tend to be grim when constructing or even reading a regular expression. Given regular expression's dubious reputation, it's not surprising that technical communicators have shied away from exploring regular expressions as a useful tool for their work.

What are Regular Expressions?

Regular expressions are searches with some extra syntax. When you perform a basic search, the search engine tries to find a pattern of characters in a specific order. If I searched for "cat" in a document, the search engine would try to find that exact sequence of characters in the text.

Regular expressions, like basic searches, match text patterns. Basic searches are valid regular expressions. Adding regular expression syntax to a basic search enables it to match more patterns. Instead of only matching a sequence of characters, regular expressions can match based on surrounding content, account for variance, and even match unknown content.

Why Should I Use a Regular Expression?

Repetitive, manual text editing isn't typically valuable, skilled, or enjoyable work. Using regular expressions can automate previously manually tasks, making them faster and easier to do increasing the amount of time technical communicators have for more valuable work. Even if using regular expressions takes as long as the manual approach, the time is more valuable because it was spent developing a skill.

Like any other tool, regular expressions are great at handling specific tasks. If the task involves updating, adding, or removing content based on a pattern, it's worth using a regular expression to simplify that task. If there's not a pattern, regular expressions won't help.

Part of the trick with using regular expressions is knowing what patterns to look for. Developers dislike regular expressions in part because they need a fairly complete knowledge of regular expression syntax and what regular expressions are capable of. As technical communicators, we need a much narrower understanding of regular expression syntax to use them effectively for our most common tasks.

Building a Regular Expression

1. Start with the text you want to edit. Identify a consistent rule about how the content should change (e.g., I want to remove a string ID and its content from multiple files).
2. Identify a common pattern across each occurrence where you need to edit the text. (e.g.,

remove the string ID and everything that follows it to the end of the line).

3. Translate that pattern into a regular expression using regular expression syntax (e.g., `stringID=.*?\n`)

Dealing with Variance

Regular expression syntax includes special characters to match text when there's some known variation in all the occurrences. Matching the variations of a name, for example, is easy to do with regular expressions. When the names differ by one character (e.g., Eric and Erik) use square braces to specify the allowed characters to match in a given position (e.g., `Eri[ck]`).

For longer variations, use parenthesis and a pipe character to insert OR logic. `Eri(c|k)` or even `(Eric|Erik)` both work. A set of parenthesis with characters is called a group.

Groups are also an effective way to deal with optional parts of the match. You can specify the number of times a group is matched by adding a `+` (one or more), `*` (zero or more), or `?` (zero or one) character after the group. These frequency indicators can also follow a single character. It's perfectly fine to have groups inside groups in regular expressions.

Consider many ways URLs appear:

- `www.symantec.com`
- `http://www.symantec.com`
- `https://www.symantec.com`
- `symantec.com`
- `http://symantec.com`
- `https://symantec.com`

The following regular expression matches all of these cases:

```
(https?:\/\/)?(www\.)?symantec\.com
```

Note that forward slashes are special characters in regular expression syntax. When matching characters that are part of regex syntax (periods, forward slashes, question marks, asterisks) escape them with a backslash.

Using Positional Context

Regular expressions can match text based on what appears before and after it. This ability is infrequently used by

developers, but is one of the most useful parts of regular expression syntax for technical communicators.

Suppose each time your full name appears together in some documents your first or last name is misspelled. You could find and replace on either name, but you'd have to manually go through each match to check the positioning. Regular expressions offer a better way, using lookaheads and lookbehinds.

This expression matches a first name only when it occurs before a last name:

```
First(?: Last)
```

`(?:)` is a positive lookahead. Here's how it works:

1. The regex engine finds "First" as usual.
2. It evaluates each character after the match to see if the content matches what's inside the lookahead.
3. If the content is the same, "Eric" is returned as a match. If not, that occurrence of "Eric" is ignored.

There are also negative lookaheads, which behave as expected:

```
First(?:! Last)
```

`(?!)` is a negative lookahead. First is only returned as a match when it doesn't occur before last.

There are also positive and negative lookbehinds. Lookbehinds evaluate the characters before a match.

To find last name when it follows first name:

```
(?<=First )Last
```

`(?<=)` is a positive lookbehind. Here's how it works:

1. The regular expression evaluates each character to see if it follows First.
2. If it does, it continues evaluating the expression, finding Last.
3. Only the match outside the lookbehind is returned.

Negative lookbehinds work as expected:

```
(?<!First )Last
```

`(?<!)` is a negative lookbehind. Last is only returned as a match if it does not follow First.

Finding Unknown Content

Technical communicators working on software projects often interact with strings in name-value pairs. Using lookaheads or lookbehinds is an easy way to modify the name or value of a given pair. If you know the name or string ID, you can get all of its content, even if you don't know what that content is.

```
(?<=name=).*
```

This expression is a positive lookbehind that finds any characters that follow a name. The `.` (period) that follows the lookbehind is a special character that matches any character. The `*` (asterisk) means the regex should match the previous character, the period, as many times as possible.

The asterisk is a greedy match. It returns the biggest most inclusive match it can and, coupled with the period, this may be a bigger match than you intended. Some regular expression engines include whitespace characters with the period match, which means the match could many lines of text.

Consider finding a paragraph with a specific class and all its content in this text:

```
<p class="foo">Hello, world</p><p>This is the second paragraph</p>
```

You might try to solve it with a greedy match:

```
<p class="foo">.*</p>
```

This matches all of the content. Because there's a greedy match, the regex engine keeps evaluating text even after finding the first `</p>`.

It's often necessary to temper greedy matches to return the least inclusive, smallest match possible. Do this by adding a question mark (`?`) after the asterisk, called a lazy match:

```
<p class="foo">.*?</p>
```

With this lazy match, the regex engine stops evaluating the expression as soon as it is satisfied after the first `</p>`. Lazy matches are safer than greedy matches and, as a result, are more frequently useful for typical text editing tasks.

Lazy matches are great for filling in gaps where you don't know exactly what to match. The previous regular expression works for a very specific opening paragraph tag, but what if there are other attributes or classes?

```
<p.*?class=""?foo.*"?.*?>.*?</p>
```

Using lazy matches where other content might appear allows this regular expression to find any paragraph tag that has class "foo," even if there are other attributes or classes present.

Suppose you want to find the last paragraph tag on a page. Using a negative lookahead with greedy and lazy matches, you can.

```
<p>(?!.*<p>).*?</p>
```

Unlike the content inside a lookbehind, which must be fixed-length, lookahead content can be variable-length. This means a lookahead can search the rest of the text to see if something is absent or present. This regular expression finds an opening paragraph tag only when the rest of the text doesn't include another opening paragraph tag. Then it matches to the nearest closing paragraph tag.

Best Practices for Using Regular Expressions

Websites like regex101.com are incredibly useful for writing and testing regular expressions. Similar tools are also available. When you write a regular expression, make sure you use a basic text editor. If you use a Microsoft Office product, it may introduce curly quotes and other smart characters that won't work in your expression.

Work with files in version control. This provides a risk-free learning environment and makes it easy to revert changes and try again. Before submitting changes, diff your changes with the original version and check to make sure the regular expressions worked correctly.

Author Contact Information

Eric Cressey
Senior UX Content Writer
Symantec
350 Ellis Street
Mountain View, CA 94043

Author Biography

Eric Cressey is a Senior UX Content Writer at Symantec. He has been a technical communicator since 2011 and specializes in API technical writing, building custom outputs for web help, UI content for web pages, and task automation.



USING WORD TO AUTOMATE WRITERS' PROCESSES

Robert Delwood

Writers have to create their own tools. Few development groups will allocate expensive programming resources to write tools for documentation groups. The documentation community itself is slow to respond to even high level tools. Products such as Flare, RoboHelp, Framemaker, or DITA compliant tools have made an impact but only in recent years. While those tools are all good at creating documentation suites, the missing cases are the low level tools that help individual writers. Due to wide variations and nonstandard writing practices, comprehensive tools are not likely from the industry. For example, writers who have to document hundreds of error messages by parsing and formatting lists given to them have few commercially provided options.

This expectation of tools is neither unreasonable nor unprecedented. Developers have access to literally thousands of low level tools through a service conveniently provided in Microsoft Visual Studio called NuGet. The reason they have tools is that its very audience, developers, can write their own tools themselves. Documentation teams need to adopt that attitude and learn to create their own tools. However, the perception is that tools are in the realm of programming, and therefore too complex or out of reach for writers. Perhaps some tools are, but we need to redefine what automation is and to understand the capabilities that we have access to. Microsoft Word, maybe the most ubiquitous of tools available, provides a rich capability to incorporate automation and to create tools. It is an underutilized resource and one that needs to be recognized as such. Microsoft aside, many of the following concepts aren't unique to Word, and you can apply these concepts in different ways.

Automation

The goal of automation is simple: Better, faster, cheaper. That is, it aims to save time, to improve quality, and to reduce opportunities for errors. All three tend to happen at the same time. I'll define automation as anything that

saves you time. That makes a broad definition but accurate. For instance, you don't typically consider copy and paste as automation, but one action pastes any number of characters saves time, improves quality, and reduces opportunities for errors. A more obvious case of automation might be for the creation of tables. Some tables get complicated with the addition of header formatting, text, colors, specific fonts, and so on. Recording it as a single macro can include all those features in a repeatable, accurate way. With a little extra work, you can prompt it for the number of rows, columns, and a table name. It could even incorporate highlighted text at the same time, placing it in the correct cells.

Ultimately, automation needs to be recognized as process improvement. First, it engages people more. Programming is seen as a programmer's realm, and until we change that, it'll be a third person solution: They're going to fix it. It needs to be first person solution: I'm going to fix it. And that's part of where process improvement comes from: Individuals trying to solve their own problems. They know their own processes better than anyone else.

Second, it engages more people. Process improvement is not the epiphany of a single person nor is it a manager coming up with an idea. While those may be aspects, it's more about the ideas collectively. It begins with one person identifying the problem. Several others get involved to figure out what the correct solution should be. The manager has to be open to new solutions. Finally, the company must be wise enough to support the time needed to make the change. Clearly, it takes a team to change processes.

To make automation comprehensive, it encompasses three parts:

Automation audit. Reviewing your procedures and habits, looking for potential efficiencies.

Elemental features. Word's built in capabilities. These range from F4 last command, copy and paste, and fields, to Building Blocks.

Programmatic features. Macros and programming resulting in repeatable scripts or Windows applications or browser based tools.

Automation Audit

The point of automation is to solve problems. The most obvious case is when the problem comes looking for you. That is, you have a task, and you need a tool to solve it. Less obvious is when you go looking for a problem. You look for ways to improve your process. An automation audit is the reviewing or the reconsidering of your overall practices with the intent of improving them.

The audit can be formal or informal. Formal is evaluating your procedures in systematic way, looking for repetitive, superfluous, or outdated steps. The audit can also be informal so instead of reviewing established procedures, you become aware of your work habits. Notice, for example, how you accomplish tasks, or which keystrokes you use most often. In both cases, look for repetition as they are the easiest to automate. This can be a series of copy and pastes or making tables, as examples. The repetition of large procedures can be missed, too. Inserting new chapters with formatting, for instance, may only be done once a month or every other month, but still that's six to 12 times a year.

Start your audit by first considering whether a procedure is even necessary. It is surprising how over time practices are accepted as routine or rote, either by an individual's personal habits, or by being documented in a desk guide. Both groups and individuals can get tunnel vision and miss seeing new approaches. In either case, they rarely challenge the procedure for relevancy. Also consider the sequence. Efficiencies can often be found just by moving one into a new position. Performing Step 2 may make more sense before Step 1. Another example is that after the writers finish a section, quality assurance reviews the documents, only to have writers apply updates afterwards, and then a second round of quality assurance. Schedule permitting, wait for the updates, then quality assurance applies only once.

If the procedure is needed, then review the actual steps. Look for repetitions, sequences of steps that always group together, or steps that can be replaced with shortcuts. In a corporate setting, many companies take a top down approach to process improvement, similar to reviewing procedures just mentioned. This does find some but I believe looking at the actual keystrokes writers make, or taking a grass roots approach, yields many more and

deeper improvements. Even if some processes can't be fixed now, keep these in mind as new technologies emerge.

Elemental Features

Elemental features is a broad term used to mean all features built into Word. Word is highly customizable so these compose a considerable set of options. There are over 1,900 commands alone, and all of them can be customized as shortcuts or menu items. Additionally, there are 62 fields, and practically limitless Document Properties, AutoText, and Building Blocks. Almost all of these can be recorded as part of a macro. Together, the power of Word's automation should be clear. But knowing which procedures you want to change is only half the equation. The other half is knowing which commands are available and which ones you can use. Word's features generally fall into three categories: Commands, fields, and Building Blocks.

Commands

Commands are the individual actions you can take. They are the most obvious features and include the ribbon options and keyboard shortcuts. Word is designed to be intuitive and out of the box, so you can use these commands to create a new document, enter text, style it, save and print the document. Some keyboard shortcuts are intuitive as well, such as the four basic ones of copy (Ctrl-C), paste (Ctrl-V), cut (Ctrl-X), and undo (Ctrl-Z). Given Word's sheer versatility, most of the commands are embedded in the ribbon, grouped by function and sometimes to several levels. For instance, mail merge uses up to 70 commands but unless you work with mail merge you may never know those commands exist.

So how do you find out which commands are available, much less ones applicable to you? Unfortunately, there is no easy way. Two obvious approaches are just to look through the ribbon and tabs, exploring each option, dropping down menus and hierarchal menus, and searching the Internet. Word also has contextual tabs that display only when a specific object is selected. For instance, two tabs appear when you select a table. This means it's not always possible to know all the options just by looking at the ribbon. Another way is from the Customize Ribbons option (File|Options|Customize Ribbons|Choose commands from|All commands). This lists all the commands, even one that are not available through the ribbon. You can print the commands list through the Macros option (Developer|Macros|Macro Name|List Commands, or Alt-

F8). This macro option also lists the keyboard shortcuts, a useful reminder of those available options.

Fields

Fields are dynamic text areas. You already use several of them such as page numbering, “page x of y”, and dates. Fields calculate for you and then automatically display and update. You place fields through the Field dialog (Insert|Quick Parts|Field). The 62 fields come in nine categories such as Documentation Information (document properties), Indexing and Tables (glossaries and tables of contents [TOCs]), and Equations (math). They use smart defaults, and often you can use them without additional changes, but you can edit and customize most fields. For instance, you can change the look of TOCs by specifying styles instead of headings.

Building Blocks

Building Blocks are reusable text and graphics. Almost anything can be stored as a Building Block such as text, footers, graphics, complete pages, and complex layouts. The writing community also refers to these as snippets. Word includes predefined Building Blocks, including headers, footers, cover pages, text boxes, equations, page numbers, tables, and watermarks but you can also add your own. In addition, if they were added to document or global templates, all team members could access them. You place and edit entries from the Blocks Organizer dialog (Insert|Quick Parts|Building Blocks Organizer).

Other Features

There are other automation features that don't fall into categories. Word has a rich find and replace system including regular expressions, called wildcards in Word, for both finding and replacing text. You can find irregular number patterns such as serial, part, and telephone numbers. Many overlook the regular expression replace features. With these you can store the matches and replace them in a new order. For example, if have a list of names in the *last, first* order such as *Heller, Joseph*, with wildcards, a Find “(<*>), (<*>)”, and Replace “\2 \1” results in *Joseph Heller*. A more dramatic example is to clean Word HTML files. For instance, Word generates notoriously convoluted HTML when converting from a Word file. You can clean up that code in five to seven search statements and end with pristine HTML code. In either case, especially given complex wildcard or search statements, storing those as macros saves considerable time and possible rework.

Another overlooked feature is that Word saves the last 24 copies into separate clipboards, so you can choose which to paste later. This feature is available from the Clipboard panel by clicking the small expand arrow (Home|Clipboard panel).

Programmatic Features

Programmatic features refer to using a programming approach to automation. This includes macros and computer programming. Macros are also the most common perception of what automation is.

Macros are the ability to record sequences directly from the user interface, and then saving, editing, rerunning, and sharing them. To create a macro, you performance the sequence while recording it. You can record almost any action, from typing a simple “Hello, Word,” to perhaps a series of complex find and replaces. The exact same sequence plays back when invoked. While mouse movement and clicks are not recorded, keyboard navigation and tabbing are. Macros can't be modified so you have to re-record the macro to make changes or additions. They can be edited programmatically, and Word even has a built in code editor.

Programming is using computer code to control Word. Typically, you can use programming languages like Microsoft .NET's C# or VB.NET to build a Windows application controlling Word by tapping into Word's API. This approach provides the most control over Word (there are even several commands available only through the API, and not the graphical interface). It also is perhaps the most complex approach, requiring programming experience. There are times, however, that this is the only practical solution. For example, you might have a several hundred page document with the requirements that you find all hyperlinks, whether styled as links or not, list them in a separate document, and test each link. The other automation techniques would be onerously prohibitive. Yet, this application could be developed in about an hour by an experienced programmer and would complete the task in a few minutes.

The return is easy to see. An hour of a developer's time could save literally 40 hours of a writer's time, and yield better quality if you consider the accuracy and reduced opportunities for errors. The savings would multiply if you run that task several times, if not routinely.

Putting Them Together

The previous sections were just a very brief introduction to the individual automation techniques. Now comes the creative process of using them to make your efficiencies.

Review the features. Before starting, review the features available by looking over the shortcut list. With as many as there are, some will be new and others surprising (check out the Spike, for instance). At the same time, review the default keyboard shortcut list. Scroll through the Fields and their option, and look at the Building Blocks. In short, get more familiar with the capabilities. Automation will be easier and more productive to those who know Word better and deeper.

Start small. Don't begin with any solutions at first. Rather, try to incorporate the existing capabilities just to get comfortable with them. For instance, save all the bolding or italics formatting until the end of the writing and then use F4 to speed through them. Try writing and formatting a document without using the mouse at all.

Macros. Using macros is likely for any serious automation solution. It's really the only way to record a sequence of events, and it's going to take a little practice. You may have to rehearse the sequence and make several attempts to record it correctly. Since it only records events, there is no time concern, so go through the steps slowly. You also won't have any repetition ability, such as going through a sequence or a series of find and replaces a number of times, but you can have a macro call a macro. Use that if you need to repeat a sequence two or three times.

Programming. Macros are editable code and Word has a built in editor (Developer|Code|Visual Basic). By default, that editor does not display in the ribbon, so use File|Options|Customize Ribbon and select Developer in the right scroll box). VBA (Visual Basic for Applications) is the scripting language and there are many books and Web sites that explain that. However, VBA is a simple and clear language as far as programming languages go, so modifying it doesn't have to be difficult.

Conclusion

This was only a very brief overview on a larger topic. Most features were only mentioned superficially, and many were never mentioned at all. But it's important to see automation as more than just macros, and that it isn't intimidating. Quite the opposite. It should be inviting. If writers are to help ourselves, then Word automation in specific, and process improvement in general, is a simple and produc-

tive start. A full understanding takes three components: Knowledge of your processes, knowledge of the tools, and creativity to join the two together.

Resources

Delwood, Robert. *The Secret Life of Word*. (Laguna Hills, CA: XML Press), 2013.

Author Contact Information

Robert Delwood
delwood@isomedia.com

Author Biography

Robert Delwood is a programmer, writer, and programmer-writer currently in the Chicago law community but formerly with NASA's Johnson Space Center and Microsoft. With more than 18 years' experience, he's written and documented topics from Windows kernel-level device drivers and speech recognition APIs/SDKs for Microsoft, to help desk procedures and application manuals for the military. He specializes in Microsoft Office automation with VB/VBA and .NET VSTO. He's authored several books, the most recent one, a college-level text book, *The Secret Life of Word*, (XML Press, <http://xmlpress.net/publications/word-secrets/>) about Word's automation for technical writers, non-programmers, knowledge workers, or anyone who wants to do more tasks quickly with Word.



THE SOUND OF HELP—IMPROVE YOUR VOICEOVERS

Bryna Ann Gleich

As multimedia becomes a standard element of technical communication, practitioners who previously focused on text may need to expand their skills and tools so they can perform, record, and edit voiceovers for use in screencasts or podcasts. This paper describes the importance of voice, addresses performance anxiety, recommends techniques for practice, recommends strategies for recording, discusses the use and limitations of audio editing software, provides an overview about different types of microphones and their use, and introduces options for acoustic room treatments for a range of budgets.

Screencasts and other multimedia have become an integral part of many technical communicators' portfolios. For audiences to receive maximum benefit from the content, it must be clear, concise, complete, and pleasant. Technical communicators must ensure that all aspects of the content they deliver—including text, audio, and visual content—are high quality. For content creators who previously focused on strategies and skills for content that is primarily textual, this can be a challenge. Technical communicators who provide this content must therefore develop additional skills and strategies to best serve their audiences.

Voiceovers Convey Human Empathy

Although text and illustrations can be used to effectively communicate with an audience, there is undeniable power in the sound of the human voice or the appearance of a human on camera.

Since many technical communicators produce screencasts, but rarely appear before a camera, we must effectively wield our voices as communication tools. We're expected to teach complex material with patience, to explain recent or impending changes while reassuring users that these changes are in their best interests, and to help our audiences believe that they can accomplish their goals with the help of the user assistance we provide. We must use our voices to make human connections with our audiences so they trust our messages.

If you're a new speaker, the mere thought of sharing your voice with the world might be daunting. You may even be frightened of being judged by listeners who you will never meet.

Fortunately, we don't need to sound quite as amazing and entertaining as the esteemed voices of Morgan Freeman or David Attenborough. You don't need to be a professional audio engineer, own a studio, or have a background in music or drama.

With simple techniques and tools, you can become more comfortable and skilled with creating high quality voiceovers.

Inner and Outer Voice

One of the first challenges you may need to overcome is to learn to tolerate the sound of your own voice.

You might remember the first time you heard a recording of your voice. Perhaps you recorded a new greeting for your voicemail and then were shocked when you listened to the playback. The recording probably sounded too high pitched and the "warmth" was missing. Your mind rebelled, "That's not what I *really* sound like! I don't really sound like *that!* Do I?"

In fact, you probably do sound like that to most people who hear you speak. Modern microphones and speakers usually accurately reproduce your voice. Your equipment isn't lying to you.

Technology and Development

Still, your voice does sound noticeably different when *you* hear the playback. This is normal and there is a logical explanation for this phenomenon. The problem is that your ears have been misleading your brain for your entire life.

When you speak, air from your lungs is forced through your vocal cords, causing them to vibrate. These vibrations create sound waves. These sounds are expelled from your mouth, where your facial muscles also manipulate the sound waves to create your voice and speech. When the sound leaves your lips, it travels through the air around you and bounces around your environment. It reaches other listeners in the room, your microphone, and even your own ears. Depending on your environment, you (and your microphone) may even detect an echo as the sound is bounced off very distant surfaces. This is the voice you hear when you play back a recording of yourself. This is the public, “outer” voice that the world hears when it hears you.

But the vibrations also travel an inner path, through your neck and jaw, to reach your eardrum from the inside. Since this path consists of flesh and bone—rather than air—the sound waves are transformed differently. When this “inner” voice reaches your ears, it’s lower pitched. The extra bass creates a warmer tone. Because this sound is specific to your body, only you can hear it. No other listeners or microphones can detect it.

When you speak, you are accustomed to hearing both your outer and inner voices. Hence, when a recording forces you to hear the outer voice, divorced from the inner, you’ll think it sounds “wrong.” When you listen to a recording, you’re forced to hear yourself as the world hears you.

Technical communicators who frequently perform with on stage with sound amplification systems, or who sing in ensembles and concentrate on their outer voice to stay in tune, may find it easier to accept and appreciate the sound of their “real” voices. For others, increased exposure to the sound of your “outer” voice may help. Edit your own voiceover recordings to increase your exposure to your own voice. This helps you objectively identify the strengths and weaknesses in your own recordings.

Techniques to Improve Talent

To create a high quality voiceover, you must become comfortable performing and learn how to make the most out of that performance. Fortunately, there are many ways to improve your performance.

Listen to Great Speakers

To be a good speaker, it helps to know what good speakers sound like. Identify multiple speakers who you respect and enjoy. Focus on speakers who sound natural and conversational, rather than speakers who intentionally modulate their voice to “hype” a message. Listen to speakers who have different ages, accents, and genders, yet pleasant voices. Make note of what you like and what bothers you. What do you find engaging or distracting about each speaker?

I recommend *National Public Radio*, *Public Radio International*, *BBC Radio*, podcasts, and audiobooks from *Audible*. There are also many helpful voice coaches who provide tips and tricks via *YouTube*.

Listen to Yourself

Before you need to record a voiceover for an important project, record a sample of yourself and listen to it to identify your strengths and weaknesses.

You can practice tongue twisters to improve some of your weaknesses. For example:

1. If your S sounds are harsh or hiss, practice “Sally sells sea shells...”
2. If you struggle with harsh plosives that make the waveform spike or the mic “pop” (P, D, K, G, etc.), practice “Peter Piper picked a peck of pickled peppers...”
3. If you mumble and slur your consonants, practice “Betty’s bitter batter...”

You can also practice by reading a children’s book, such as *Fox in Socks* by Dr. Seuss. For an added challenge, try to clearly enunciate a tongue twister while biting a pencil.

Additionally, if you have a noticeable accent, be aware of it.

- “Good” accents can add interest and help hold a listener’s attention. They’re genuine and sincere. They may even add character or “local flare” to your recording.
- “Bad” accents may be associated with negative stereotypes, which make it harder for some listeners to pay attention to the content of your speech. They may be associated with classism, ignorance, or even scam artists, which make it harder to listeners to trust your message. Very strong accents may be hard for some listeners to

understand, especially if you have a national or global audience.

It's ok to record voiceovers with an accent. However, if your accent is “bad,” consider practicing problematic phrases using neutral pronunciation to reduce it. For example, as someone who has lived almost all of my life in the American South, I know if my mouth is wide when I pronounce vowels (especially the “i” in “ice tea”) listeners from other regions (including my grandfather from Kansas) think I sound ignorant. So before recording, I practice keeping my mouth “round” as I say problematic vowel sounds. Or, I'll imagine that I'm speaking to a someone from out of state to help me switch off my accent.

Use a Script

Always prepare a script and share it with your stakeholders in advance of recording. A complete, correct, and concise script ensures the voiceover recording will also be complete, correct, and concise. Concentrating on a script may also help you avoid accidental filler words (such as “um”) since you don't have to think about your word choices as you speak.

Ideally, you should record audio separately from the screencast action. This reduces multitasking and makes it easier for you to be deliberate with your speech and, later, equally deliberate your cursor movements. In editing, you can then sync the video to match your audio. It's easier to adjust the speed of a track without sound, than a track with sound. If you try to speed up a track that already has audio, it'll sound like a chipmunk is talking.

Before you record the audio, read the script aloud to practice it and mark anything you want to concentrate on. When appropriate, edit the content of the script to sound more natural, avoid accidental tongue twisters, or make the performance easier.

If you're familiar with musical notation, you can use those symbols to mark up your script. Otherwise, make up your own symbols as needed. You may want to indicate

- when to pause to take a good breath during a long sentence.
- when you need to emphasize something.
- tempo or speed. Slow down for key points or if you tend to stumble on a tricky phrase. Speed up slightly to show excitement.
- pitch changes. Although most pitch changes will come naturally and subconsciously, if pitch is hard for you, mark it. Use a rising pitch at the end

of a question. Use a falling pitch at the end of a statement. For an introduction, try to start with a higher pitch and then slide downward. The high start gives the listener's ear a starting point, so later you can raise your pitch for emphasis and yet the emphasis won't sound out of place.

Use Your Whole Body

Whenever possible, stand up to improve lung and diaphragm capacity. This also insures you won't accidentally record a squeaky desk chair. If you're used to public speaking, you're probably already accustomed to standing while you perform.

Even though no one will see your hands on a screencast, you may find it easier to use your hands to help emphasize your speech. Keep your hands empty. Put down anything that could make fidgeting sounds. Step back from anything that you might bump with your hands.

Loosen up. Do whatever you need to do to relax: stretch, talk to yourself, make silly faces. Shake off any nervous energy or negativity that might otherwise sneak into your voice.

Remember that your facial muscles and expression affect the sounds that you produce. Even if you're tired, nervous, or stressed, you want your listener to think you are a pleasant speaker. Since the listener can't see your subtle body language, exaggerate your body and facial expression to ensure the desired “body language” is apparent in your disembodied voice. Generally, you'll probably want to speak with a huge smile. Occasionally, you may want other expressions, such as for a stern warning.

Try this: Create two audio recordings. First, while yawning or frowning, try to sound friendly. Next, try to sound friendly while grinning from ear to ear. Listen to the results and ask a few people to listen too. Many listeners can hear the difference in a speaker's body language. Listeners will probably report that the second recording sounds more genuine than the first one. As a bonus, acting happy may make you feel happy (so you'll be even more likely to enjoy recording voiceovers).

Imagine the Audience

When you speak, imagine the audience and speak as though you were talking to them instead of to a microphone in an empty room. Don't imagine an audience of millions of strangers on *YouTube* (which may cause stage fright). Imagine a friend or family member who needs

Technology and Development

your help. Imagine you are having a casual, friendly conversation with that person as you perform the voiceover. These days, there's a good chance that someone in your audience is sitting at their kitchen table, trying to get some work done after their kids have gone to bed—imagine speaking to them.

Successful Recording Sessions

When you're ready to record your performance, follow these strategies to ensure the recording session is efficient.

Test your Equipment

Even if you used the same equipment recently, recheck your hardware connections and software settings before you record. Record a sample of yourself and listen to it. Then adjust your equipment and your performance accordingly. Test your equipment to reduce problematic surprises. Maybe you'll need to relocate because of a noisy maintenance crew next door. Perhaps you need a glass of water or cough drops because your allergies are making your voice gravelly. The battery on your wireless mouse could be dead. Don't assume everything is set up perfectly for you to use on recording day; test it yourself.

Keep Going

When you're ready to record, press the red record button and then try not to stop until after you've recorded the entire script. If you make a mistake, just pause, breathe, and restate what you meant to say, but keep recording.

During editing, you can remove the pauses and segments you don't want to keep.

Sometimes what you thought was a mistake will actually sound like your best take when you listen back to it.

Mark the Wave Form

As you record, you may be able to watch as the audio recording software creates a wave form. With experience, you'll be able to recognize some of the peaks and valleys in that wave form as errors by sight, without listening back to all of it. You'll see your sneeze, lip smack, or occasional stuttering, in the waveform's pattern.

You can also mark the waveform as you record it to make editing easier. Some recording software enables you to use hot keys to mark wave form as you record it (CTRL + M).

For a low tech solution, clap your hands twice near your microphone. This creates a clear pair of spikes on your wave form. When you edit, look for the pair of spikes where you marked the end of an error or something you said well.

If you're also using a DSLR camera, clap your hand directly in front of the camera lens. In editing, you can sync the spike in the audio wave form with the visual of the clap in front of the camera. This is similar to the clapper boards used by major film studios.

Audio Editing Software

Although most video software enables you to make basic edits to your audio files, dedicated audio editing software, such as *Adobe Audition* or *Audacity*, usually provide more options to improve your voiceover recording. Within the audio software, the sounds you recorded appear as a series of squiggly lines known as a wave form.

You can use audio editing software to:

- remove a steady, repetitive, background noise (such as the hum of an air conditioning unit or a computer fan).
- remove isolated clicks, lip smacks, loud breaths, and more.
- reduce plosives. Plosives typically occur on harsh consonants like P or K.
- reduce mic rumble, which can occur when a speaker is a little too close to the microphone.
- tweak volume levels.
- tweak the tone of a recording.
- boost the bass a little.

However, audio software can't fix all audio problems. For example, it probably won't fix:

- the chatty cubical next door. Since these sounds are unpredictable human voices, software won't be able to isolate it.
- an extremely loud or quiet voiceover recording. If you attempt to adjust the waveform too much it may sound artificial, clipped, or compressed. It's better to set your microphone "gain" correctly so that the recording is already close to the right volume.
- a speaker who is extremely close to the mic and thus has a lot of plosives and mic rumble.

- an echo.
- stilted, unnatural speech.

You can improve a lot of errors with editing software, but the original recording and performance are still important.

Microphones

You must consider your recording space and the type of microphone you plan to use. Then use the microphone correctly for that situation. If you use a high quality or expensive microphone improperly or under the wrong conditions, you will not capture a high quality audio recording.

Pick Up Patterns

Whether you are using microphones your company previously purchased or seeking to purchase new hardware, be aware that microphones have different pick up patterns. These patterns are like an invisible, three dimensional, “bubble” around each microphone. The shape of the “bubble” varies based on the type of microphone.

To use the microphone effectively, you must ensure that the source of the sound (your voice) is within that “bubble,” but not so close to the mic that you pick up sounds you don’t want (such as too much breathing, lip smacking, or other incidental noises). For some microphones, moving your mouth even an inch can drastically change how the microphone records your voice. For those mics, you’ll need to use good physical discipline; ensure you maintain a consistent relationship between your mouth and microphone throughout the recording. Other microphones are less sensitive and more “forgiving” of speakers who move as they record.

- **Omnidirectional**—This “bubble” is spherical. The microphone picks up sounds from all sides of the microphone relatively evenly. This type of mic is typically more “forgiving” and easy to use. However, since the microphone picks up sounds from all directions, you must take extra care to ensure that you record in a quiet location. These microphones work well in dedicated, recording studio spaces.
- **Bi-directional or figure 8**—This type of microphone has a pair of “bubbles” at the front and back of the mic, where it is the most sensitive. However, if you try to speak from the sides of the microphone, it may not pick up your voice at all.

- **Cardioid & super cardioid**—These “bubbles” are “heart shaped.” They resemble a sphere that’s being compressed by an impact from another ball. The front of the mic picks up sounds with at least twice the sensitivity as the front of an omnidirectional mic. Sounds from the sides of the mic are somewhat comparable to an omnidirectional mic. Be aware that if you start speaking in front and accidentally move your head to the side, this movement will be evident on the recording, since the audio is picked up differently in each position. Cardioid mics pick up less sounds from behind the microphone (along the handle) than omnidirectional mics. Super cardioid mics are even more sensitive from the front (some are ten times more sensitive), but they do pick up a little sound from behind the microphone too. These microphones are typically used for live stage performances or in rooms without sound treatments, since you want to pick up the performer well, but exclude sounds from other directions. Maintain a consistent physical relationship between your mouth and the mic. Be aware of other sounds in the environment.

- **Shotgun**—The “bubble” for this type of microphone is very directional. It’s less like a typical helium balloon and more like the long ones used to make balloon animals. These mics are great for recording in noisy environments on camera, because they can pick up a speaker at a greater distance while still ignoring unwanted sounds in the surrounding environment. However, you must successfully point them directly at the speaker at a consistent distance. If the shotgun microphone isn’t pointed directly at the speaker, it won’t pick up the voice at all. As a speaker, try not to move sideways, or forward and backward out of the mic’s narrow bubble. This is the least “forgiving” microphone pickup pattern. These mics are useful if you include DSLR bumpers in your screencast videos. Typically, you’ll mount your shotgun mic on a “boom” stick to hold it out of a camera’s frame and point it at stationary performer.

Some microphones have multiple pickup patterns and enable you to select which pickup pattern you need each time you use them. If you use a microphone with “variable direction,” pay attention to your surroundings each time you record, and then select the pick up pattern that is most appropriate for your current recording situation.

Proximity Effect

The volume and pitch of the speaker's voice also affect how you must position your mouth in relationship to the microphone. Typically, microphones pick up sounds which are higher pitched from greater distances. To pick up a lower pitched sound, the source of the sound must be closer to the microphone. So depending on the individual speakers, women with higher pitched voices and men with deep voices may need to use the same microphone differently.

You might notice this phenomenon during conference calls with multiple speakers in a room. A man with a deep voice may need to be physically closer to the microphone as he speaks, than a woman with a higher pitched voice. With directional mics, he may also need to take extra care to stay in front of the microphone, where the pick up pattern is the most sensitive.

The closer the speaker is to a microphone, the more bass the microphone will pick up. Bass makes the recording sound "warm," but too much bass can make it sound "muddy." Remember, you can also add a little extra bass when you use audio editing software.

What and Where

Since different microphones are optimized for use in different recording environments, it's important to consider what you are attempting to record and where you are recording it.

- **Scripted voiceover recorded in a studio**— Assuming your studio is quiet, you'll probably use an omnidirectional microphone mounted to a stand for hands free use. The microphone may be in a shock mount to reduce the chance of it picking up unwanted vibrations from other forces in the room. For most studio microphones, you should position your mouth six to twelve inches away from the microphone. To estimate this distance, hold your hand in a fist with your thumb and pinky finger extended. Place your mouth near your thumb and the mic near your pinky. If you have a pop filter, position it between your mouth and the mic. Pop filters are made of a thin, permeable fabric, and are usually inexpensive. They break the "wind" that occurs with a speaker creates a plosive sound ("P"). Pop filters also help protect the microphone from spittle, which would otherwise damage the microphone.
- **On camera**—Technical communicators who share or borrow equipment from their company's marketing department may be more likely to use equipment designed for use with a DSLR video camera. You might use a shotgun microphone mounted on a boom pole (if so, the speaker must be careful to stay within the microphone's pick up range) or a small lapel or "lavalier" microphone clipped to the speaker's person to enable the speaker to move more freely. If you use a lapel mic, plan ahead. Wear structured clothing, such as a collared or button down shirt so you can easier clip on the microphone. Ensure your hair, jewelry, and clothing do not inadvertently rub against the lapel microphone. To hide the lapel cord, consider dressing in layers. Know how close the mic needs to be to your mouth and monitor the audio levels in real time to ensure the microphone "gain" is at an acceptable level. You may need a partner to monitor your audio levels.
- **Live webinar**—Since you'll probably need to multitask, a headset microphone may be the most convenient. Use a wired connection instead of Wi-Fi to ensure your signal is as consistent as possible. Use a good quality headset and ensure your area is quiet. Before the live session, record a sample of yourself speaking, listen to it, and adjust accordingly. Depending on the mic, you may need to reposition it to reduce plosives and to avoid loudly breathing into the mic.
- **Teleconference**—If your video will include segments of recordings from interactions with clients (such as quotes from usability tests), you may need to use recordings from a teleconference. This audio can be difficult to use because it's not usually very "clean." For the best results, ensure you have a strong leader/host to control the conversation. The leader should make sure people who aren't talking are muted and that speakers do not talk at the same time. Speakers with quiet or deep voices should be seated closest to the microphones. You can also tap the phone line directly into your computer's input jack.
- **Stage performance**—Although you might use a lapel mic for a stage performance, you could also use a (super) cardioid mic mounted to a mic stand or held in your hand. Remember to speak directly into the front/top of the cardioid microphone. You may need to be very close to the mic—as close as 2 inches. Typically, you should ensure the mic is parallel to the ground or tilted at most 45 degrees from parallel, rather than vertical. If you do not have a stand, be sure to hold

the mic by the handle, rather than by the head. If you hold the mic incorrectly by the head, you'll interfere with the pickup pattern—your physical interference can reshape the three dimensional “bubble” around the mic and cause it to pick up sounds you don't want or to ignore sounds you do want.

Room Treatments

Remember, the sound a microphone picks up is affected by the space around the microphone. Sound waves leave the speaker's mouth, travel through air, and bounce around the room, where they are picked up by the microphone (and your ears).

Many technical communicators who produce screencasts will use a microphone in a room alone. The room may be a dedicated studio or a multipurpose space. You can modify or “treat” the space to improve the quality of your recordings.

Consider your budget and affinity for “do it yourself” projects:

- If you have thousands of dollars and want a space that's pre-made, consider purchasing a stand alone, vocal isolation booth (such as a *Whisper Room*). Booths come in a variety of sizes, shapes, and may even include windows and other upgrades.
- If you have hundreds of dollars and a few couple friends who know how to use basic power tools, you can purchase acoustic foam, special sheet rock, thick carpet, and other products to use to convert a small space (such as a closet) into a recording studio. Numerous websites and online videos demonstrate how to construct a studio for your space.
- If you need a portable solution, you can buy reflection filter for about a hundred dollars. You can build one even cheaper by placing your microphone into a box full of foam and speaking into the open side of the box. (When created my own, it seemed physically awkward to use. I didn't feel comfortable speaking into a small box while also reading from my script. It felt cramped.) Reflection filters do not make a space sound proof. However, they do *reduce* the sound waves that would otherwise bounce freely into the back or sides of your microphone. They make it sound like you are recording in a smaller, more intimate space.

- If you have zero budget and no space for dedicated studio, consider recording in a closet full of clothes or under a blanket fort. Although it may be physically awkward to work inside a closet, the fabric will dampen your voice from bouncing around the room, thus preventing a noticeable echo. The fabric also dampens some noises from your surroundings. If you have a modern car, you can also drive your car to a quiet location, turn it off, and then record your audio from inside the car. Your modern car is sort of like an audio isolation booth on wheels. When you record inside your car, you take advantage of the audio engineering that normally cancels out sounds of the road and engine as you drive.

Conclusion

When screencasts and other forms of audio/video media are expected deliverables, technical communicators who are responsible for this content must ensure they produce high quality voiceovers and audio recordings.

To do this, you must become comfortable with the entire recording process. To improve your vocal performances, practice and apply simple techniques. Prepare for each recording session and vocal performance in advance. To effectively manage the recording session, be aware of your surroundings and the content you want to record. If your current recording space is inadequate, you can improve the space without breaking your budget. Learn to use hardware, including microphones, properly. After audio is captured, use special software to edit the wave form to further improve the audio recording.

When technical communicators produce high quality screencasts, they ensure audiences have a pleasant experience with complete, correct, and concise content.

References & Resources

- Adobe Audition Help & Support (April 2016) <https://helpx.adobe.com/audition/topics.html> .
- Adobe Premiere Help & Support (April 2016) <https://helpx.adobe.com/premiere-pro.html> .
- Audacity User Wiki (April 2016) http://wiki.audacityteam.org/wiki/Audacity_Wiki_Home_Page .
- Audible (Audiobooks) <http://www.audible.com/> .
- Ayres, Meryl. “The Story Behind Slack's Interface-Focused Walkthrough Video.” Wistia Blog (March 2016) <http://wistia.com/blog/slack-product-video>

Technology and Development

- BBC Radio <http://www.bbc.co.uk/radio>
- Beatty, Chris. “What Does YOUR VOICE Sound Like to Others?” (August 2009) <https://www.youtube.com/watch?v=eWLWOSgYlG8> .
- Brown, Rupert. “How to build an acoustic diffuser.” The DSPproject. (May 2013) https://www.youtube.com/watch?v=oHTmNyo_000 .
- DeWees, Bill. Voice Over Expert—Video Playlists. <https://www.youtube.com/user/VoiceOverExpert/playlists>.
- E-Home Recording Studio. “A Beginner’s Introduction to Microphone Polar Patterns.” (April 2016) <http://ehomerecordingstudio.com/microphone-polar-patterns/>.
- Fox 11 News 6. “Million Dollar Voices.” (uploaded by tpvideoprods (July 2008) <https://www.youtube.com/watch?v=3S8N2s3L4mQ> .
- Glass, Ira. “If You Don’t Have Anything Nice to Say, SAY IT IN ALL CAPS: Act Two—Freedom Fries.” This American Life (Aired January 2015) <http://www.thisamericanlife.org/radio-archives/episode/545/if-you-dont-have-anything-nice-to-say-say-it-in-all-caps?act=2>.
- Johnson, Tom. “Slides and recording for ‘Perfecting the audio narration in instructional video’ at Info Dev World.” I’d Rather Be Writing (October 2014) <http://idratherbewriting.com/2014/10/23/slides-and-recording-for-perfecting-the-audio-narration-in-instructional-videos-at-info-dev-world/>
- Johnson, Tom. “Finding an Acoustic Environment (Voiceover).” I’d Rather Be Writing (May 2010) <http://idratherbewriting.com/finding-an-acoustic-environment-developing-a-personal-voice-in-audio-series>
- Joyner, Paul; Robjohns, Hugh; White, Paul. “Studio SOS: Building a DIY Vocal Booth.” Sound on Sound Magazine (March 2006) <https://www.soundonsound.com/sos/mar06/articles/studiosos.htm>, <https://www.youtube.com/watch?v=HdEYNdzi4kw> (part 1, March 2010), <https://www.youtube.com/watch?v=U8oXmTpUT10> (part 2, March 2010)
- Lam, Joe. “DIY Portable Sound Booth—Test and Review.” Media 6D. (June 2013) <https://www.youtube.com/watch?v=rWgLCpOaAzo> .
- Levine, Dan. “Voice Over Training and Technique.” Such a Voice (November 2008) <https://www.youtube.com/watch?v=XmsoPMJRtZE> .
- Lavigne, Chris; Mills, Dan. “Recording Audio for Business Videos.” (April 2016) <http://wistia.com/library/recording-audio-for-business-video> .
- Librivox Community Forums. “Making your reading sound great.” (Original post: January 2006) <https://forum.librivox.org/viewtopic.php?f=23&t=872> .
- Librivox: Free Public Domain Audiobooks. “Newbie Guide to Recording.” (April 2016) http://wiki.librivox.org/index.php/Newbie_Guide_to_Recording .
- Librivox: Free Public Domain Audiobooks. “Volunteer for Librivox.” (April 2016) <https://librivox.org/pages/volunteer-for-librivox/> .
- National Public Radio <http://www.npr.org/> .
- Public Radio International <http://www.pri.org/> .
- Rhett & Link. “Why You Hate the Sound of Your Own Voice.” Good Mythical Morning (October 2012) <https://www.youtube.com/watch?v=N970lhPK1hY> .
- Sound on Sound. “Is a Reflection Filter worth the money?” (August 2011) <http://www.soundonsound.com/sos/aug11/articles/qanda-0811-2.htm> .
- TechSmith. Blog. (April 2016) <http://blogs.techsmith.com> .
- TechSmith. “Camtasia Studio 8—Get Great Sound.” (June 2012) http://assets.techsmith.com/docs/pdf-camtasiastudio/get_great_sound_with_camtasia_studio_8.pdf .
- TechSmith. “Tips for Voice Overs.” (March 2010) <https://www.youtube.com/watch?v=a6hXoU426Hc>
- Voice123 Community Forums. <http://voice123.com/forum/> .
- Wavelength Media. “How Do Microphones Work?” (April 2016) <http://www.mediacollege.com/audio/microphones/how-microphones-work.html>.
- Whisper Room, Inc. “Sound Isolation Enclosures.” (April 2016) <http://www.whisperroom.com>.
- Wikipedia. “Microphone Polar Patterns.” (April 2016) https://en.wikipedia.org/wiki/Microphone#Microphone_polar_patterns .
- Wikipedia. “Pop Filter.” (April 2016) https://en.wikipedia.org/wiki/Pop_filter .
- Wikipedia. “Proximity Effect (Audio).” (April 2016) [https://en.wikipedia.org/wiki/Proximity_effect_\(audio\)](https://en.wikipedia.org/wiki/Proximity_effect_(audio)).

Author Contact Information

Bryna A. Gleich
Sr. Technical Writer
K-12 User Education
Blackbaud, Inc.
2000 Daniel Island Dr.
Charleston, SC 29492
bryna.gleich@blackbaud.com
<https://www.linkedin.com/in/brynagleich>
brynagleich@gmail.com

Author Biography

Bryna Ann Gleich is a Senior Technical Writer at *Blackbaud*. *Blackbaud* combines technology solutions and expertise to help nonprofit, charitable giving, and education communities achieve their missions. *Blackbaud* works in over 60 countries to support more than 30,000 customers, including nonprofits, K–12 private and higher education institutions, healthcare organizations, foundations, and other charitable giving entities, and corporations.

Blackbaud employs over two dozen technical writers on a team known as User Education. Within the User Education team, Bryna currently leads the “AV Club.” This sub-committee is devoted to:

- establishing standards to ensure high quality videos,
- advocating video best practices,
- training new and existing employees,
- coordinating video efforts across products and departments,
- learning ways to get the most out of our existing tools,
- researching new video tools,
- exploring and improving delivery methods,
- analyzing view metrics,
- and enjoying professional development and innovation as it relates for videos as user education and user assistance.

One of the “AV Club’s” initiatives for 2015 was to determine how technical writers on the User Education team (in all offices nationwide) could achieve consistently, high quality audio. This initiative was vital because of increasing collaboration with the Marketing department (who required very polished video content). Additionally, User Education needed to provide usage guidelines and recommend hardware for Customer Support to purchase. Much

of the information presented in this paper is a direct result of that initiative.

Bryna has worked on multiple, agile, scrum teams and products since she joined *Blackbaud’s* technical writers in 2011. She currently works with K-12 products, which enables her to use her prior experience as high school teacher to relate to the users she serves. In 2008, she graduated from Clemson University with a bachelor of arts in English. She switched to major in English after successfully completing her sophomore year as an Industrial Engineering major. She was initially attracted to Industrial Engineering because of its focus on human factors and optimizing complex processes. She’s thrilled that technical writing enables her to bridge all of these interests.

She currently lives in Charleston, South Carolina with her husband and her dog.



RELEASE NOTES AUTOMATION

Tanya Ivanova and Barry Grenon

Time to market—getting your product to your customers, fast—is a key differentiator for a company’s success. With software development moving towards Agile methodology and Continuous Delivery, there is less time in a release cycle for the documentation. To keep up, writers and their managers need to look for opportunities to automate certain kinds of docs—Release Notes, for example. The solution developed by the Technical Publications team at Genesys enables writers to capture feature and defect information from our tracking system (JIRA by Atlassian) and publish it to online, wiki-based Release Notes, on short notice. Review and editing processes ensure technical accuracy and style compliance, giving users access to complete, quality content on the release date.

It is a quiet Friday afternoon when you receive an urgent request to write a Release Note for a Hot Fix release that has to go out this week—in other words, in the next two hours. You drop the document you planned to finish today and start chasing engineers for any information, aside from the ticket number, that would be of interest to product users. On the same day, your fellow writer finds out that the next major release of her product will be ready one week early—and that she has to include about fifty tickets into the Release Notes. She starts chasing the engineers, who are scattered around the world, to collect details about various tickets they are responsible for. Yet another writer in your company is on a team that follows the Continuous Delivery process, releasing their product every two to three weeks, each time requiring a Release Note for some two dozen tickets. This last writer feels that he never stops chasing engineers for Release Note information.

Sounds familiar? If you are a technical writer for software, you either have been in one of these situations yourself or have heard about them first-hand. What you might not have heard is that technology coupled with consistent processes could take the stress out of these situations. An automated system for production of Release Notes could help you publish your Hot Fix Release Note within half an hour; enable your fellow writer to collect and format all descriptions for fifty tickets within ten minutes; and set the Continuous Delivery team for efficient preparation of Release Notes for their regular iteration releases. To keep up with the increasingly faster release pace, technical writ-

ers and their managers realize that automation for certain documentation, such as Release Notes, is a necessity.

The Solution in a Nutshell

With software development moving towards Agile methodology and Continuous Delivery, teams are releasing new products and are adding new features faster than ever, leaving less and less time in a release cycle for the documentation. How can writers describe all these features and numerous software corrections given that shorter release cycles leave minimal preparation time for technical documentation? The solution developed by the Technical Publications team at Genesys enables writers to capture feature and defect information from a defect tracking system and publish this content to online, wiki-based Release Notes, in automated fashion, fast. Review and editing processes ensure technical accuracy and style compliance, so that complete, quality content is available for users on the release date.

The Genesys solution is an automation tool that consists of two main technical elements: a custom plug-in built into the company-wide defect tracking system (JIRA by Atlassian) and a script that moves the Release Note content from JIRA to online, wiki-based pages. The key process elements include enforcing Release Note inputs in JIRA and reviewing descriptions both for technical accuracy and for compliance with the company style. The benefits

of the automated Release Note solution include improved efficiency, clear communication between developers and writers, consistent process across product teams, and fast turnaround during the release crunch. Ease of template maintenance for Release Notes across the department is a side benefit of the full implementation. Time savings from using the solution go towards quality improvements in other documentation content.

Why Release Notes?

The Release Note is a perfect example of a document that must be prepared very quickly, yet tends to change until the last minute on release day. Automated production of Release Notes ensures that product teams prepare complete, quality content in an efficient, stress-free way and that product users have timely access to Release Note content on the product release date. Content for Release Notes often comes from various sources. Even when a single developer is responsible for product implementation and is best positioned to describe details of a defect fixed in the product, a product manager might contribute his vision to new feature descriptions, while validation testing might give a tester a further, unique view on both the feature implementation and defect fix. Finally, the writer adds the end-user perspective and professional language to feature and defect descriptions.

revisions, with other stakeholders receiving notifications through the JIRA Watch mechanism. As a result, the feature and defect descriptions are ready for Release Notes by the time that the feature or defect is closed in the tracking system. Occasionally, a new defect is discovered that the developer has no time to fix before the next release; in this case, the developer can describe the known issue or work-around in the ticket, for the writer to capture in Release Notes as well.

During the final release preparation stage, the automation tool helps the writer to collect all the tickets that belong to a given release, properly format the descriptions, and include them into a Release Note, which is then published for the end users. The ticket collection mechanism relies on the native JIRA filtering capability. Features and defects that failed testing can be easily excluded with a small modification to the filter expression. Similarly, late-breaking features or new known issues can be added to the filter—and, thus, to the Release Note—at the last minute. The tool transfers descriptions from JIRA tickets onto a wiki page, which it creates in a draft Release Note version for a particular component. After the writer and the product team are satisfied with the final look of all the ticket descriptions in the context of the full Release Note draft, the production person approves the format. Finally, the writer uses the publication function to make the new Release Note page visible to the users.

Using a centralized tracking system, such as JIRA, to store Release Note information in the feature or defect ticket itself enables all parties to contribute their inputs and

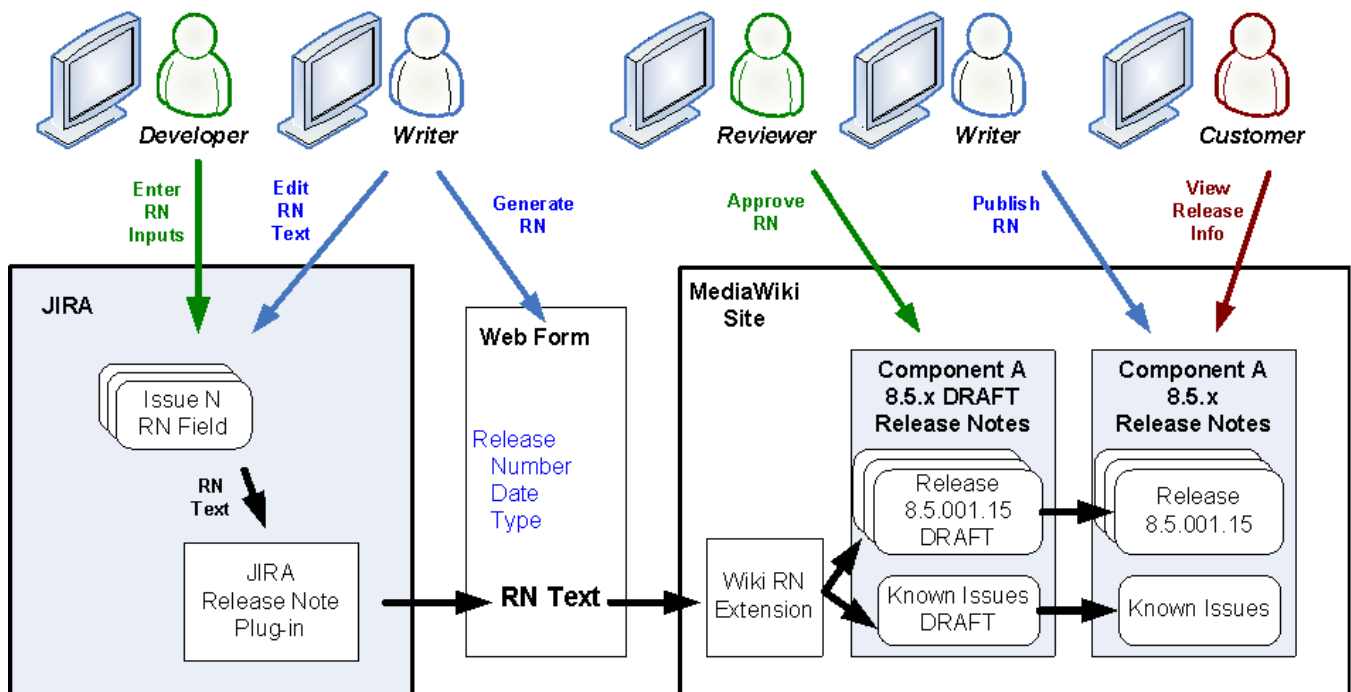


Figure 1. Release Note Automation

How Was the Solution Developed?

In 2011, Genesys piloted a new issue tracking system, JIRA by Atlassian, on a handful of projects that used Agile methodology. In JIRA terminology, an “issue” is any type of request for a new addition or correction to the product. It became obvious from the beginning that, with over a hundred active projects in development and with an ever-growing number of issues for each, teams needed a system for describing user-facing issues and retrieving these descriptions at product release time. The Technical Publications team proactively recognized and voiced this need early in the requirements definition phase. The committee responsible for the new tracking system supported the request. The JIRA development team at Genesys added a special field called “Release Note” to issues of certain types. To take advantage of the information in the new field, a platform developer on the Technical Publications team created a Release Note plug-in (RN Plug-in) that was added to the Genesys JIRA instance in order to collect content of the Release Note field from relevant issues.

Around the same time, the Genesys Technical Publications team started moving user documentation from Adobe FrameMaker-sourced PDFs and RoboHelp-sourced Help files to an online documentation system, MediaWiki, with customized extensions such as [Ponydocs](#) (which

was created by [Splunk](#) for their own documentation and was released as open source). In light of the transition to online documentation, the Technical Publications team decided to change the format and delivery mechanism of Release Notes, which were previously provided as stand-alone HTML files. Instead of a single HTML file per software component that listed all releases for a given release family, a collection of online pages now describes releases of the component, one page per release. The wiki platform developer created a script that collects the Release Note content from JIRA issues and then adds this content onto a new draft page in the online Release Notes system. If users need to be made aware of any known issues, the script brings those in from JIRA as well. The platform developer also implemented a publication function, which enables a writer to publish a new Release Note from a draft version to the online version visible to end users within two minutes.

Note: The Genesys documentation site is currently open only to Genesys customers, partners, and employees.

Who Is Involved with the Process?

Although the process involves multiple members of a product team, the initial steps are spread out throughout

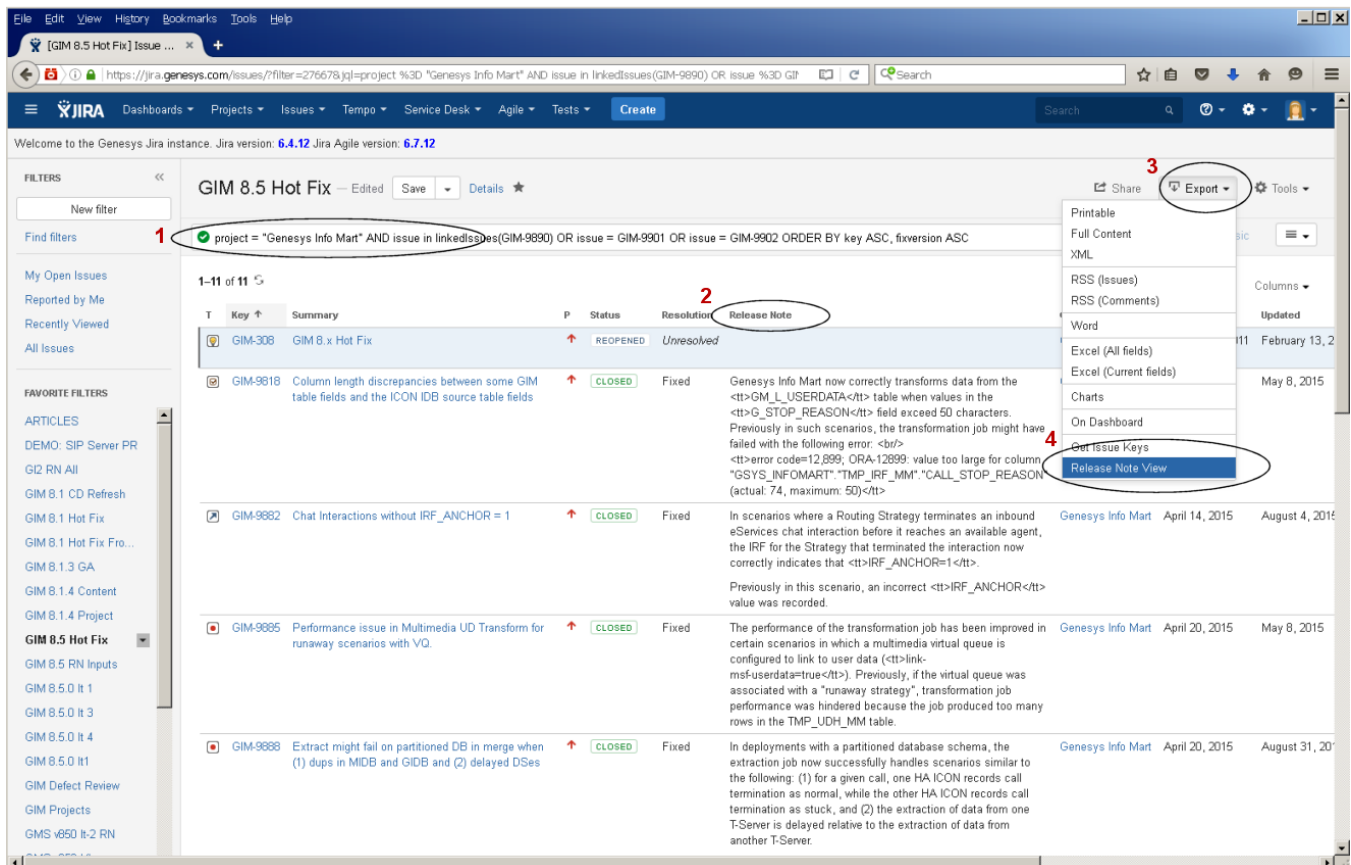


Figure 2. Accessing the Release Note Plug-in in JIRA

Technology and Development

the release cycle, ensuring that JIRA contains approved descriptions that are ready for Release Notes at any moment.

Role-based Process Steps

During the development cycle:

1. Product Team decides which issues (new features, improvements, defect fixes) to expose in Release Notes.
2. Developers supply issue descriptions in the Release Note field in JIRA.
3. Project/Development Manager helps Writer to construct a JIRA Filter for the release.
4. Writer checks that all relevant issues have inputs in the Release Note field and edits descriptions in JIRA.
5. Product Team verifies individual descriptions in JIRA.

During the release preparation cycle:

1. Writer uses the JIRA Filter and RN Plug-in to collect issues and extract their descriptions from JIRA.
2. Writer uses the Wiki RN Extension to generate a release entry (and any new Known Issues) in a draft version of the online Release Notes.
3. Product Team reviews and approves the complete release entry (and new Known Issues).
4. Production Editor approves the Release Note format.
5. Writer uses the Wiki RN Extension to publish the approved Release Note to the Genesys documentation site.



Search this manual Search this site

Technical Documentation | Contact Us | Genesys Forums | Corporate Website | Customer Care | Legal Notices

Browse All Products

Product Offerings

View All Product Offerings

System Guides / Release Notes / Other

Release Notes

Genesys Info Mart 8.5.x Release Note

Known Issues and Recommendations

Release 8.5.0

Genesys Info Mart 8.5.0 Deployment Procedure

8.5.004.06

8.5.003.17

8.5.003.16

8.5.003.12

8.5.002.11

8.5.002.09

8.5.001.21

8.5.003.17

Version 8.5.x (latest release)

8.5.003.17

Genesys Info Mart Release Notes

Release Date	Release Type	Restrictions	AIX	Linux	Solaris	Windows
02/29/16	Hot Fix		X	X	X	X

Contents [hide](#)

- Helpful Links
- What's New
- Resolved Issues
- Upgrade Notes

What's New

This release contains the following new features and enhancements:

- Genesys Info Mart now stores data to distinguish an agent from other persons in a contact center. A newly introduced value, `Person`, is set in the `RESOURCE_RESOURCE_SUBTYPE` column for any persons who are not agents. The previously existing value, `Agent`, is now used in the `RESOURCE_RESOURCE_SUBTYPE` column only to identify Agents (that is, the resources for whom the `IsAgent` flag is set in the Person configuration object). Both subtypes are associated with the `Agent` resource type that is stored in the `RESOURCE_RESOURCE_TYPE` column.

Helpful Links

Releases Info

- List of 8.5.x Releases
- 8.5.x Known Issues

Deployment Procedure

Figure 3. Published Hot Fix Release Note

Using the Automation Components

When the descriptions are ready in JIRA, the writer accesses the RN Plug-in, as follows (see Figure 2):

1. Enter a filter expression for the release.
2. Verify that relevant issues have the Release Note field populated.
3. Click Export to expand the command menu.
4. Select Release Note View to export Release Note descriptions from JIRA.

What Does the Release Note Plug-in Do?

1. JIRA RN Plug-in sorts the issues from a filtered JIRA view (see Table 1).
2. JIRA RN Plug-in creates an HTML view of the RN entry for visual verification.
3. JIRA RN Plug-in calls the Wiki RN Extension.

What Does the Wiki RN Extension Do?

1. Wiki RN Extension creates a wiki page for the release entry and updates the Known Issues page.
2. Wiki RN Extension automates the publication step.

The screenshot of the Genesys documentation site in Figure 3 shows an example of the published Release Notes for a Hot Fix release number 8.5.003.17.

Main Challenges

As with any new tools and processes, it has been challenging to make all product teams adopt the automated Release Notes process across the company.

- To get teams to buy into and cooperate with the new approach, training and communications sessions were held for the Engineering organization, Project Management organization, Technical Publications team, and individual product teams.

Process video and guidelines were prepared and posted on the intranet. The approach was piloted with two different product teams before being rolled out to the rest of the company. Also, product teams were given flexibility with the timing for process adoption and division of responsibilities within each team.

- To make writers more comfortable with the JIRA filtering mechanism, samples of the syntax for filter expressions were provided in the process document. Assistance from product teams with the filter definition was established as part of the process.
- To enforce that Developers enter Release Note inputs in JIRA, a special item in the release checklist was added by the Project Office. Frequent reminders from writers, Project Managers, and even Development Managers help to make this step part of team culture.
- To administer creation of Release Note placeholders on the wiki site and to troubleshoot generation and publication issues, a special group was organized within Technical Publications, with an alias created in the company's email system.

KEY BENEFITS

The automated Release Notes approach simplified routine tasks for writers, brought significant improvements into operations of product teams, and benefited the company as a whole.

Creativity and Innovation

Automation of dull, maintenance tasks frees up writers to do what they do best—add value to the products by working on user-focused content and innovative delivery.

Efficiency Gain

A Release Note being a mandatory documentation piece for every Genesys product delivery, the new system

RN Section Name	JIRA Issue Type	JIRA Issue Type
What's New	New Feature, Improvement	Closed, Code Done, Resolved
Resolved Issues	Defect, Task	Closed, Code Done, Resolved
Known Issues and Recommendations	Defect, Task	Open, In Progress, Reopened, Resolved, or Closed with Resolution: Will Not Fix

Table 1. Summary of the issue types that the RN Plug-in takes from the filtered JIRA view and how it sorts them, depending on the issue status.

Technology and Development

resulted in a huge efficiency gain. Agreed-on processes and automation speed up team operation during a release crunch. As an example, David, a writer for one product line that includes around 20 software components, traditionally engaged from three to five other writers for the final two weeks of his product release cycle to help him with Release Notes. A product from this product line was one of the two pilots for the new Release Notes system. In the two years since his team agreed to move Release Notes online, David has never asked other writers for assistance with Release Notes at a release stage.

Streamlined Communications

Entering feature and defect descriptions into JIRA tickets and having a URL with the up-to-date Release Note draft is bringing reviewers onto the same page—literally. JIRA Watch capability notifies a writer when Release Note information is added to the ticket, and notifies reviewers when a writer updates the description. Changes to the complete Release Note draft are visible in the wiki page history. Additionally, the need for a release-level filter prompts the product team to openly communicate the release scope.

Process Flexibility

Using JIRA as the system for Release Notes storage and internal communications gives the product teams flexibility to work in the rhythm that suits their particular team composition, within the standard workflow.

Competitive Advantage

Having all release information in the system and readily available for publication enables the product teams to decrease the time to market and help the company to differentiate itself from the competition.

Future Expansion

Cloud Release Notes

Genesys now has a SaaS (software as a service) product offering. The Cloud writing team is taking advantage of the Release Notes process already in place and started adjusting the system for the benefit of Cloud customers.

Localized Release Notes

Online Release Notes are becoming a single source of release information, for English-speaking and international customers alike. A translation support matrix has recently been added to the online Release Notes; this eliminates the need to prepare separate Release Note files for individual languages. The solution has been successfully piloted with two products, which were localized into 7 and 14 languages, respectively.

Open Source Offering

In the spirit of open source software – from which our group has benefitted greatly – Genesys is releasing a simplified version of our JIRA-to-wiki solution, so that other groups can use the solution for themselves.

The open source solution is composed of two parts:

- JIRA Plug-In
- MediaWiki Extension

JIRA Plug-in

This is the core part of the solution. Leveraging a new “Release Note” field in JIRA, the plug-in lets you as a writer gather issues and submit them to an interface page, where you can then make the following choices before pushing to your publisher:

- Classify each issue as a new feature, correction, or known issue.
- Select “write to wiki” to automatically write to your MediaWiki instance, or “raw HTML” to grab unformatted HTML code that you can then use in your own CMS.

MediaWiki Extension

This is a small MediaWiki extension that you can add to your MediaWiki installation. It lets the wiki accept the content pulled from the JIRA Plug-in and automatically write that content to straight MediaWiki pages.

You can use the JIRA Plug-in and related MediaWiki extension to write Release Notes text to your wiki out of the box. However, the goal of releasing these open source components is to provide sample applications that you can use as models when setting up your own solution. Many teams use JIRA, but since there are so many front-end CMSs to choose from, this sample is intended to get your

developer started with setting up a JIRA Plug-in that connects to your own team's CMS.

Conclusion

I remember a week during the first year of my technical writing career when I had to prepare 13 Release Notes. Most of the releases that week were so-called Hot Fixes, meaning that the Release Notes had to be turned around almost immediately. That was a long and stressful week, which I regard to this day as the "writer's hell". I wish Genesys had had the current technology and online Release Notes solution in place back then.

Since June 2014, when two product teams conducted a pilot of the automated system for production of online Release Notes from JIRA, the Genesys Technical Publications team has moved Release Notes for over 20 products to the new online format. As of April 15, 2016, writers have published Release Notes for over 550 major and Hot Fix releases for 80 distinct software components. Writers who work on legacy products also benefit from using the JIRA RN Plug-in and process for HTML Release Notes automation.

With our actions taken to make the JIRA RN Plug-in available as open source, we trust that technical writers and documentation department managers from other software companies will find the tool and processes equally valuable for their teams.

Author Contact Information

Tanya Ivanova
 Senior Manager, Analytics and Business Applications
 Genesys
 2001 Junipero Serra Blvd.
 Daly City, CA, USA, 94014
 +1 650.713.9127
 tanya.ivanova@genesys.com

Barry Grenon
 Senior Manager, Information Architect
 Genesys
 50 Smythe Street
 Saint John, NB, Canada, E2L 0B8
 +1 506.645.1310
 barry.grenon@genesys.com

Author Biography

Tanya Ivanova has over 15 years of experience in technical writing for software companies. She is currently a Senior Manager with the Technical Publications department at Genesys Telecommunications Laboratories. Tanya has lead the initiative for automating production of Release Notes at Genesys, from creating the solution architecture, to directing the platform developer, to working with JIRA administrators on dependencies, to creating training on the new tool and processes, to promoting the solution among writers and product teams.

Barry Grenon is a Senior Publications Manager and Information Architect at Genesys, a customer experience software company recently included in GlassDoor's Top 50 Places to work. Barry leads the team that builds and maintains their custom documentation platform, docs.genesys.com. He is a long-time technical writer, with a background in creative writing, including an MFA and publications to his credit from that other life. In tech-comm, Barry's interests are in content engineering and UX and how the two can work together.



PROGRAMMING LANGUAGE STORYTIME

Sarah Kiniry

Contrary to common opinion, not everyone needs to write code to be able to read it. “Digging into” the code is often the best way to extract information when writing software documentation, but the idea of reading all that code can be daunting. However, a few simple tricks and an understanding of the patterns involved can allow technical communicators to get started reading code in no time.

Communicating about software, whether for user-oriented instructions, developer-oriented API documentation, or interactive walkthrough content, is a huge part of the technical communication industry. More often than not, the technical communicator on this type of project acts as a de facto translator between software developers and the rest of the world.

To craft this communication effectively, technical communicators need to learn how to gather information independently, without constant input from the developers themselves. The logical answer to this problem, of course, is to dive into the software itself to gather information and develop a strong vocabulary in the developer’s native language, code. Too often, this is where a disconnect arises.

Reading Is Not Writing

When faced with a screen full of text in an unfamiliar language, the learning curve is so daunting that many simply decide not to bother. Those who do persevere often spend long hours learning how to be a competent developer even though they do not really need that skill set.

We often forget a very important factor in the equation: reading is not the same as writing. Technical communicators do not always need to become skillful, or even novice, developers. Instead, they need to become adept readers of the appropriate programming language. As with any form of communication, being a talented writer is not the same as being a good reader. A tourist in Rome probably does not need to be able to write an essay in fluent Italian in order to read the city’s signage and find the right streets. Similarly, looking for answers in code does not require the

ability to craft it. If you can learn to read, you can learn to read code.

With the skills to answer their own questions, technical communicators can often operate independently from developers to gather their own information. Three questions are all that a reader needs to gather information from code and find the “story” that the code tells:

- What does the application do?
- What text will the user see?
- Which files or settings are involved?

Technical review by subject matter experts will always be a part of the writing process. However, bringing them into the process later, to review existing content rather than fill in myriad informational gaps, saves everyone time. It can even help to forge stronger relationships between team members. In Agile environments, this modified flow of information allows documentation writers to escape from all-too-common miniature-waterfall scenarios and operate alongside the team.

What Does the Application Do?

Writing a computer program is merely writing a set of instructions, not unlike a recipe. By extension, reading code allows one to figure out what the final product will be, based on those instructions. (Wang, 9) The most important question for a technical communicator to answer is, “What does this application do?” Answering this one question generally provides most of the necessary information. The application’s action will also dictate which other questions should follow, depending on the

audience. If an application stores configuration information to a database, the end user might need to know the specifics about the information stored, but not how the program writes it to the database. Conversely, if writing for a highly technical audience, those database details may be far more important than the individual settings.

What Text Will the User See?

The text that users see in graphical interfaces, or even in a command line interface, is always an important part of communicating about any application. User text provides clues for the code's reader just as it does for the end user. It can also serve as a signpost to locate other elements in an interface. For example, knowing that a button displays with a certain label, or knowing that a certain warning message displays after a user action, could allow the reader to more easily find that place in the code.

In many cases, identifying this text will also allow writers to update and improve those messages during the development process. Looking for this type of text may also reveal the need for labels or help text that are missing in other areas of an interface, improving user experience.

Which Files or Settings?

Files and settings are also useful clues, particularly when identified alongside the action that uses or affects them. A setting's name within the code often indicates its function. The fact that a setting saves to a specific configuration file might indicate more about the setting than its name alone. Additionally, finding the locations involved in code allows you to track down and examine related settings or sections of code. In complex applications, developers almost always spread functionality across many files.

It may seem surprising to anyone new to reading code, but the more familiar one becomes with a specific codebase and the style that a specific group of developers uses, the easier it is to draw helpful inferences from these conventions. An avid reader of a codebase might quickly learn, for example, that any new settings in a particular configuration file will correlate to a setting in a specific user interface. In products that include an externally-accessible API, knowing the specific files that contain those API functions means the reader can quickly "weed out" the need to investigate new functions in other files, which will be for internal use only.

Clues for Reading Code

When looking at code, questions can be answered through looking at comments, function and setting names, and variables and their values. The entire story of the code's actions, from start to finish, is all there waiting to be read. Because modern computers can use "high-level" programming languages, which more closely resemble human languages, reading code does not mean reading long lists of binary 1s and 0s (Wang, 15).

The key to finding these answers is to remember that a large portion of the code can often be "glossed over." In many cases, recognizing a few key clues is enough to piece together the puzzle, without understanding every single character of the code. For example, in a user interface that contains a password field, understanding the specifics of how the page validates that password against a database of users is not likely to be important, but recognizing that the password field exists and uses a specific label is.

Of course, because so much of the programming process is at the discretion of the developer, the usefulness of these items depends on the specific developer who wrote the code. Some developers include detailed comments and helpful, human-readable function and setting names. Other code, in comparison, may be quite obscure to all but the most skilled developers. Sometimes, though, a few nudges from readers of code is enough to encourage developers to find their way back to best practices and more helpful, human-readable code.

The clues to use to find an application's "story" can be defined in three broad categories:

- Comments, the human-readable notes that developers include in code for their own use.
- Functions, the reusable actions that can be defined by either developers or the programming language itself.
- Variables, the settings and values that the code uses to run, or makes changes to.

Comments

Every programming language uses a set of special characters ("comment characters") that set descriptions or reference text apart from the actual code. The computer does not use these sections of code, skipping over them when the application runs. Instead, they remain in the code for developers to read when they make changes. These comments can act as developers' to-do lists, notes about

harder-to-read code, or perhaps justifications of particular choices that the original developer made.

```
// Set a string.
var str = "This is a string of text.";

// Retrieve only the word "string"
var res = str.slice(10,16);
```

Figure 1. Comments within simple JavaScript code.

Developers also use comment characters to “comment out” code, causing the computer to temporarily ignore a certain section of code without the need to delete it from the file and then add it again later. While no programming language requires these comments to be present for a functional program, they are an important “best practice” that most developers almost always follow (Wang, 277).

- For readers of code, these comments can act as a roadmap to the code’s actions. Some applications even include full in-code documentation to describe the application. In most programming languages, two styles of comments exist: single-line comments and multi-line comments.

//	Java, JavaScript, C, C#, PHP
#	PHP, Perl, Ruby, Python

Figure 2. Single-line comment characters in popular programming languages.

- Single-line comments, sometimes referred to as “line comments,” use a comment character at the beginning of the comment. The comment ends at the line break, and the computer will begin to use the code normally at the beginning of the next line. Single-line comments often begin on the same line as code, noting information about that particular line.

/* Comment */	Java, JavaScript, C, C#, PHP
// Comment \n Comment	C
=pod	Perl
=cut	
=begin	Ruby
=end	
""" """	Python

Figure 3. Multi-line comment characters in popular programming languages.

- Multi-line comments, sometimes referred to as “block comments,” use comment characters to mark the beginning and end of the comment. The computer ignores everything between the two comment characters, even if that section includes otherwise-functional code. Developers generally use this style of comment for larger sections of in-code documentation, or to comment out large sections of code.

Functions

The term “function” can refer to any reusable code that performs one or more actions. In many programming languages, however, the term for these actions may differ; “subroutine,” “method,” “subprogram,” and “procedure” are just a few of the other terms that can describe variations on this concept.

The names of functions within code can act as keywords to give clues about the code’s actions and the information on which the actions are performed. A function might be an externally-usable API function, or it could be completely internal. All programming languages include at least some pre-determined functions, such as the ability to open and read files, or the ability to display text to a user. They may also contain predefined control structures, which allow for certain actions to take place only under certain conditions. For example, in Perl, the “while” control structure causes the specified action to be repeated as long as a certain condition exists (Schwartz, 37).

```
// Count down from 10.
my $number = 10;

while ($number > 0) {
    print '$number...';
    $number -= 1;
}
print 'Countdown complete!';
```

Figure 4. The “while” control structure in Perl.

Developers can also craft their own reusable code to allow them to perform specific actions multiple times without the need to write the full code for the action each time. These custom functions act as miniature programs within the larger application (Wang, 132).

```
def my_fn(name)
    var = "Howdy, " + name
    return var
end
```

Figure 5. Defining a function in Ruby.

When the code creates, or “defines,” a function, a delineated section of code contains the function’s actions. Many languages, included in the “curly brace family” of languages, use curly braces to delineate these sections (Wang, 63). Other languages, like Ruby, use other methods to mark the end of the function. While the format used by each programming language varies, function declarations generally perform several basic actions:

- Naming the function.
- Accepting input data.
- Performing one or more actions on the input data, or using it to modify other data or files.
- Returning a response, which might include data or a message of success or failure.

```
sub do_something {
  my $value = @_;
  if $value {
    return "Yay!";
  }
  ...
do_something($value);
```

Figure 6. Defining and then using a Perl function.

When code uses a predefined function, it only includes the function name and any data that the function will use. Again, the specific formatting for each programming language varies, though not as widely as for function definitions.

```
# Print a string to the user.
print("Welcome to my Python program!")
```

Figure 7. A print function in Python.

All programming languages include at least one predefined function that displays text to the end user. Learning to identify a programming language’s display (or “print”) functions can also make it easier to find user text within the application code. To identify user text, the best method is to learn to identify the programming language’s built-in print or display functions. Some applications may also use proprietary output functions for this purpose. For example, if an application uses a localization system to allow translation of its user interface, it may be important to learn how to recognize and read the functions for that localization system.

Variables

Often, functions accept data as input and perform actions on that data. Functions may also include hardcoded data, or may pull data from other locations, like configuration files, databases, or even the output of other functions. Variables store each piece of data that the code will use, allowing programs to hold as many pieces of data at once as the program needs (Wang, 142). Like functions, many terms exist to describe different types of variables, and some programming languages prefer certain terms over others. Variables may represent a single piece of data, an array of many individual values, or a hash, which contains multiple key and value pairs.

<code>variable_name</code>	Java, JavaScript, C, C#, Python
<code>\$variable_name</code>	PHP, Perl, Ruby
<code>@variable_name</code>	Perl, Ruby
<code>tvariable_name</code>	Perl

Figure 8. Variables in several popular programming languages.

The names of these variables give more information about the data on which the application operates. For example, a variable with the name “filename” probably stores a file’s name. Some styles of development even use structured naming conventions, such as Hungarian notation, in which the names of variables (or even functions) include abbreviated information about the type or purpose that precedes the programmer-selected name.

Files	example.txt, example.jpg, example.php
Directories	Linux: /example/directory or example/directory/ Windows: C:\example\directory or .. \example\directory\
Settings	managed, unmanaged; blue, green, red; 0, 1

Figure 9. Examples of important value types.

Variables can store three broad types of values:

- Text values, which include any number of characters (Schwartz, 24).
- Numbers, including integers and decimal values.
- Boolean values, which are technically integer values, but use those numbers to store values that mean “yes” or “no” (Wang, 149).
- Which values provide you with the most information depends largely on the type of software. In many applications, the files or directories that the code operates on will give you many details.

For example, a file path may help you find the exact images that users will see or the specific configuration files that the code uses to retrieve or save settings. In other applications, settings that are specified in the code may be more important, like Boolean values to show whether or not a user has selected certain preferences, or variables that differentiate behavior between different user account types.

Getting Started With Hello World

When learning to read a specific programming language, one of the best places to start is with a “Hello World” script. “Hello World” is a sort of “See Spot Run” for code. Almost any beginning programming course will start with one of these. In every language, the story is the same. The script simply performs whatever actions are necessary to display the text “Hello World” to the user. Because the story is the same each time, the reader already knows the answers to their questions, which allows them to focus on recognizing the patterns for that programming language:

1. What does the application do? It displays a message to the user.
2. What text will the user see? “Hello World”
3. Which files or settings? No other files or settings are involved.

```
/* A Hello World script. */

void main()
{
    printf("Hello World!");
}
```

Figure 10. An example of a “Hello World” script in C#.

Of course, even a simple action can be performed in several ways in a single programming language. In fact, the idea that “there’s always more than one way to do it” is the philosophy of the entire Perl community (Wang, 559). Because of this, it’s always beneficial to find additional examples of simple scripts, to see as many of the variations as possible.

Resources

“C Programming Tutorial.” Programiz (2016). <http://www.programiz.com/c-programming>.

“C# Programming Guide.” Microsoft Developer Network (2016). <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>.

Harrington, Andrew N. “Hands-on Python Tutorial.” Computer Science Department, Loyola University Chicago (08 September 2014). <http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/index.html>.

Heddings, Lowell. “Ruby Function (method) Syntax.” How-To Geek (2016). <http://www.howtogeek.com/howto/programming/ruby/ruby-function-method-syntax/>.

“JavaScript.” Mozilla Developer Network (2016). <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.

“JavaScript Tutorial.” W3Schools (2016). <http://www.w3schools.com/js/default.asp>.

“PHP 5 Tutorial.” W3Schools (2016). <http://www.w3schools.com/php/default.asp>.

“PHP Manual.” The PHP Group (2016). <http://php.net/manual/en/index.php>.

Ponnappa, Sidu and Jasim A Basheer. “Ruby Primer.” RubyMonk (2012). <https://rubymonk.com/learning/books/1-ruby-primer>.

“Trail: Learning the Java Language.” Oracle Java Documentation (2015). <https://docs.oracle.com/javase/tutorial/java/index.html>.

References

- Schwartz, Randal L., brian d foy, and Tom Phoenix. Learning Perl (Sebastopol, CA: O’Reilly), 2011.
- Wang, Wallance. Beginning Programming All-In-One Desk Reference (Hoboken, NJ: Wiley), 2008.

Author Contact Information

Sarah Kiniry
 Technical Writer II
 cPanel, Inc.
 2550 North Loop West Ste 4006
 Houston, TX 77092
 +1 713.742.3617

Author Biography

Sarah Kiniry is a technical writer for cPanel, Inc., a web hosting software company. In this role, she provides full documentation support for an Agile development team, as

Technology and Development

well as participating in content strategy and other initiatives for the company as a whole.

In a past life, Sarah was a stage manager and audience development manager at several of Houston's professional theaters and performing arts venues. She believes that technical communication is really just another performing art, and that documenting an interface isn't that much different from telling any other story.

You can read more about Sarah's journey into software documentation at technicolorwriter.com.



REST API DOCUMENTATION BEST PRACTICES

Marta Rauch, Fellow

When documenting REST APIs, satisfy developers by understanding their needs and incorporating documentation best practices.

Technical communicators can enhance developer user experience with REST APIs by incorporating the best practices described in this session.

What Are REST APIs?

APIs, or Application Programming Interfaces, provide communications between products and services. Examples include cloud services, mobile and wearable apps, and products connected by the Internet of Things (IoT). A useful definition is provided by Programmable Web: <http://www.programmableweb.com/api-university/what-are-apis-and-how-do-they-work>.

REST stands for REpresentational State Transfer, an architecture for client-server web communication. REST APIs are fast, scalable, and reliable. They use HTTP methods such as GET, POST, PUT, and DELETE.

Opportunities for Technical Communicators

According to research by Programmable Web, finance and enterprise are the fastest growing API categories, and the number of REST APIs is increasing dramatically. As Laura Heritage has observed, up to 90% of new APIs are REST APIs. The rapid proliferation of REST APIs brings opportunities for technical communicators who create useful content that incorporates best practices and satisfies developer users.

Documentation Best Practices

Developers who use REST APIs have unique requirements. On the one hand, it is critical to provide complete and accurate information. On the other hand, API documentation needs to be simple enough to help new developers quickly get up to speed.

Best practices for REST API documentation can be summarized as follows:

- Help developers get started quickly
- Include useful reference information
- Provide sample code
- Provide a list of REST endpoints
- Represent resources with a description language
- Provide a modern, usable UX

To get developers started quickly, they need to know:

- Why they should use the API
- The base URL
- How to get an API key
- How to handle authentication
- Error handling
- HTTP status codes
- Scenarios and use cases

It is essential to include necessary reference information to help developers use the REST API. To make larger docs more readable, group topics into categories. Reference information should contain the following information:

- Name

- Method (such as GET, POST, PUT, or DELETE)
- URL structure
- Overview
- Parameters
- Descriptions
- Remarks
- Examples
- REST endpoints

Reference material should include a list of REST endpoints to give developers an alternate way to access information. Think of this like a table of contents that developers can scan to see what is available and go quickly to a topic of interest.

A key best practice is to include sample code to help developers understand real-world applications. Provide example requests and responses and quick access to request and response information. Samples should be in the programming languages that your customers use. Be

sure to test code samples before including them in the documentation.

It is also useful to represent resources using an API description language. This allows documentation to be generated and automated. Example frameworks include the Open API Initiative (OAI), formerly called Swagger; RAML; and JSON Schema Hypermedia.

Finally, provide a usable, modern interface and user experience (UX). An Interactive web interface enables quick access to resources and examples. As with any user interface, it is important to know your target users, develop early prototypes, and incorporate user feedback. The following example incorporates developer input and is an example of an interface that satisfies its target users: https://docs.oracle.com/cloud/latest/jcs_gs/JSRMR/

Conclusion

REST APIs have unique requirements for satisfying developer users and getting them quickly up to speed with their tasks. To ensure a positive user experience, it is important

ORACLE Help Center Welcome Marta ▾

REST API for Oracle Java Cloud Service

Table of Contents

- About the REST APIs
- All REST Endpoints

Get Started

- Quick Start
- Send Requests
- Authentication
- Use cURL
- Status Codes
- Use Cases
- Context Path Deprecation Notice

Tasks

- Backup Configuration
- Backups**
- Cloud Tools
- Coherence Data Tiers
- Health Monitoring
- Managed Servers
- Patches
- Restorations
- Scaling
- Service Instances

Tasks / Backups

Backups REST Endpoints

Sort by **Task** Path Method

Back up your Oracle Java Cloud Service instances.

Task	Method	Path
Archive or Download a Backup	POST	/paas/service/jcs/api/v1.1/instances/{identityDomainId}/{serviceId}/backups/{backupId}
Delete a Backup (Asynchronous)	POST	/paas/service/jcs/api/v1.1/instances/{identityDomainId}/{serviceId}/deletedbackups
Delete a Backup (Synchronous)	DELETE	/paas/service/jcs/api/v1.1/instances/{identityDomainId}/{serviceId}/backups/{backupId}
Initiate an On-demand Backup	POST	/paas/service/jcs/api/v1.1/instances/{identityDomainId}/{serviceId}/backups
View a Backup	GET	/paas/service/jcs/api/v1.1/instances/{identityDomainId}/{serviceId}/backups/{backupId}

Figure 1. Example of a Modern, Usable Interface

to understand your target users and develop a suitable content strategy. Use the best practices described here to create useful REST API content that helps your developers accomplish their tasks.

For more information on this session, see the slides on SlideShare: <http://www.slideshare.net/MartaRauch>

Resources

Marta Rauch's presentations and posts:

- SlideShare, <http://www.slideshare.net/MartaRauch>
- Pinterest boards on REST APIs <http://pinterest.com/martarauch/>

STC Intercom API issue: <http://intercom.stc.org/magazine/september-2014/features-september-2014/>

STC webinars:

- Johnson, Tom. Best Practices for Documenting REST APIs. http://www.stc.org/education/online-education/recorded-seminars/item/interviewing-smes-preparing-conducting-and-polishing-2-copy?category_id=88
- Maddox, Sarah. Intro to APIs. http://www.stc.org/education/online-education/recorded-seminars/item/part-1-in-api-series-introduction-to-api-technical-writing-2?category_id=88
- Malin, Joe. Helping the Code Tell the Story: How to Present Effective Code Samples. http://www.stc.org/education/online-education/live-seminars/item/part-3-in-api-series-helping-the-code-tell-the-story-how-to-present-effective-code-samples?category_id=53
- Rauch, Marta and Bs, Guru. REST API Overview and Documentation Best Practices. http://www.stc.org/education/online-education/recorded-seminars/item/rest-api-overview-and-documentation-best-practices-2?category_id=88
- API Webinar Recording Series Bundle. http://www.stc.org/education/online-education/recorded-seminars/item/api-webinar-series-recordings-bundle?category_id=88

Gruenbaum, Peter. SDK Bridge online courses:

- API Documentation 1: JSON and XML for Technical Writers. <http://sdkbridge.com/online-courses/>

- API Documentation 2: REST for Writers. <http://sdkbridge.com/online-courses/>
- API Documentation 3: The Art of API Documentation. <http://sdkbridge.com/online-courses/>

Maddox, Sarah. Intro to API Technical Writing. <http://www.slideshare.net/sarahmaddox/api-technical-writing>

Maddox, Sarah. API blog posts. <https://feathers.wordpress.com/>

Johnson, Tom. API documentation posts: <http://idratherbewriting.com/category/api-documentation/>

Johnson, Tom. Documenting REST APIs online class. http://idratherbewriting.com/docapis_course_overview/

References

Programmable Web API Research. <http://www.programmableweb.com/api-research>

REST API definition, Wikipedia. https://en.wikipedia.org/wiki/Representational_state_transfer

Lane, Kim. API Documentation, API Evangelist. <http://documentation.apievangelist.com/>

Heritage, Laura. RAML: The Birth of an API Description Language Fit for the Enterprise. <http://blog.soa.com/raml-birth-api-description-language-fit-enterprise/>

Heritage, Laura. API Description Languages: Which Is the Right One for Me? <http://www.slideshare.net/programmableweb/laura-heritage-api-description-languages-which-is-the-right-one-for-me-see-more-at-httpwwwapiconukcomapiconukschedulesthashz8brx9ykdpuf>

Gruenbaum, Peter. The Five Biggest API Mistakes and How to Avoid Them. http://www.prestwoodboards.com/ASPSuite/KB/document_view.asp?qid=102081&GroupID=

Bleyle, Jody and Rondeau, Jennifer. Documenting REST APIs. https://docs.google.com/presentation/d/1jejYiTagK7CnJ-ajiRO1-kbD6kzeA0R04o3W5_yKTVk/edit#slide=id.g3782ffb8_1_5

Srikishnan, Yogeshwar. REST API Documentation Part I. <https://dzone.com/articles/rest-api-documentation-part-1>

Srikishnan, Yogeshwar. REST API Documentation Part II. <https://dzone.com/articles/rest-api-documentation-part-2-1>

Technology and Development

Mitchell, Lorna. Documentation First: A Recipe for API Success. <https://dzone.com/articles/documentation-first-a-recipe-for-api-success>

REST API for Oracle Java Cloud Service. http://docs.oracle.com/cloud/latest/jcs_gs/JSRMR/

Author Contact Information

Marta Rauch
Senior Principal Information Developer
Oracle
@martarauch

Author Biography

Marta Rauch is a senior principal information developer at Oracle, where she leads mobile, cloud, and REST API projects. Marta enjoys participating in design jams and developer challenges at her company to prototype solutions for Internet of Things, big data, augmented reality, mobile, and wearable technology.

A frequent presenter for conferences and webinars, Marta has published articles for IEEE, HCII, STC Intercom, and CIDM Best Practices. She contributed information to [Developing User Assistance for Mobile Apps](#), by Joe Welinske. Her wearable topic will appear in [The Language of Technical Communication](#), by Ray Gallon and Richard Hamilton, and her augmented reality topic appears in [The Language of Content Strategy](#) by Rahel Bailie and Scott Abel.

An STC Fellow, Marta has received 15 STC awards for individual and team projects at the regional and international level. She is VP of the Silicon Valley chapter and a member of the Nominating Committee. She is Tools & Development track manager for Summit 2016 proposals, and judges International Summit Awards. Marta holds a degree from Stanford University and certificates from Notre Dame De Namur and University of California.



GETTING CREATIVE WITH TECHNICAL COMMUNICATIONS

Liz Roscovius

Whether it's a contract, report, or documentation, technical communication pieces aren't always fun to read. They fill a purpose but it's sometimes limited in scope or considered a "necessary evil" to avoid personal injury and lawsuits. Sometimes they're a last resort that people reference only because they have problems or questions. This begs the question: How do you make your audience want to use your communications? In this progression, we'll introduce audience analysis and information design strategies to help you make communications pieces both enticing and easy to understand.

Some communication pieces just aren't exciting. Luckily, there are a variety of things you can do to make communications less painful for an audience. At the end of this presentation, attendees will understand:

- Ways to better understand their audience
- The importance of considering the entire communications life cycle
- Basic design principles to improve comprehension

This presentation will cover strategies and offer practical examples that show how to make communications pieces more intuitive, visually appealing and, ultimately, more user-friendly.

Understand your audience

The first step toward getting creative with documentation is to understand the target audience. Put yourself in their shoes. Are the people who buy your product or service the same people who are using it (and reading documentation)? Performing an audience analysis may already be part of your standard process but it's important to revisit it from time to time in order to meet changing customer needs. Take a step back and make sure you haven't missed something.

Is the document for technical people or average consumers? Think about why they might not want to read/interact with your communication piece so that you can take steps to fix/alleviate it. How do they want to consume information, e.g., do they prefer videos or visuals?

One resource might be coworkers in your immediate or other departments, such as sales, marketing, and tech support. If they have access to customer demographics, support call data, HTML analytics, customer reviews, or field service logs, they may be able to offer useful information about who you are writing for. Has someone else worked in or with your employer or client longer than you? Draw on their experience and knowledge to gain important insight.

Get another set of eyes on a rough draft. What do others think? If you have access to a usability or human factors team, get their input. Their primary focus is customers so their insight could be very helpful. Is it possible to test with customers? Talk to them! Getting direct input can yield tremendous insight.

Other resources might include:

- Company website—see what it says, directly or indirectly, about its customers and how it communicates to customers.
- Social media—note which platforms are utilized. Sometimes, you can draw some inferences based on that. Example: Linked In posts may

Technology and Development

indicate a B2B premise and a slightly reserved model. Twitter might also be B2B but indicates a more casual approach. Pull ideas from videos, if available.

- Online trade publications
- Databases e.g., Forrester research, Pew Research
- Competitors' communications collateral, such as website, brochures, marketing materials
- Brand guidelines—sometimes contain a wealth of information on target audience

Incorporate Key Design Principles to Improve Comprehension

For many reasons, technical and business documents sometimes have more restrictions and limitations than those of a marketing nature. There may be usability and liability considerations that limit available options. Even with the strictest standards, however, there are usually one or two tactics one can take to help make a document more engaging and effective.

Information Architecture

Start by thinking about the overall communications strategy. If possible, offer information in separate communications pieces, designed to meet specific goals. The goal should be to provide what customers want, when they want it and how they want it. Taking this approach will break information into bite-sized morsels for easier, more digestible consumption. Are you presenting information that may be confusing or hard to understand? Think about alternative form factors, such as an online version that auto-fills/completes a form. This is where the best opportunity lies to apply what you understand or have learned about the audience.

Next, critically evaluate overall organization of each document. Look at it through the eyes of the target audience. Make sure every section necessary to the intent and purpose of the document. Split long sections into shorter ones or combine short, related sections, where possible. If the document is a long one, consider incorporating “chapter or section” pages to help break up expansive text. They don't have to be elaborate works of art. A solid color, pattern, photo, or relevant graphic can be very effective.

If document guidelines allow it, consider both landscape and portrait layouts. Sometimes, that alone can make a

huge difference. Depending on what you started with, you might also gain additional real estate.

Sidebars are a great way to emphasize key information while incorporating white space. Do not feel compelled to fill it, however. This design component loses its effectiveness if overused.

If animated GIFs or embedded video are an option, consider adding visual links to those in the document. Doing so will help reduce the amount of text while offering valuable information and more reader engagement.

Language and Writing

Customers don't typically want to waste a lot of time trying to find desired information so be as clear and concise as possible. Get to the point. Keep minimalism in mind. Think about how and where content can be simplified. Aim for the appropriate reading level and test frequently to see where you are at. Use commonly-used terms, where possible, and avoid technical jargon. Don't say something in 15 words if you can say it in ten. Avoid passive voice and unnecessary adjectives.

Keep sentences short and limited to a single idea. Break up long sentences, where possible. Make sure it reflects desired brand image. Consider voice and tone throughout.

Design

There are a number of design elements, including text and graphic treatments, that can help make technical communications pieces more user-friendly.



Figure 1. Photos, logos, color, and other elements can breathe life into what could be a dry document.

For example, choose typography that meets brand requirements yet is clean and easy-to-read. Be sure to keep online restrictions in mind. Be creative with text styles, if guidelines allow it. Capitalizing, italicizing, and bolding carefully-selected text can dramatically improve readability.

Add color, if possible. Used sparingly and with thought, color can have a tremendous impact by drawing attention to select elements.

Add or enlarge graphics, where possible and appropriate, but don't do it just for the sake of doing it. If a graphic helps tell the story, let it take the stage and have text play a supporting role. Even small, inline icons can help break up a large amount of text. Subtle and strategically-placed watermarks can help as well.

Don't be afraid of white space. Look at column width, gutters, margins, line spacing, and character spacing to see where you can add some air. Increase leading and kerning for easier-to-read text and user-friendly white space. Impact on a single page may appear to be negligible but, applied to an entire document, can have a huge impact. Even the smallest of tweaks can work to create a document that breathes and, as a result, appears more approachable.

Other Considerations

Collect examples of communications pieces you consider particularly compelling. Collecting ideas doesn't have to

be a special, one-time effort. We consume a ton of media on a daily basis. You may need to "retrain" yourself to recognize the things you see every day as possible ideas. Resources might include:


- Check competitive sites for online communications. You can do this with any type of communications piece – user manual, annual report, case study, sell sheets, and video.
- Check with internal sources – is there an idea management program in place? If not, is it feasible to start one and start collecting those ideas?
- Google it – you can learn a lot from simple text or images searches
- Maintain a personal idea file and use it as a resource for new, innovative ideas.
- Leverages professional memberships, e.g., check with forums, ask peers for ideas, etc.
- Check out online videos for ideas, inspiration

Keep accessibility in mind, e.g. utilize alt text, video transcripts, and styles that help users better comprehend key information. Consider alternate output file formats. PDFs, for example, are not very accessible for some. HTML might be a better option.

Be flexible. There will be times you can't implement your best ideas. When that happens, keep an open mind. Hang on to those ideas because there may come a time when you can use them. Remember that times are changing.

Pairing your speaker with a device

1. Turn ON your speaker. A tone sounds, the LED blinks quickly, and your speaker enters pairing mode.


Note: If you do not hear a tone and the LED doesn't blink quickly, press and hold the function  button for 1 to 3 seconds to put your speaker in pairing mode. A tone sounds and the LED blinks quickly.

2. Turn ON your Bluetooth device (like a phone or tablet).
3. On your device, locate your Bluetooth settings and turn on Bluetooth.
4. In Bluetooth Settings, scan for devices if necessary, then select **Amazon_BT2** from your device's list. The LED stops blinking and lights blue. Your device and speaker are paired and your speaker is ready to use.

5

PAIRING YOUR SPEAKER WITH A DEVICE

- 1 Turn ON your speaker. A tone sounds, the LED blinks quickly, and your speaker enters pairing mode.

NOTE: If you do not hear a tone and the LED doesn't blink quickly, press and hold the function  button for 1 to 3 seconds to put your speaker in pairing mode. A tone sounds and the LED blinks quickly.

- 2 Turn ON your Bluetooth device (like a phone or tablet).
- 3 On your device, locate your Bluetooth settings and turn on Bluetooth.
- 4 In Bluetooth Settings, scan for devices if necessary, then select **MBT_100** from your device's list. The LED stops blinking and lights blue. Your device and speaker are paired and your speaker is ready to use.

5

Figure 2. Before (left) and after (right): Minor text treatments, such as expanded character and line spacing, introduce white space into what might otherwise be dense text.

Technology and Development

There is a new generation of consumers that is put off by lots of text so less can oftentimes be more.

In the meantime, focus on what you can do. Take a minimalist approach to writing and incorporate basic design elements that are feasible. Throughout the entire process, keep the audience and communications goals in mind. In the end, even small efforts can work together to create a communications pieces that are both enticing and easy to understand.

Resources

Roscovius, Liz. "Getting Creative with Technical Communications." (Proceedings, STC Summit Conference, 2016). <http://www.slideshare.net/lizRoscovius/>

Linked In Technical Writer Forum
<https://www.linkedin.com/groups/112571/112571-6121746783463890946>

Author Contact Information

Liz Roscovius
Director of Global Communications
D2
P.O. Box 644
North Sioux City, SD 57049
+1 605.232.4170 x117
liz.roscovius@d2worldwide.com
@LizRoscovius

Author Biography

Liz Roscovius has worked in the communications field for nearly 30 years. As Director of Global Communications at D2 (Documents & Design), she works with customers around the world, in fields ranging from consumer electronics and furniture to financial to industrial, to provide creative and technical communications support. Her interests include developing effective, user-friendly documentation on time and within budget, with particular focus on document design, content strategy, branding, and project management. She manages a team of technical writers, editors, and graphic designers.

Liz holds BAs in Art and Mass Communications, from the University of Wisconsin-LaCrosse and Briar Cliff University in Iowa, respectively, as well as an MA in Journalism/Mass Communications, with an advertising and marketing focus, from the University of Nebraska-Lincoln. She has won more than 40 awards for design, layout, writing, and graphics in STC and other competitions.



LET THE FORCE OF APIs BE WITH YOU: BUILDING THE DEVELOPER EXPERIENCE RIGHT

Nabayan Roy

Today, we are living in an API economy, if I can say so. Everything is slowly moving into a new paradigm, which is service. Product companies are slowly, but surely gearing up to this change. And with the advent of cloud computing and its rapid adoption; it's an open war and APIs are your Lightsabers in this war. APIs have become the most attractive intellectual and digital asset—the growth and penetration of the Web and the gradual shift in mindset of companies from being product-centric to being service-oriented has also seen an increase in the API traffic.

When you create an API, you are inviting developers to become your customers. Even if your API is exposed only internally or to selected partners, your users are developers. And just as you spend the time, money, and effort to provide the best user experience through user interface design and interaction; you should expend the same level of time and effort to provide the best possible developer experience through API design and onboarding process.

The Rise of Web APIs

Over the past five to ten years we have seen a massive increase in the availability of cloud-based solutions. We now listen to music via Spotify, watch streaming movies on Netflix, and store our photos, files and memories on Facebook or Dropbox. As this trend moves from the consumer to the enterprise, there will be a fundamental shift in the way systems are designed and operated. Analysts predict that within a few short years there will be some 50 billion devices connected to the internet.

The API is the glue and the enabler that allows organizations to leverage the opportunities inherent in these new approaches, react to constraints and externalities in their broader environment, and deliver innovation, agility, and ongoing business sustainability.

APIs are not just the latest high tech product, they are the next evolution of the Internet. In 1995 companies were trying to decide whether or not they needed a website, and by 2000, having a website became critical for almost all aspects of business. In 2010 companies were trying to decide whether or not they needed an API, and in 2015, having an API is becoming critical to doing business in the rapidly expanding API economy.

What is Developer Experience?

According to Wikipedia, user experience refers to a person's emotions and attitudes about using a particular product, system or service. Therefore, we can say that user experience applied to developers can be termed as developer experience. Another definition says, "It is the sum of all interactions between the developer and an API."

When you start monetizing, you need to ensure that your APIs provide an end-to-end experience for the developer. What hurts developers is when they see nothing more than a link that points to a new API. Now that is a bad developer experience! Great developer experience is not just a great API design—you will need documentation, SDKs, sample apps, forums, interactive tutorials, debugging tools and everything else that goes into a creating a great "developer experience". The developer community needs to be handheld first, and then slowly you let them walk alone and take wings. Documentation is the single most important resource for developers.

Guiding Principles for Developer Experience

Designing any experience means closely understanding those who will undergo the experience. Developers are heterogeneous, hence you must tailor your experience

based on the type of developer. The first step is engaging the target audience by creating something they will want to experience. The ability to understand the feelings and thoughts of your developers and to be able to effectively place yourself in their shoes forms a critical piece in building the right developer experience from the ground up. 90% developers spend almost 40% of their time on reading the documentation, so it's important to provide an experience that makes using APIs fun, easy, and rewarding. Developers are hooked on to your APIs because of the end-to-end experience and the support they get; they come to fall in love with such an organization. Another important aspect when building the developer experience is around the principles of simplicity—when it comes to designing your service or your homepage, remove clutter and make it as easy as possible for your developers to achieve their desired outcome. And since we are talking about Web APIs and developer experience, we should always give importance to the trust factor in our design thinking. Before asking your developers to complete an action, make every effort to help them understand why the task is needed. Being honest and clear in explanations builds trust at each step, leading to increasingly easier conversions down the funnel. Last, but not the least is the trajectory principle—able to provide different learning paths for different types of developers: beginner, intermediary, or expert. We need to be smart and reasonable enough to allow intermediate users to move towards becoming experts so that they can extend their abilities and increase their productivity and satisfaction.

The Key Elements of the Developer Onboarding Experience

The most important aspect for any onboarding experience is to show the value of your product or service. When you are building your developer experience, your primary aim should be to design your APIs that would help your developers reap value in their business and make them better. It is not about your API design, but more about how your API transforms your developers' business or work. Remember the Super Mario game—we all were interested in gobbling up the red flower because it would transform the character giving him some super powers. That is exactly what your APIs should be—developer experience is all about transforming your developers from good to better to best.

The next important element is the landing or home page of your developer portal. In today's world, you might never get a second chance, with your developers, to make a first impression. Your homepage should invite developers to stay, explore, or take action. The call to action (CTA) lays

out what the developer is expected to do. Keep your sign-up process simple and fast; preferably allow social login as well to remove any first time experience barriers.

For any kind of developer experience you are trying to build—Hello World is the Holy Grail. Getting started with your APIs should be simple and frictionless for developers, so they can get off the starting line as quickly as possible. From a one-minute experience to the one-hour experience; you can provide different types of getting started onboarding materials for your developers. This goes a long way for developers to see examples of what they're trying to do, and show them what else is possible with the API, providing a nice overview of the API itself.

A large percentage of developers are expecting SDKs in their preferred programming language. Providing a convenient SDK to developers creates a great incentive to use your API over your competitors. Light the path by creating add-ons and tools that are right for your developers. Make it easier for them by giving them relevant libraries and SDKs, with appropriate comments and instructions, which they can integrate in their products. Link to outside libraries and open-source projects that exist.

The real value of your product lies in the sustained improvement of the developer using it. Help build a developer network. Deliver as much useful content on your blog, hire developer evangelists to teach courses and give talks at meetups, and make contributions to open source. By doing these, you are creating a tribe of developers who want to work with your APIs. As part of your developer experience, your organization must think about including events, which are a great opportunity to get the word out about your API program, to promote its benefits, and to increase adoption.

As with any product, there would be early adopters of your APIs during the pilot phase. The advantage is that you get early feedback about your API and you can improve it accordingly. Second, you can showcase the success on your developer portal. It always helps the success of your API adoption when you are able to demonstrate real world examples of how another developer or organization is using your APIs to derive value to their work or business.

Conclusion

When creating a developer portal and the surrounding developer experience, try to focus on these things:

- Design an API with your developer (DX)— build APIs with great functionality and usability in

order to foster a positive emotional response for your developers

- Know your developer type—who is using your API and how they are using it (through analytics, forums, interviews, other metrics.)
- Give developers the resources needed to quickly start using your API
- Create opportunities for partners to earn revenue by using your platform
- Establish a channel that connects your API partners' customers to you and you to them

and lean content, user and developer experience (UX/DX), 3D printing and augmented reality, information architecture, and project management. He is also a Certified Scrum Master (CSM).

Great developer experiences are invisible. When your developers are having a great experience, they rarely notice the hard work that has been put into place to make it happen. This is what encompasses the right kind of developer experience you want to build for your organization in this API economy.

References and Additional Resources

Don't Make Me Think by Steve Krug

The Elements of User Onboarding by Samuel Hulick

Developer Evangelist Handbook written by Mozilla
Developer Evangelist Chris Heilmann

100 Things Every Designer Needs to Know About People
by Susan Weinschenk

Nordic APIs

API Evangelist Website. Author Kin Lane

Autodesk Forge Platform

Author Contact Information

Nabayan Roy

Principal Learning Content Developer

Autodesk

+65 6643.6158

nabayan.roy@autodesk.com

About the Author

Nabayan Roy is a principal learning content developer and product owner at Autodesk, Singapore. He has been involved in technical communication for nearly 15 years in various roles, such as information architect, content strategist, and programmer writer.

Previously, Nabayan has presented at various international conferences across the globe. His interests include agile



THE MAKING OF A LEARNING PLATFORM: DESIGNING ADDICTIVE, USER-CENTRIC CONTENT

Pavi Sandhu

Trailhead is a new learning platform, created by Salesforce, that has become popular due to its effectiveness as a learning tool. Its innovative features include guided learning paths, modular content, interactive challenges, gamification, and social rewards. The design of Trailhead posed many challenges, which required placing the needs of customers at the heart of all decisions about creating and organizing content. Trailhead's content model illustrates fundamental principles of user-centric design and learning theory, which can be applied to make any instructional content more engaging, accessible, and useful.

As technical communicators, we all face a fundamental challenge: how do we motivate our audience to engage with the content we create? Reading manuals or clicking through online help isn't exactly fun for most people. Therefore, any innovation that makes the learning experience more enjoyable and effective can add enormous value, helping our customers, and therefore our company and career, to be more successful.

This was the challenge we faced at Salesforce, which has grown rapidly, from a startup to the world's fourth largest software company, with over 20,000 employees. As the number of products and features multiplied, there was a corresponding increase in the amount of documentation. Our technical writers were doing an excellent job of providing clear, comprehensive, and accurate information, in the form of online help, user guides, workbooks, cheat sheets, release notes, and so on. However, the sheer volume of content, without a clear path to navigate it, was overwhelming. This made learning to use our products increasingly difficult, for both new and experienced customers.

To address this problem, about two years ago, our Documentation team, in collaboration with Developer Marketing, began designing and building a new learning platform, called Trailhead. The goal of Trailhead is to empower customers to use Salesforce more effectively, by radically simplifying and redefining the learning experience. We were inspired by the surge in popularity of learning websites, such as Udacity, Coursera, edX,

Khan Academy, and Lynda.com and the advent of a new learning model, known as the MOOC (Massive Open Online Course). A MOOC, as defined by Wikipedia, is an online course aimed at unlimited participation and open access via the web. Many universities are now providing a wide range of courses as MOOCs, enabling anyone, anywhere to register and learn online. These sites provide a valuable alternative to traditional classroom learning, by making high-quality education accessible to a much larger audience.

What is Trailhead?

Trailhead adapts and extends the approach to online learning popularized by MOOCs to create a new model for product documentation. It provides a guided learning experience, introducing customers to specific product features through a set of free, interactive, web-based tutorials. Each unit contains a blend of text, graphics, and videos, and uses a playful, informal style to engage and retain the customer's interest.

Trailhead has these key features.

- User-centric content—The content is organized by role, level, and goal, to meet specific learning objectives. This ensures customers have a pre-defined learning path to follow, and don't have to spend time discovering and organizing relevant content.

Technology and Development

- Self-paced, online learning—Customers can learn what they need anytime and anywhere, and progress at their own pace.
- Modularity—The content is arranged in a hierarchy, with three levels: trails, modules, and units. Each module is designed to be simple, self-contained, and directly useful, and consists of short units, which can be read in 10-15 minutes each.
- Interactive challenges—At the end of each unit, customers can verify what they just learned, by answering multiple-choice questions or performing specified tasks in the Salesforce application. On completing a challenge, they can click a button and get instant feedback on their work.

Since our software is cloud-based, we can make API calls to check if the customer completed the task correctly, and provide relevant hints if they did not.

- Gamification and social rewards—We award points and badges on successful completion of units and modules, respectively. Badges are displayed on the customer's public profile and are sharable on social networks, such as LinkedIn and Twitter. This provides customers peer recognition for their expertise, and positive reinforcement to continue learning.

The screenshot shows the Salesforce Trailhead home page. At the top, there is a navigation bar with the Salesforce logo and 'trailhead' text, a search bar, and 'Login' and 'Sign Up' buttons. Below the navigation bar, there are tabs for 'Home', 'Trails', 'Modules', and 'Projects'. The main content area features a large banner image with a trail map and a play button, with the text 'the fun way to learn Salesforce' and 'Everyone can learn Salesforce. Whether you are an admin, user, or developer, there is a trail for you.' Below the banner, there is a blue button that says 'Start learning for free'. Below the banner, there are buttons for 'Trails', 'Modules', and 'Projects'. The main content area is titled 'Follow a guided learning path' and lists several learning trails with their durations and levels.

Trail Name	Duration	Level	Target Audience
Admin Beginner	8 hrs 45 mins left	Beginner	Admin
Admin Intermediate	5 hrs 55 mins left	Intermediate	Admin
Migrate to Lightning Experience	3 hrs 45 mins left	Intermediate	Admin
Developer Beginner	15 hrs left	Beginner	Developer
Developer Intermediate	14 hrs 30 mins left	Intermediate	Developer
Develop for Lightning Experience	10 hrs 35 mins left	Intermediate	Developer
Learn CRM Essentials	4 hrs 35 mins left		
Sell Lightning Fast with Salesforce	1 hr 55 mins left		

Figure 1. The Trailhead home page shows a set of trails, which are guided learning paths through specific areas of Salesforce.

Trailhead's innovative approach to learning has been remarkably successful. Since its launch, in Oct 2014, Trailhead has received an enthusiastic response, generating dozens of blog posts and thousands of tweets, with 65,000 customers completing over a million units and earning 250,000 badges. It's rare, if not unprecedented, for technical documentation to generate this level of excitement. Not only are our customers having a satisfying experience, we're helping make their jobs easier while increasing adoption and usage of our products. Trailhead has been so effective that Salesforce is now adopting it for internal use, as our primary platform for onboarding and training all employees.

The interesting question, from a technical communication perspective, is: Why is Trailhead so popular, and can we draw any lessons from its success that are applicable in other learning contexts? This paper explores some ideas relevant to that question.

One of our goals with Trailhead was to organize the large amount of content we had already created, and present it in a form that is so relevant, engaging, and useful that customers are motivated to learn. A key element of the solution was to design learning paths that would provide a guided sequence of topics, to address the most important customer use cases.

Consider this analogy: a typical university library has millions of books, covering all the courses offered by the university. Any student can readily access all this information, but can't be expected to put together a program of courses that would lead to a degree entirely by self-instruction. The knowledge required for all the degree programs is contained in the library stacks, but it's not very useful to a student until it's organized into a curriculum, consisting of a sequence of courses, each with its own syllabus and readings. Providing that organization, and then developing activities to support the learning of that content, is the task of the university's departments and professors. This analogy highlights the crucial importance of curriculum development – to facilitate learning, by selecting and organizing relevant information to fit a specific educational purpose.

We faced a similar challenge at Salesforce, but with a more limited repository of information. We had to organize not a university library, but the library of Salesforce product documentation. Trailhead was our approach to solving this challenge. However, learning paths though useful are not sufficient by themselves. A more holistic, integrated approach to learning is required.

Insights from Learning Theory

As technical communicators, it's useful to think of ourselves as teachers, with our customers as students. This is a valuable metaphor, enabling us to leverage insights from the field of education to meet our fundamental goal, that is, to help customers learn to use our products more easily. Translated into educational terms, this is equivalent to answering the question: How can teachers help students learn more effectively? This issue has been the focus of research by educators for over forty years, leading to important insights, which are directly relevant to technical communication. Let's take a brief detour into learning theory to explore some of these concepts.

Research into student learning received a major boost in the 1970s, with the publication of a classic study by Marton and Saljö [1]. They gave a number of students some text to read, and told them they would be asked questions on it afterward. The students responded in two distinct ways. One group read the text superficially and remembered isolated facts, without understanding the connections between them. Another group went below the surface and focused on understanding the meaning the author was trying to convey. Marton and Saljö distinguished the two approaches by the terms surface learning and deep learning, respectively. The words "surface" and "deep" refer to particular approaches to learning, rather than the innate traits of the students.

In any group of students, there is a wide variation in the level of motivation and ability, which translates into variations in the outcome of the learning experience. The most talented and motivated students learn a great deal, even in adverse conditions. The challenge of teaching is to make even average students have extraordinary results. A great deal of educational research has focused on how to make teaching more effective, by guiding students to engage in deep learning rather than surface learning. Researchers have distinguished between two broad approaches to teaching: student-focused and teacher-focused [2]. In teacher-focused approaches, learning is a result of the teacher's methods and activities, while in student-focused approaches, the emphasis is on what students do to achieve understanding.

The Australian educational psychologist, John Biggs, is one of the major innovators in the field of student-focused learning [3-5]. He classifies all theories of teaching into three levels, depending on their primary focus.

- Level 1 – Learning depends on the students' level of motivation and intelligence. The focus is on what the student is. Hence, any failures in learn-

ing are the student’s fault. This can be called the “Blame the student” model.

- Level 2 – Learning is effective when the teacher moves beyond simply transmitting information to conveying concepts and principles. The focus is on what the teacher does. Hence, any failures in learning are the teacher’s fault. This can be called the “Blame the teacher” model.
- Level 3 – Learning is a result of students engaging in activities designed to directly support the desired learning outcomes. The focus is on what the student does. Learning outcomes are the shared responsibility of the teacher and student, with each having a specific role.

According to Biggs, most teachers start at Level 1 but move to higher levels as they grow in experience and ability. He considers the Level 3 approach the most effective and has developed a theory he calls constructive alignment, which provides explicit guidance on how to maximize the impact of teaching.

Biggs explain his theory in these words: “The ‘constructive’ aspect refers to the idea that students construct meaning through relevant learning activities. That is, meaning is not something imparted or transmitted from teacher to learner, but is something learners have to create for themselves. The ‘alignment’ aspect refers to what the teacher does, which is to set up a learning environment that supports the learning activities appropriate to achieving the desired learning outcomes.”

There is a great deal of support for Biggs’ theory from work by other researchers. Similar ideas were first advocated by Tyler in the 1940s [6], and developed in the 1980s by Shuell [7], who summarized his findings as follows. “If students are to learn desired outcomes in a reasonably effective manner, then the teacher’s fundamental task is to get students to engage in learning activities that are likely to result in their achieving those outcomes... It is helpful to remember that what the student does is actually more important in determining what is learned than what the teacher does.”

Learning objectives are concise and specific statements that describe what the student will be able to do as a result of a learning activity. At its most basic, Biggs’ theory requires alignment between the intended learning objectives, the learning activities the student engages in, and the assessment criteria used to test what the student has learned. The relationship between these three key components is summarized visually as the triangle of effective learning (Figure 2).

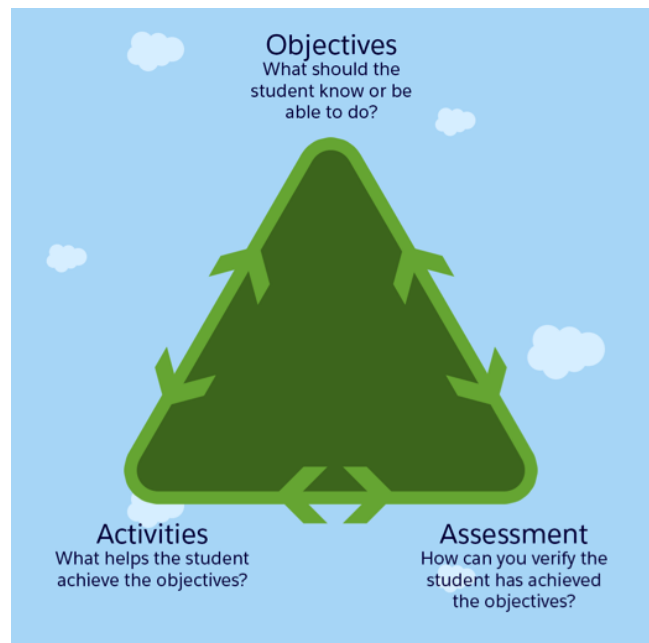


Figure 2. The triangle of learning, illustrating John Biggs’ theory of constructive alignment.

The key takeaway of this research is that teachers should focus on creating teaching and learning activities (TLAs) that are congruent with the learning objectives (LOs). Therefore, the proper order for creating any instructional content is as follows (Figure 3).

1. Define the intended LOs, that is, what the student should know or be able to do.
2. Design tests to assess if the student has met the LOs.
3. Develop TLAs that help the students achieve the LOs and meet the assessment criteria.

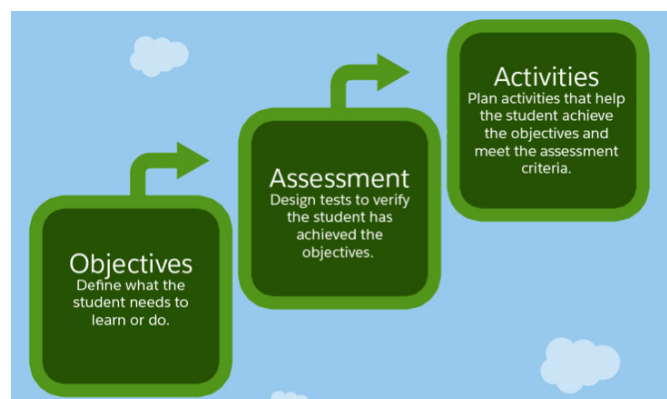


Figure 3. The steps involved in developing an integrated curriculum.

The advent of the web provides a significant opportunity to implement this process effectively. That’s because the web, being an interactive medium, is ideal for active

learning. A major advantage of web-based instruction is it expands the types of TLAs possible, enabling scalable, interactive learning without a human teacher. The key to success is to utilize the web's interactivity to increase the students' level of engagement, by transforming the learning experience from passive to active. Trailhead's design incorporates the above principles, which is a key reason for its effectiveness as a learning tool.

Designing User-Centric Content

Switching gears from the field of learning theory, it's useful to consider some ideas from the history of technology innovation, where similar concepts have surfaced in a different context. The distinction between teacher-focused and student-focused strategies in education closely parallels that between feature-centric and user-centric design in the creation of technology products. The classic example of this is the difference between the operating systems created by Apple and Microsoft. Prior to Apple's introduction of the Macintosh in 1984, the dominant paradigm for interacting with computers was via the command line, as exemplified by Microsoft's MS-DOS. This required the user to type in obscure commands to perform even simple tasks, such as, navigating between directories and files.

The Mac was a radical advance because its graphical interface provided a desktop with files and folders you could navigate visually by pointing and clicking, eliminating the need to learn and type arbitrary commands (Figure 4). This introduced a new paradigm of computing in which users no longer had to adapt their behavior to fit the computer. Instead, the computer was designed to fit how people naturally think. This simple but profound shift in perspective launched the personal computer revolution, making computers accessible and useful to a mass audience, instead of being confined to a specialized group of hobbyists and researchers.

Trailhead has the same relationship to traditional documentation that the Mac OS had to MS-DOS. Trailhead represents the concept of learning paths through the visual metaphor of trails, in the same way that the Mac OS uses the visual metaphor of a desktop. Trailhead content is also user-centric in that before writing a module on any feature area, writers spend a significant amount of time thinking about the use cases of greatest importance to customers, and then construct learning objectives, assessments, and content, specifically to address them.

Each module provides a predefined learning path to fit the customer's goals, so there's no need for customers to make decisions on what to learn and how. In effect, the documentation adapts to the customer instead of the customer having to adapt to the documentation. This ensures customers can focus their time and attention on high-value activities that directly contribute to learning instead of on discovering and organizing relevant content.

Conclusion

Trailhead uses the interactivity of the web, to create an environment in which the customer is guided to engage in deep learning. It accomplishes this by aligning and integrating the learning objectives, assessment, and content, so they all work together to serve the goal of helping the customer learn. Trailhead encourages and facilitates active learning whereby customers learn by doing activities specially designed to meet objectives that ensure mastery. Finally, customers are tested to demonstrate they have acquired the intended knowledge and skills.

According to researcher John Cowan [8], "Teaching is the purposeful creation of situations from which learners should not be able to escape without learning or developing." The key to Trailhead's success is that it creates an environment in which customers can't escape without learning, through a novel implementation of the princi-

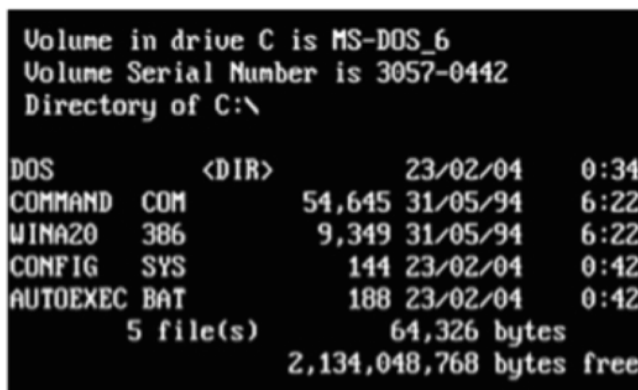


Figure 4. The user interfaces of Microsoft's MS-DOS and Apple's Mac OS illustrate the difference between feature-centric and user-centric design.

Technology and Development

ples of user-centric design and constructive alignment. The same principles can be applied to assess and enhance the usability of any instructional content, regardless of its audience and format.

Resources

Trailhead: <https://developer.salesforce.com/trailhead>
The Making of Trailhead (video recording of session at Dreamforce 2015 conference): <http://salesforce.com/vidyard.com/watch/SiQIp9cWyVVkp4eYXdYdZg>
Wikipedia entry on MOOCs: https://en.wikipedia.org/wiki/Massive_open_online_course

References

- Marton, F. and Saljö, R. "On qualitative differences in learning. I-Outcome and Process." *British Journal of Educational Psychology*, 1976: 46, 4-11.
- Prosser, M. and Trigwell, K. *Teaching for Learning in Higher Education*. (Buckingham: Open University Press), 1998.
- Biggs, John. "Individual differences in study process and the quality of learning outcomes." *Higher Education*, 1979: 8, 381-394.
- Biggs, John. "What the Student Does: Teaching for Advanced Learning." *Higher Education Research and Development*, 1999: 18, 57-75.
- Biggs, John. *Teaching for Quality Learning at University*. (Buckingham: Open University Press), 2003.
- Tyler. R. W., *Basic Principles of Curriculum and Instruction* (Chicago: University of Chicago Press), 1940.
- Shuell, T. J. "Cognitive conceptions of learning." *Review of Educational Research*, 1986: 56, 411-436.
- Cowan, John. *On Becoming an Innovative University Teacher*, 2nd edition, (Buckingham: Open University Press), 2006.

Acknowledgements

The author gratefully acknowledges the expert assistance of Lydia Mann, in reviewing the text, and Spencer McCall, in creating the graphics.

Author Contact Information

Sandhu, Pavi
Lead Technical Writer

Salesforce
1 Market St., Suite 300
San Francisco, CA 94105
+1 415.987.5780

Author Biography

Pavi Sandhu is a lead technical writer at Salesforce, in San Francisco. As a member of Trailhead's founding team, he designed its innovative content model, wrote its first module, and created templates and guidelines that provided the framework for all content development. He is also the author of *The MathML Handbook*, a guide to displaying mathematics on the web. Pavi has 18 years of experience as a writer, manager, and entrepreneur, at companies such as Wolfram Research, Oracle, and Collabis, and a PhD in physics from Boston University.



POWER TO THE PEOPLE: TARGETED CONTENT THROUGH DYNAMIC FILTERS

Dustin Vaughn

In this always on, always connected world, consumers expect content that is not only optimized for their device, but also optimized for their needs. These expectations add to the ever-present need for technical communicators to deliver more: the right content, on the right device, at the right time. This goes well beyond just publishing content to a format like PDF or HTML that can be displayed on many devices. The need is to impress and empower your customers by providing content that can be filtered down to fit their specific needs.

Expectations

Technology changes things. Due to technological advances, one of the most dramatically impacted areas is customer expectations. There are three fundamental expectations that must be addressed. Content should be:

1. Accessible from various devices
2. Limited to relevant content based on user preferences
3. Engaging and easy to consume in a short period of time

Accessible from Various Devices

The online shopping capabilities of amazon.com and the search capabilities of Google drive the expectation in many that every business should provide those same functions at a similar level. While few companies have the resources of Amazon or Google, there is still a desire and an expectation to connect on a deep level with timely and relevant content. Gone are the days where posting a PDF to a website counted as “mobile content”. End users expect content that is optimized for their device, whether it be a desktop, a tablet, or a mobile phone.

Relevant and Based on User Preferences

Many organizations maintain tremendous volumes of both internal and external content: product manuals, policies and procedures, marketing brochures, forum content, and more. With such vast resources available, it is also important that the content be discoverable to the audience in addition to accessible from various devices.

Engaging and Easy to Consume

Attention spans are decreasing rapidly. According to Anne Fischer on Fortune.com, “The average adult attention span has plummeted from 12 minutes a decade ago to just 5 minutes now.” When you consider this fact in conjunction with increased customer expectations, it becomes clear that getting the right content on the right device becomes absolutely critical to customer satisfaction.

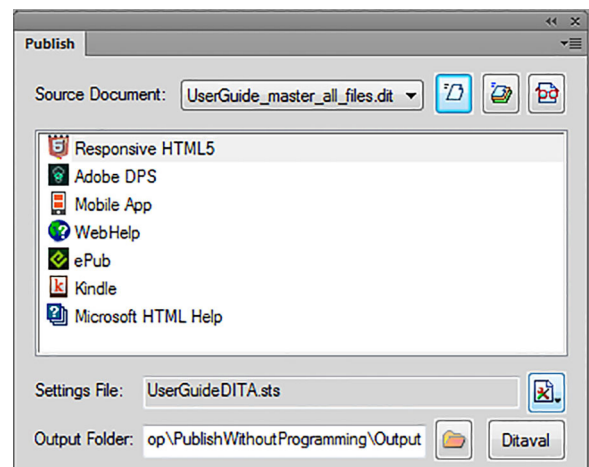


Figure 1. Publishing Multiple Formats from a Single Source

Solutions

These expectations place a significant strain on organizations in general, and technical communicators specifically. This often requires much more from the same individuals, with little or no increase in funding. To address the first expectation, platform-independent outputs such as HTML5 present a customized view of the content that responds to the size of the screen that is accessing the content. Other device-specific outputs such as Kindle (KF8, for instance), can be an excellent way to cater to specific target audiences. Regardless of the number of output types that are generated, it is important that they can be generated from a single document or set of documents to guarantee that the output is consistent.

For the second expectation, the organization must help their user find relevant content in a timely manner. This can be done by producing a variant of the content for each type of variable that is expected. However, this can be very difficult to manage as the number of variants grows. Another way to address the issue is through the use of dynamically filtered content. For example, a user may be able to select the content they would like to see from various categories, such as:

- Product
- Region (contact info, local laws, etc.)
- Department
- Audience (novice, expert, etc.)

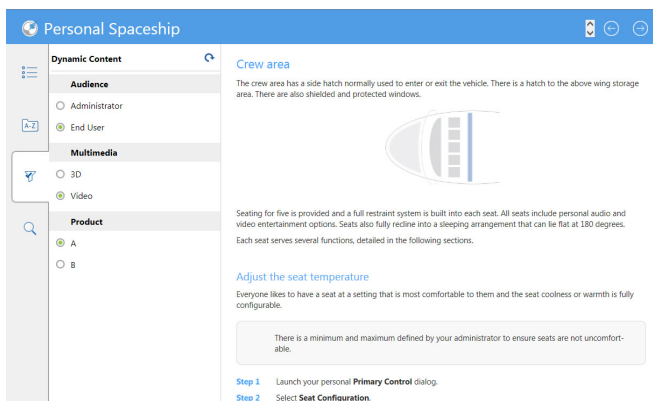


Figure 2. Dynamic Content Filters in HTML5

This self-selection of content makes the discovery of content easier by removing content that is not relevant (could be a word, a phrase, a topic, or an entire section of content). The Table of Contents, Glossary, Index, and even search results can be filtered based on these selections.

To address the third expectation (decreasing attention spans), it is beneficial to add interactivity to content where possible, such as:

- Pictures
- Videos
- Product simulations
- 3D Models

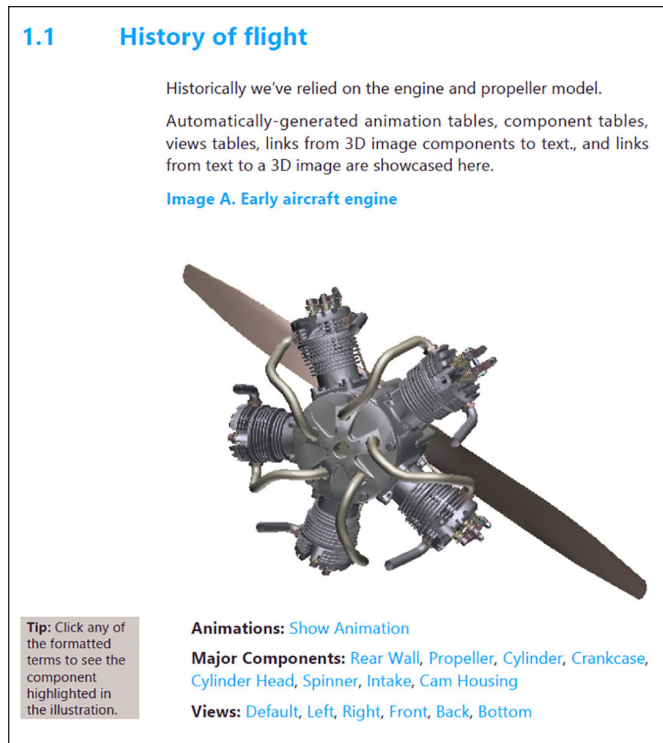


Figure 3. 3D Model with Interactive Links

In many cases it is important to meet these expectations without programming requirements, as it is not always feasible for organizations to either involve additional internal resources, or outsource this implementation. This can be accomplished with multiple tools, or in the best case through an integrated set of tools that addresses the entire lifecycle of content: from creation, to review, and then to publishing. While many tools only address a portion of these requirements, only the Adobe Technical Communication Suite addresses each of these requirements without requiring any programming knowledge to implement.

References

Fisher, Anne. "Giving a speech? Conquer the five-minute attention span." Fortune.com (10 July 2013).

<http://fortune.com/2013/07/10/giving-a-speech-conquer-the-five-minute-attention-span/>.

Author Contact Information

Dustin Vaughn
Solutions Consulting Manager
Adobe
252 Marker Rd
Rotonda West, FL 33947
+1 979.985.5536

Author Biography

Dustin Vaughn is a solutions consulting manager at Adobe who manages a team of experts in Adobe's Technical Communication and eLearning products. Dustin has served as a panelist, speaker, and Adobe representative at various industry conferences. He designs, develops, and hosts technical, solution-oriented demonstrations for strategic global customers. Dustin also worked as a solution designer and presenter for a cross-organizational and multi-product project designed for a key US Government client.



DEVOPS DOCS: FAST, GREAT CONTENT FOR THE CLOUD

Vanessa Wilburn

Writing for cloud products? Great! But what does that mean for writers? Companies adopt Agile and specifically DevOps to be fast, and doc teams are expected to follow suit. Most people think of engineering teams when you talk about DevOps. But doc teams can benefit from the speed and agility that you gain from continuous delivery. This paper covers how doc teams can adapt and thrive in a DevOps world by writing fast, writing well, and showcasing business value. In the end, your companies will see content that drives revenue, lowers costs, and even engages customers.

Cloud products are driving faster publication cycles, even daily pushes. Automation tools and different writing techniques speed up your publication. You can still ensure quality when you're writing so quickly.

Introduction to DevOps Documentation

Cloud products are different; their market demands new and updated features fast. The cloud allows those features to be instantaneously available to the market. Since the competition can push features in dramatically quicker times, companies have to be equally quick with features. Traditional development for on-premises software often included long test, translation, and fulfillment phases. Doc

teams used that time in a waterfall fashion to finish and refine their work.

In DevOps, that time has evaporated. DevOps squads are pushing code daily, weekly, and monthly. (Teams in the DevOps world are called squads.) Similarly, the docs for those squads need to be ready and published at the same time. Writing teams that have participated in agile have leveraged sprints, multi-disciplinary teams, and minimally viable products (MVP) to focus their work. But those agile processes are not enough.

A unique aspect of DevOps docs is the cloud-only delivery of content. Some on-premises products still delivered physical documentation, such as printed books. The practice has mostly disappeared, but in place of physical books,

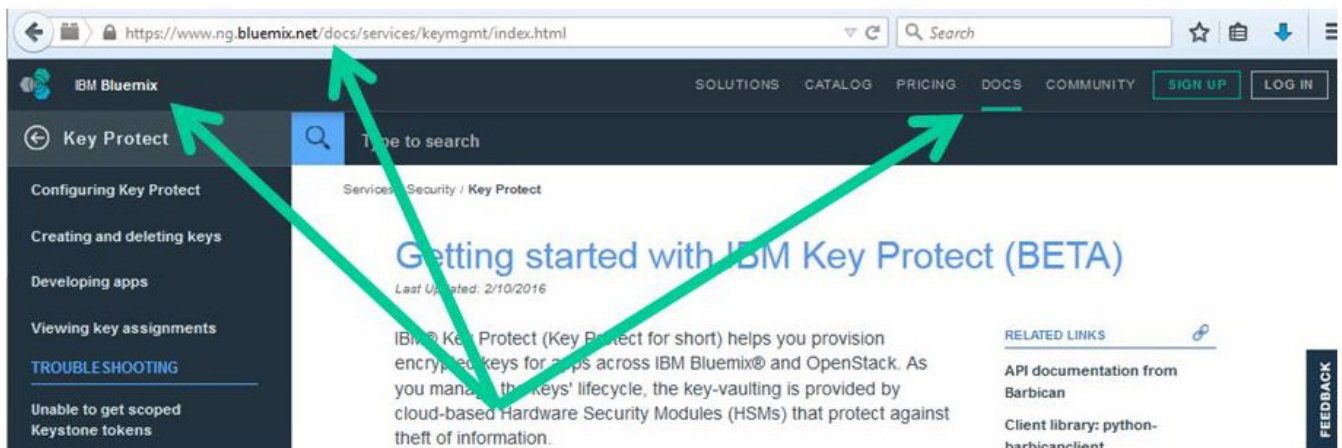


Figure 1. Example of cloud docs

those teams could still be shipping PDFs and HTML help systems along with the code on DVDs or installable images. That practice resulted in docs that became out-of-date the day that were burned onto an installable image. Quality refinements and fix pack updates resulted in complex processes to update these installed docs. Often the customers never updated their docs, even if they had the opportunity. For example, you can envision all those PDFs floating around on thumb drives. Finally docs, just like software code, were deeply tied to versions, such that doc teams were maintaining 8-10 versions of the same documentation sets (sometimes even more!).

What cloud docs look like is simple: HTML pages served off a website alongside the cloud product, including API docs in Swagger or other API docs frameworks. The following image shows how the docs sit on the same URL as the IBM offering. You can also see how docs are accessible from the same GUI via its top menu system.

Faster Publication

Now that you've heard how docs fit into the DevOps world, let's talk about how my Doc Squad delivers end-user docs, starting with speed. Our toolchain is essential to speed: both in automation and in its tooling.

To start, we created our own build system on Jenkins, intentionally separate from the engineering squads' Jenkins, so that we were not impacted by their bursts, outages, and space contention. Our own build system gave us a separate staging environment, with specialized builds for unique requirements outside of our regular offerings for the production doc site. Other build tools you might use are IBM Rational Build Forge, Apache Maven, Ant, or a combination of those, others, and your own custom tooling. There are many build tool options.

For source, IBM has a Content Management System (CMS) called IDCMS, which is aware of our source (dita), making it simple to deal with source control and shared files. We also rely on another build system from our peer team that controls our public staging and production deployments. These build systems all pull from the same CMS. To give you a taste, other parts of our toolchain include gitHub for engineering source files, sprint planning in Kanban boards, DITA development tools, quality reviews in IBM Eclipse Help System Collaboration, user feedback via Usabilla, and multiple web analytics tools for usage statistics.

All this tooling and automation lead to writing techniques that are adopted straight off the Spotify model. Our writers work in pairs, giving us higher quality (two sets

Build History	
#578	Apr 12, 2016 8:24 AM
#576	Apr 7, 2016 8:15 AM
#575	Apr 6, 2016 8:13 AM
#574	Apr 5, 2016 8:12 AM
#573	Apr 4, 2016 8:13 AM
#572	Apr 1, 2016 1:35 PM
#571	Apr 1, 2016 8:21 AM
#570	Mar 31, 2016 4:03 PM
#569	Mar 31, 2016 8:22 AM
#568	Mar 30, 2016 11:42 AM

- Last build (#578), 8 hr 15 min ago
- Last stable build (#578), 8 hr 15 min ago
- Last successful build (#578), 8 hr 15 min ago
- Last failed build (#568), 13 days ago
- Last unsuccessful build (#568), 13 days ago

Figure 2. Jenkins build page

of eyes) and coverage during vacations and illnesses. Those writers are actively collaborating on content with engineers: doing pull requests on engineering files, such as YAML, JSON, and Python.

Again from Spotify, our Doc Squad is maniacally focused on MVP (minimally viable product) as it regards to content strategy. We deliver only the most essential content quickly, not content for every nook and cranny of the cloud offering. This process is also in line with industry practices of progressive disclosure and embedded content. Modern documentation sets are optimized to address the top use cases for the majority of users. In that way, we've stopped overwhelming users with "nice to know" info. Instead, we deliver what they "need to know" to achieve their business goals.

Quality When You Have Little Time

Quality assurance might sound daunting when we're working so fast. Yet, our tools have reduced manual (and thus error-prone) activities. We've improved our quality from grammar, punctuation, and style errors with Acrolinx IQ. Accessibility testing is done iteratively as we deliver smaller chunks of content, headed by an accessibility focal point. Doc builds can flag XML validation problems, hyperlink issues, and bad file references. An iterative translation plan means that we translate and publish files every month (instead of yearly or longer cycles).

But tools are not the whole answer. As mentioned previously, knowledge sharing and expertise occur through paired writers. Those pairs are responsible for peer

reviews, translation, diagrams, and so forth. Our Doc Squad is spread across Austin, Raleigh, and Hursley. And, we collaborate with engineers across the United States, Ireland, England, Germany, Israel, India, and China. Thus, we rely on Slack, Screenhero, Mural.ly, video conferencing, and other collaboration tools.

To ensure collaboration, team building is a priority. So Slack is a great place to have silly conversations alongside technical discussions (hello Giphy). We play games online, the second Friday of each month. We've even appointed an Ambassador of Fun, further cementing team cohesion.

A key element of collaboration is integrating closely with the engineering squads, such as writers who attend engineering scrums. When we work in this embedded manner, we're able to deliver content quickly—in an environment that changes frequently and is based on a rapidly changing market and customer requirements.

Low-cost user testing is another way that we validate and improve the docs. Our strategy is to test early and with low-fi tools. We partner with our Design counterparts as they do user testing. When it's time-inefficient to find a customer to test with, we tap fellow engineers, technical sales, and product managers who have similar experience. Paper prototypes, sketches, and design walkthroughs are just a few of the methods we use to see how well the content works for users.

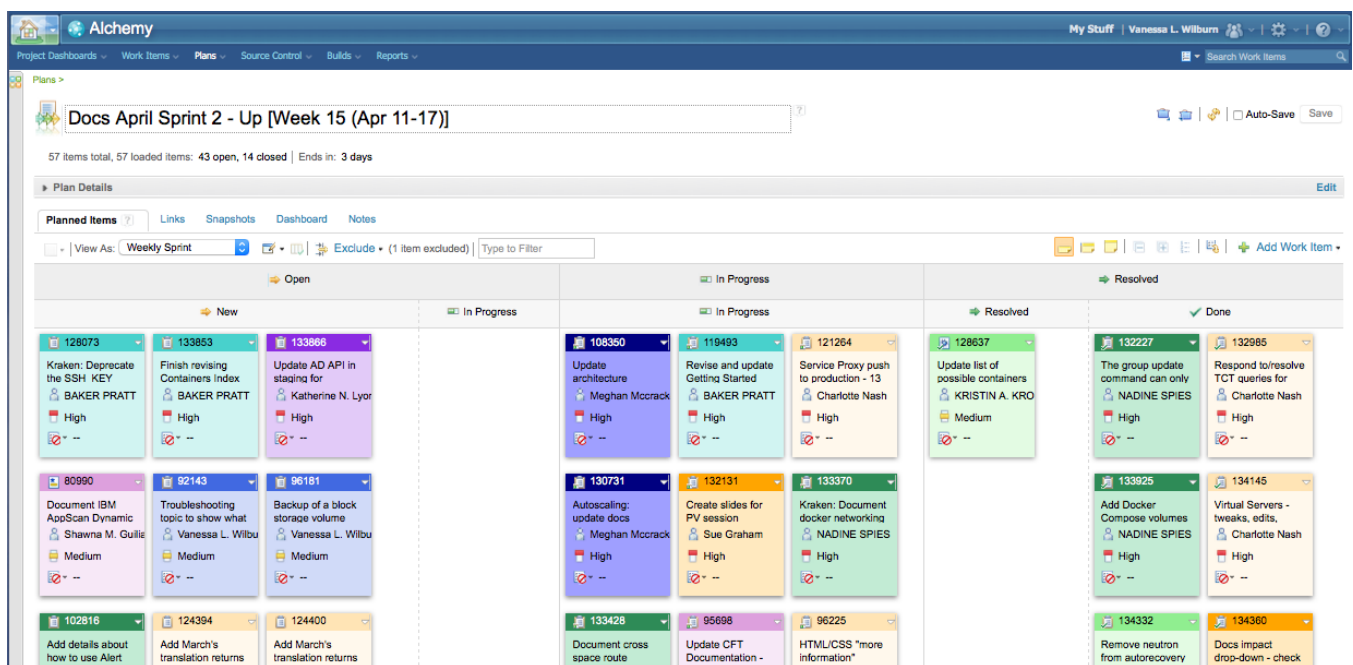


Figure 3. Kanban board in IBM Rational Team Concert
2016 STC Technical Communication Summit Proceedings

Continuous Integration, Continuous Improvement

Our future is true to DevOps: continuous improvement in an incremental fashion. We will have even more automation with our toolchain. For example, we've already developed build files to make translation shipments easier. We also expect to automate more of our translation file handling. Another possibility is for us to leverage our build environment to run even more automation, such as accessibility testing and pre-flight translation tests. Our future also means broader access to the business metrics for docs, such as acquisition and conversion of customers, self-help and support-call diversion, and engagement/reach.

Doc squads can adapt and thrive in a DevOps world: writing fast, writing well, and showcasing business value. In the end, IBM can see content that drives revenue, lowers costs, and even engages customers.

References and Resources

- Cantella, Julian, and Michelle Corbin. "Embedding the Editor: Tips and Techniques for Editing Embedded Assistance." Intercom 63.3 (September 2012). <http://intercom.stc.org/2012/09/embedding-the-editor-tips-and-techniques-for-editing-embedded-assistance/>.
- Carey, Michelle, Moira McFadden Lanyi, Deirdre Longo, Eric Radzinski, Shannon Rouiller, and Elizabeth Wilde, eds. Developing Quality Technical Information: A Handbook for Writers and Editors, Third Edition. Chapters 1 and 12. (Upper Saddle River, NJ: IBM Press), 2014.

Kniberg, Henrik. "Spotify engineering culture." Spotify (27 March 2014). <https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>.

Krug, Steve. Don't Make Me Think: A Common Sense Approach to Web Usability, Second Edition (Berkeley, CA: New Riders), 2005.

"List of build automation software." Wikipedia (14 April 2016). https://en.wikipedia.org/wiki/List_of_build_automation_software.

Author Contact Information

Vanessa Wilburn
Senior Content Strategist
IBM
Austin, Texas
+1 800.426.4968

Author Biography

As an information architect and content strategist for IBM Cloud, I deliver the content experience in the form of user interfaces, web-based information, product manuals, and other self-service information. I design and run usability tests to assess how users experience "content on the glass." The design work includes user flows, personas, prototypes, and taxonomies. Moreover, I've performed duties as a project lead, technical lead, and technical editor in support of IBM software. You can learn more about me on LinkedIn: <https://www.linkedin.com/in/vanessawilburn>

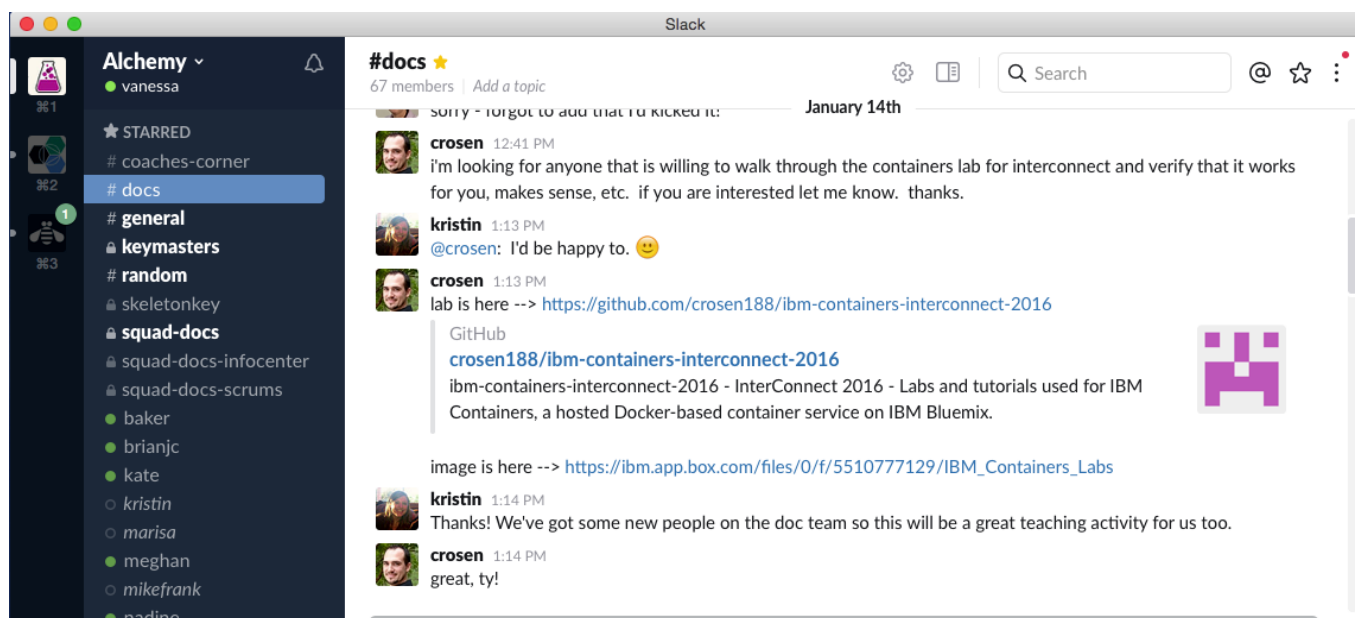


Figure 4. Collaboration within Slack

TECHNICAL EDITING

Cash in on your Financial Editing Skills

Jessica Kreger

145



CASH IN ON YOUR FINANCIAL EDITING SKILLS

Jessica Kreger

Your technical editing skills can be an asset in the finance industry. Find out how to apply your skills in this profitable field. Review specific writing examples that are acceptable from a regulatory standpoint, as well as general style guidelines. Learn about U.S. regulatory agencies, writing for an international audience, and resources for maturing your financial industry knowledge.

Editing Guidelines

TradeStation is an award-winning online brokerage firm that offers multi-channel trading through desktop, web and mobile apps. My technical documentation team and I support our clients through reference material. My team does not communicate directly with our clients or offer brokerage advice, because our help writers are not registered brokers. We do not purposely offer any suggestions, as you can see in the following examples of user documentation:

- ✓ “To place an order, enter a symbol in the Symbol box and click *Buy*.” This is acceptable, because it is telling the client how to use the tool.
- ✓ “To place an order, enter a symbol in the form of AAAA in the Symbol box and click *Buy*.” This is acceptable, too, and is even better. AAAA is an example of a symbol and provides useful instruction in the format the client must enter.
- ✗ “To place an order, enter <pick of the day> in the Symbol box and click *Buy*.” This is not acceptable. We can’t suggest what you should buy.
- ✓ “Trading hours are 9:30 a.m. to 4 p.m. Eastern Time.” This is acceptable.
- ✗ “The best time to trade is <pick your favorite time>.” This is not acceptable. We can’t suggest what time you should trade.

Style Guidelines

- Our style guidelines include maintaining a user-focus and a topic-based approach. We use the tools of information architecture to chunk information into digestible topics. To learn more about streamlining your style, see Ginny Redish’s *Letting Go of the Words, Second Edition: Writing Web Content that Works*.
- Beyond the words, in financial editing we try to avoid chartjunk, a term Edward Tufte coined in his seminal book, *The Visual Display of Quantitative Information*. His newer book, *Beautiful Evidence*, is another great resource for elegantly explaining and presenting data.

Regulation Nation

When editing in the finance industry, you must be compliant with both internal and external regulations. Some of the regulatory agencies that monitor the financial industry are the Commodity Futures Trading Commission (CFTC), the Financial Industry Regulatory Authority (FINRA), the National Futures Association (NFA), and the Securities and Exchange Commission (SEC).

Global Economy

- We operate in a global economy, and my team empowers active traders around the world. We manage multiple language versions of our software with different features and help systems for

Technical Editing

each. For example, we publish our order bar topic in three different languages: English, Chinese, and Japanese.

State. She is a Senior Member of the STC and belongs to the Technical Editing SIG.

There are a world of benefits to cash in when working in the finance industry. No two days are ever alike in the market, you're always on your feet. And to me, that's a major asset.

Resources

Barron's (25 March 2016). <http://www.barrons.com>.

Bloomberg Markets (25 March 2016). www.bloomberg.com/market.

Investor's Business Daily (25 March 2016). <http://www.investors.com>.

Online Trading Academy (25 March 2016). <http://www.tradingacademy.com>.

TradeStation University (25 March 2016). <http://www.tradestation.com/education/university>.

References

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

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

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

Author Contact Information

Jessica Kreger
Technical Documentation Manager
TradeStation Securities
8050 SW 10th Street, Suite 2000
Plantation, FL 33324
+1 954.652.7279

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 MA in professional writing from Carnegie Mellon and a BA in English from Penn

WRITING AND COMMUNICATION

BP/Horizon Oil Rig Explosion, GM Switch Malfunction, and Newark Water Contamination Debacle	149
<i>Carolyn Boiarsky</i>	
Building Quality Experiences for Users in any Language	159
<i>John Collins</i>	
Single-Sourcing and Localization: Maximizing Content Across Languages	165
<i>Laura Dent</i>	
Self-Publishing for Print and eBook Distribution	169
<i>Richard Hamilton</i>	
Applying Minimalist Writing Principles to Improve User's Guides: Readers and Writers Win	173
<i>Michael Harvey, Fellow</i>	
Frenemies: Tech Comm and Tech Support Working Together	177
<i>Rick Lippincott, Associate Fellow</i>	
Simultaneous Product and Information Development	181
<i>Janice Manwiller</i>	
Editors and SMEs: Building Productive Working Relationships	185
<i>Ann Marie Queeney</i>	
Using PDF as Source	189
<i>Liz Roscovius and Mike Sawyer</i>	
Improving the Review Cycle: Concurrent Editing	195
<i>Mike Sawyer</i>	
FAQs Should Support Documentation, Not Replace It	199
<i>Stephanie Saylor</i>	
Your Secret Weapon: The Documentation Handbook	205
<i>Kate Schneider</i>	



BP/HORIZON OIL RIG EXPLOSION, GM SWITCH MALFUNCTION, AND NEWARK WATER CONTAMINATION DEBACLE

Carolyn Boiarsky

Messages written by engineers to managers to communicate potential risks have often gone unheeded. Rhetorical analyses of these messages indicate that one of the major causes for the recipients' failure to take appropriate action to avert an impending disaster is that the text is writer- rather than reader-based. The failure of writers to take into consideration their readers' knowledge, reading processes and patterns has been exacerbated by the use of electronic media in which writers integrate social media conventions into their messages and perceive electronic media as analogous to the telephone. Recent messages concerning risk situations, such as the BP/Horizon Oil Rig explosion, the GM switch malfunction and the Newark Water contamination debacle, re-enforce these research findings. Recommendations for writing reader-based messages are provided.

Many of the messages written by engineers to managers to communicate potential risks have gone unheeded, according to Investigative reports of such large-scale disasters as the Chicago flood (Inouye and Jacobazzi 1992), the Challenger shuttle accident (Presidential Commission on the Space Shuttle Challenger Disaster 1986) and the Columbia shuttle break up (Columbia Accident Investigation Board 2003). Rhetorical analyses of these messages indicate that one of the major causes for the recipients' failure to take appropriate action to avert the impending disasters is that the texts are writer- rather than reader-based (Herndl, Fennell, and Miller 1991; Boiarsky 2015). According to Flower (1979), writer-based prose is "Verbal expression written by a writer to himself and for himself. It is the working of his own verbal thought. In its *structure*, writer-based prose reflects the associative, narrative path of the writer's own confrontation with her subject" (19). The writer writes for him/herself, ignoring the reader's needs and failing to account for the reader's prior knowledge and reading style and processes. In contrast Flower defines reader-based prose as "a deliberate attempt to communicate something to a reader. To do that it creates a shared language and shared context between writer and reader" (1979, 20).

The failure of writers to take into consideration their readers' knowledge as well as their reading processes and patterns has been exacerbated by the use of electronic media. In emails and texts the content of social media messages is often integrated into business correspondence. This confluence of genres often results in emails and texts including personal information along with business content, thus obscuring the focus of the message. In addition, messages often slip into the style of a personal blog or a twitter, becoming disorganized and either rambling or providing only minimal information, ignoring background data, definitions, and details that readers need in order to understand a text and, thereby, make an appropriate decision (Boiarsky 2016). Writers also often read electronic messages as if they were Facebook entries. They stop reading after the opening paragraphs, thus, failing to notice additional requests that have been included further into the message. This results in readers responding only to a single item, requiring the initiator of the message to follow up the initial message and, thereby, delaying action (Nielsen 1997; Outing 2011).

In addition to the influence of social media on business correspondence, there is a misconception that electronic media is analogous to the telephone. It is not. Unlike the telephone, electronic media is a-synchronous and one-

way. (Ashley 2003). The average time between messages is 28 hours (Skovolt 2009; Dabbish, Kraut and Fussell 2003). Furthermore, it is not two-way. If a reader does not understand a message, it may take two days, based on the 28 hour average, for the reader to get clarification. The writer's email requesting additional information may take one day to reach the reader with the reader's response taking another day.

While electronic messaging has become ubiquitous, much information continues to be written in traditional formats, such as reports, proposals and white papers. These are often sent as attachments and not written directly as emails and texts. Furthermore, information continues to be sent as hard copy in snail mail through the United States Postal Service, Federal Express or other carriers.

This paper examines the problems of electronic messaging, including the influence of social media and the misconception that it is analogous to the telephone as a form of communication. The paper looks specifically at the problems encountered in communicating risk prior to the BP/Horizon Oil Rig explosion. It also looks at the continued problem of writers' failing to recognize readers' reading patterns when communicating risk in hard copy letters and reports as evidenced in the texts related to the malfunction in the GM ignition switch and the contamination of Newark's water.

Influence of Social Media on Electronic Messages

The style that has evolved for social media messages (email, texting, Facebook, Twitter and LinkedIn) is often transferred to business messages, resulting in the use of an informal voice, the inclusion of personal information, and insufficient information.

Informal style. The formats for letters and memos have been amalgamated into a single format. The formal salutation of "Dear Ms. X" is often adapted to the more informal "Hi" of social emails and the closing "Sincerely" or "Yours truly" is all but eliminated. In the email chain in Figure 1, Sarah Dobbs refers to her readers as "Guys." ("Gals" would certainly have been considered sexist.) Note also the transcription of her Southern accent in the final paragraph of the first letter in the chain.

Integration of personal information. The confluence of social media with business content has resulted in the inclusion of personal information (the interjection of the "I" aspect of the social media.) in a technical message.

The interjection of personal information was one of the major problems in the correspondence between engineers and managers preceding the BP/Horizon oil rig explosion. The email chain in Figure 2, sent three days before the rig's blowout, assumes an eerie cast in the aftermath of the explosion that killed 11 people and injured 17. In the first email, the writer, John Guide, BP's well's leader, appears to recognize that he is in a risk situation and is in need of specific instructions. However, David Sims, BP's onshore manager for Gulf of Mexico oil drilling operations, fails to provide them, writing instead of his plans to go dancing, apparently perceiving Guide's email as having no more import than a Facebook "status update."

Insufficient information. Because social media messages are sent to a set of friends who have similar frames of reference, writers usually do not need to provide a lot of background information. However, this is not necessarily true of readers of business correspondence who may be located in different divisions in different geographical locations or who may be accountants or marketing analysts rather than engineers.

Lack of information was one of the major problems in the correspondence between engineers and managers preceding the BP/Horizon oil spill. (National Commission on the BP Deepwater Horizon Oil Spill and Offshore Drilling. *Deepwater: The Gulf Oil Disaster and the Future of Offshore Drilling: Report to the President 2011*). By responding to technical correspondence in the same way that they would to a social media message, writers often failed to recognize the multiple requests in a message. Often they responded only to a single request, usually the first. Note that in the Dobbs/Morel chain in Figure 1, emails go back and forth twice in order for Dobbs to obtain the information she needs. In addition, she doesn't have all the data she may need on tieback. Morel asks if she wants more information which, if she does, would mean at least another message and probably another day of delay.

Misperceptions of Electronic Messaging as Synchronous and Two-way

Electronic media, though often perceived otherwise, is a-synchronous and one-way. In most cases, the reader is not watching the writer write the message. If a message contains complex concepts or is concerned with controversial topics, synchronous, two way communication may provide a more effective method for communicating information.

From: Dobbs, Sarah
Sent: Tuesday, March 30, 2010 10:33 AM
To: Morel, Brian P; Hafle, Mark E
Cc: PTNEDA, FRANCISCO
Subject: Pip Tags and Casing

Guys -

We would like pip tags near the casing hanger, just below the production liner top, just above the sand, and one near TD.

Also, what swayed the decision to 7" liner? Was it availability or cementing concerns?

And, for historical knowledge, in August it looks like y'all discussed (with Hu) the option of 10-3/4" casing to 3000' below the mudline to accommodate a larger 1 5K SCSSV. It was our understanding that the casing was unavailable, so that option was eliminated. Is that correct?

By the way, do y'all know if there is an SoR floating around for this well.

Sarah Elisabeth Dobbs
BP Gulf of Mexico Deepwater
Completions Engineer

From: Morel, Brian P
Sent: Tuesday, March 30, 2010 10:54 AM
To: Dobbs, Sarah; Hafle, Mark E
Cc: PINEDA, FRANCISCO
Subject: RE: Pip Tags and Casing

7" is so we can run a long string instead of a tieback and still cement. If we had run 7-5/8" we would not have been able to cement as a long string with the amount of casing available, and would have had to double the source -3000' extra to bring the 7-5/8" above the 11-7/8" lianger a few hundred feet. We did not have this pipe in stock or easily available, and were able to get it from Nexcn.

Not running the tieback, saves a good deal of time/money as well as reduces complexity due to the conventional casing hanger or having to run a second packer and PBR assembly for the Versaflex hanger. If you want more details let me know.

We will take care of getting the pip tags. Do you want the casing hanger one moved to the top of the cross-over instead?

Thanks
Brian

From: Dobbs, Sarah
Sent: Wed Mar 31 21:15:19 2010
To: Morel, Brian P
Subject: RE: Pip Tags and Casing
Importance: Normal

Yes, that is great. Thanks.

Sarah Elisabeth Dobbs
BP Gulf of Mexico Deepwater
Completions Engineer

Figure 1. Email chain between Sarah Dobbs and Brian Morel related to BP/Horizon Oil Rig

David, over the past four days there has[sic] been so many last minute changes to the operation that the WSL 's [well site leaders] have finally come to their wits end The quote is —flying by the seat of our pants. Moreover, we have made a special boat or helicopter run every day.

Everybody wants to do the right thing, but this huge level of paranoia from engineering leadership is driving chaos. This operation is not Thunderhorse [one of the world’s largest rigs ever built. The rig was damaged by Hurricane Dennis.] Brian has called me numerous times to make sense of all the insanity. Last night's emergency evolved around 30 bbls [barrels] of cement spacer behind the topping and how it would affect any bond logging (I do not agree with putting the spacer above the plug to begin with). This morning Brian called me and asked my advice about exploring other opportunities both inside and outside of the company.

John, I've got to go to dance practice in a few minutes- Let's talk this afternoon.

for now, and until this well is over, we have to try to remain positive and remember what you said below - everybody wants to do the right thing. The WSLs will take their cue from you. If you tell them to hang in there and we appreciate them working through this with its (12 hours a day for 14 days) - they will. It should be obvious to all that we could not plan ahead for the well conditions we 're seeing, so WE have to accept some level of last minute changes.

Figure 2. Email between John and David related to BP/Horizon Oil Rig

In the chain of emails in Figure 1, if Dobbs had used a telephone instead of electronic mail, Morel might have been able to determine how much information Dobbs needed so that he could have provided it in a single response.

Stack (2014) emphasizes the need to select a method of dissemination based on the complexity of a message and the potential for conflicting responses. She has constructed the chart in Figure 3 as a way to demonstrate the relationship between various communication channels and the complexity of information being communicated. Telephone or person to person might have been the best way for Sims and Guide to take care of the problems in the email chain in Figure 2.

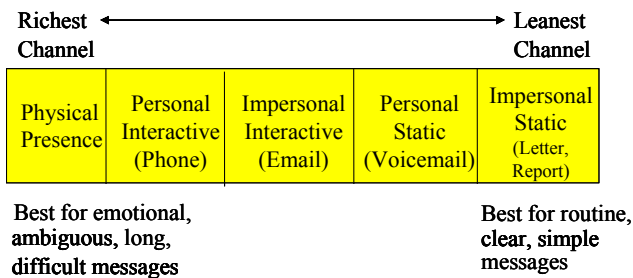


Figure 3. Chart for engaging in appropriate media, ranging from simple to complex messages

Failure to Consider Readers’ Reading Patterns

Research has indicated that people read hard copy in a “T” pattern, reading the first paragraph and then skimming down the middle of the following paragraphs. New research indicates that people reading emails and texts follow an “F” pattern (Nielsen 2006), reading the first sentence, skimming the remainder of the paragraph, reading the beginning of the second paragraph and then skimming down the left margin of the remainder of the document. Regardless of whether a message is in the form of an email or text message or if it is in the traditional form of a proposal or report that is sent either as an electronic attachment or as a single piece of hard copy, readers will read the beginning of it and then, in the majority of cases, either skim or scan the remainder or simply discard it. If important information is included in the middle, readers will probably fail to read it.

The failure of writers to recognize readers’ reading patterns is one of the major problems found in two recent communications related to risk situations--the GM report, indicating the problem of a switch in an auto accident, and the letter to the Newark school principals, warning of contaminated water. Note that in the GM report in Figure 4, the information that was later found to be the root cause of the problem is not the first bulleted item, but the second item discussed in the list of potential causes. In fact, the cause is not indicated until the second sentence in that item.

In the Newark letter to school principals in Figure 5, the information to run the water for a period of time before

drinking is located after two long paragraphs that are probably of little interest to the readers. In all probability the readers stopped reading somewhere in the middle of the first paragraph, thinking the letter was simply about the federal requirements for a report.

Effective Models of Risk Communication

If engineers are to successfully communicate risk to their readers so that their readers act appropriately, then they need to organize their information to match their readers' reading patterns as well as eliminate social media content and style from their business messages.

In 2011 the writers for the U.S. Army Corps of Engineers recognized these needs when the Mississippi River was cresting and there was a risk that Baton Rouge and New Orleans would be flooded again, thereby requiring that the Morganza Floodway be opened in order to divert the river's waters away from the two cities. In the letters in Figures 6 and 7, the engineers communicate the risk of flooding to the residents who are on the land that would be flooded when the spillway is opened. Unlike the single letter that is sent to the Newark school administrators which combines the government's role and the reader's need for action, the engineers separate these two purposes, providing a letter for each. The focus of the first letter (Figure 6) is on the reader's contract with the government while focus of the second letter (Figure 7) is on the actions that the reader must take. Note how in letter #1, the writer begins by providing a reminder to the landowners. However, in letter #2, the writer begins by discussing what is about to happen and what must be done.

RECOMMENDATIONS

Based on the rhetorical analyses of both successful and failed attempts to communicate risks, the following recommendations are made.

- Develop writer-based texts
- Begin all risk communication with the following information in the first paragraph
 - The purpose of the message
 - A one sentence summary of the information in the message
 - A list of the topics to be covered in the message
 - An indication of the number of items to which the reader needs to respond
- Refrain from including irrelevant or personal information.
- Consider communicating by telephone or person-to-person when issues are complex

References

- Ashley, Julia Lynn. "Synchronous and A-Synchronous Communication Tools." LuxMentor, (April 2015) <https://luxmentor.wordpress.com/2015/04/29/technology-assessment/2>.
- Boiarsky, Carolyn. Effects of Electronic Media on Technical/Scientific Communication: A Look at the BP/Horizon Deepwater Oil Rig Explosion. (Proceedings STC Summit Conference, 2015).
- Boiarsky, Carolyn. Risk Communication and Miscommunication: Case Studies in Science, Technology, Engineering, Government and Community Organizations. (Boulder, CO: University of Colorado Press), 2016.
- Columbia Accident Investigation Board. Report of the Columbia Accident Investigation Board.

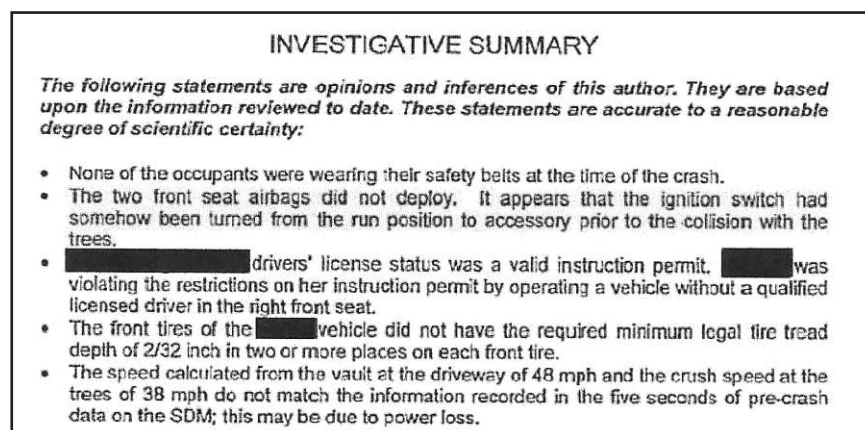


Figure 4. Summary of report of the potential causes of an accident involving a GM car



DEPARTMENT OF THE ARMY
NEW ORLEANS DISTRICT, CORPS OF ENGINEERS
P.O. BOX 60267
NEW ORLEANS, LOUISIANA

February 28, 2011

REPLY TO
ATTENTION OF

Real Estate Division

TO: ALL LANDOWNERS, RESIDENTS AND LESSEES IN THE BAYOU DES GLAISES LOOP, OLD RIVER CONTROL STRUCTURE PROJECT, THE WEST ATCHAFALAYA FLOODWAY AND THE ATCHAFALAYA BASIN FLOODWAY AND ALL LANDOWNERS/TRAILER OWNERS OR LESSEE-OPERATORS IN THE MORGANZA FLOODWAY

You are reminded that the property in which you reside, operate and/or are doing business is located within a floodway, or a portion of a floodway, which was developed under appropriate Acts of Congress by the United States Army Corps of Engineers. The function of the floodways is to safely pass the flood waters that may be generated in the Mississippi River and its tributaries to the Gulf of Mexico. The United States Government holds a perpetual right to flood the properties and improvements thereon. The Government's recorded easements provide that the United States shall in no case be liable for damages to property or injuries to persons that may arise from or be incident to operation of the aforementioned floodways.

In the event the operation of any of the floodways is required, public notice will be given through your local civil defense officials, as well as all news media, sufficiently in advance to allow time for evacuation of people and livestock and for removal of personal belongings. Upon receipt of such notice, expeditious action must be taken to protect life and property. Once the floodwaters enter the floodways, the height of the water is estimated to reach between fourteen (14) feet and twenty-five (25) feet above ground elevation depending upon the location of the property within the floodways. As a resident or user of one of the above floodways, you must be aware that it is likely that your home, business and/or personal property could be flooded. In some areas of the floodways, the strength of the currents could thrust improvements from their foundations and carry them through the floodways as debris.

As a user of the floodway, it is your responsibility to minimize environmental contamination during operation of the floodway. Farmers and commercial pesticide applicators that maintain supplies of pesticides must secure and protect these products. In case of high water, measures must be taken to prevent uncontained spills of pesticides, all in accordance with Louisiana Administrative Code Title 7, Pesticides. Also, any property owner that stores or permits storage of hazardous materials, hazardous waste, or NORM (Naturally Occurring Radioactive Material) on their property, must take similar precautions. If you operate any water or gas wells within the floodway, they must be sealed and capped to prevent any contamination from floodwaters.

Figure 5. Letter providing instructions to school personnel about water contamination



DEPARTMENT OF THE ARMY
NEW ORLEANS DISTRICT, CORPS OF ENGINEERS
P.O. BOX 60267
NEW ORLEANS, LOUISIANA 70160-0267

REPLY TO
ATTENTION OF

May 6, 2011

Real Estate Division
Management, Disposal, and Control Branch

TO: THOSE WITH IMMOVABLE AND/OR PERSONAL PROPERTY, LIVESTOCK, AND THIRD PARTY INTERESTS WITHIN THE OLD RIVER CONTROL COMPLEX, MORGANZA FLOODWAY, LOWER ATCHAFALAYA BASIN FLOODWAY (HWY 190 TO MORGAN CITY BETWEEN THE EAST AND WEST ATCHAFALAYA BASIN PROTECTION LEVEES), THE WEST ATCHAFALAYA FLOODWAY, AND THE UPPER POINT COUPEE DRAINAGE PROJECT

The existing and predicted stages on the Mississippi River indicate that it may be necessary to operate the Morganza Floodway to handle and safely divert these floodwaters through the Morganza Floodway. If operated, these waters will ultimately proceed to the Gulf of Mexico through the Atchafalaya Basin Floodway.

In the event that it becomes necessary to actually operate the Morganza Spillway, public notice will be given through your State and local emergency officials and other governmental authorities, as well as major news media, in advance to allow time for evacuation of people and livestock and for removal of personal belongings, including trailers. Any instructions you have already received or will receive from State and local emergency officials should be followed. Expeditious action must be taken to protect life and property.

Once the floodwaters progress through the Morganza and Atchafalaya Floodways to the Gulf of Mexico, the height of the water could reach between five (5) and upwards of twenty-five (25) feet above ground elevation, causing widespread flooding and inundation. Possible impacts due to the water flow currents include: damage to structures, removal of structures from foundations, and sweeping of other items, such as trailers and livestock, downstream.

Measures must be taken to prevent uncontained spills of pesticides, in accordance with Louisiana Administrative Code, Title 7, Pesticides. If you store or permit storage of hazardous materials or waste, or Naturally Occurring Radioactive Material (NORM) on your property, please take the same precautions. If you operate any type of well within the floodways, it must be sealed and capped to prevent contamination. In addition, secure any propane tanks in the area.

In summary, please keep advised of instructions from State and local officials as your safety is our number one priority.

Figure 6. Letter reminding citizens that they are on land that could be flooded



Cami Anderson
State District Superintendent

THE NEWARK PUBLIC SCHOOLS
Office of Facilities Management at
Harold Wilson School
190-218 Muhammad Ali Avenue
Newark, New Jersey 07108
Phone: 973-733-7340
Fax: 973-733-6852



David C. Hespe
Commissioner of Education

Keith Barton
Executive Managing Director of Operations

Gerald D. Bland
Director

PLEASE POST

MEMORANDUM TO: All Principals/Vice Principals
Senior/Head Custodians
Supervisors of Custodians/Regional Building Managers
Nutritional Services

FROM: Keith Barton
Executive Managing Director, Operations

DATE: August 30, 2014

RE: **Mandatory Daily Flushing of Drinking Water Sources
To Reduce the Risk of Possible Lead Contamination**

Pursuant to the Federal Lead Contamination Control Act (PL 100-572) passed in 1988, Federal guidelines and testing protocols were established to determine the degree and source of lead contamination in school drinking water supplies, including water coolers, and suggested techniques for reducing exposure to lead. The law mandates parents, staff and employee notification of the "availability" of the test results annually. You will receive under a separate cover an "Annual Water Quality Report" which must be posted prominently in every school. Results are also posted on the district web site.

The Newark Public Schools has installed lead reduction filters on all drinking water and food prep sources in all schools constructed prior to 2006. They are maintained on a regular basis based on manufactures recommendations and quarterly random water testing reports. Report any filter observed with a date exceeding six (6) months from the last replacement to the Duty Manager. In order to further ensure lead levels are at acceptable levels the following steps are to be enforced at the school level:

PRINCIPALS

Instruct students and staff to run each fountain or drinking faucet for at least thirty (30) seconds before drinking.

CUSTODIANS/HEAD CUSTODIANS

Run and flush each individual water fountain for two (2) minutes daily prior to the opening of school and denote in your log book daily.

NUTRITIONAL SERVICES

Run and flush each cafeteria "cold" water faucet for two (2) minutes daily prior to food prep usage.

Procedures are in place to monitor and replace lead reduction water filters as needed as a part of the districts preventative maintenance program. These protocols should be continued until you are instructed to do otherwise.

Figure 7. Letter warning citizens that Morganza Floodway will probably be opened and providing instructions on what readers need to do.

Washington, D.C.: Government Printing Office. (2003). 1-140.

[productive-communication-channel-for-your-message-HA001194524.aspx](http://www.microsoft.com/en-us/help/choose-the-most-productive-communication-channel-for-your-message-HA001194524.aspx). (Acquired 2014) .

Dabbish, Laura A., Robert E. Kraut, Susan Fussell, and Sara Kiesler. "Understanding Email use: Predicting Action on a Message." Carnegie Mellon University (April 2-7, 2005). <http://www.cs.cmu.edu/afs/cs/Web/People/kiesler/publications/2005pdfs/2005-Dabbish-CHI.pdf> .

Flower, Linda. "Writer-based prose: A cognitive basis for problems in writing." *College English* 41.1, (September 1979) 19–37.

Herdnl, Carl, Barbara A. Fennell, and Carolyn R. Miller. "Understanding failures in organizational discourse: The accident at Three Mile Island and the shuttle Challenger disaster", in *Text and the Professions*, eds. C. Bazerman and J. Paradis. (Madison, WI: University of Wisconsin Press), 1991. 279-305.

Inouye, Randall R. and Joseph D. Jacobazzi. "The Great Chicago Flood of 1992." *Civil Engineering-ASCE* 62.11 (November 1992) 52-55. <http://www.oocities.org/nitekitty63/case-2.html> .

National Commission on the BP Deepwater Horizon Oil Spill and Offshore Drilling. *Deepwater: The Gulf Oil Disaster and the Future of Offshore Drilling: Report to the President.* Washington, D.C.: Government Printing Office. (2011).

Nielsen, Jakob. "F-Shaped Pattern for Reading Web Content." Nielsen Norman Group (17 April 2006) <http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/> .

Nielsen, Jakob. 1997. "How Users Read on the Web." Nielsen Norman Group (October 1997). <http://www.nngroup.com/articles/how-users-read-on-the-web/>

Outing, Steve. 2011. "Eyetrack III: What News Websites Look Like through Readers' Eyes." Poynter (24 August 2004). <http://www.poynter.org/2004/eyetrack-iii-what-news-websites-look-like-through-readers-eyes/24963/> .

Presidential Commission on the Space Shuttle Challenger Disaster. *The Report of the Presidential Commission on the Space Shuttle Challenger Accident*, Washington, D.C.: Government Printing Office. 1986.

Skovholt, Karianne. *Email Literacy in the Workplace*. Dissertation. Oslo, Norway: University of Oslo. 2009.

Stack, Laura. Choose the most productive communication channel for your message. <http://office.microsoft.com/en-us/help/choose-the-most-productive-communication-channel-for-your-message-HA001194524.aspx>

Contact Information

Carolyn Boiarsky
Purdue University Calumet
CLO 221
Hammond, IN 46324
+1 219.937.4736

Author Biography

Carolyn Boiarsky is a professor in the Department of English at Purdue University Calumet, Indiana. Her new book *Risk Communication and Miscommunication: Case Studies in Science, Technology, Engineering, Government and Community Organizations*, is due out in June, 2016. It is published by Colorado University Press in both electronic and hard copy versions. She has authored two previous books related to communicating in engineering and the technical sciences—*Technical Communication: Contexts, Audiences, and Communities* and *Writings from the Workplace: Documents, Models and Cases*, both of which were published by Allyn and Bacon. Boiarsky received the Society of Technical Communication's Frank R. Smith Article of the Year Award and has made numerous presentations at the annual conferences of the Society for Technical Communication and the IEEE/Professional Communication Society. She has also presented internationally in Limerick, Ireland and Milan, Italy, and was invited to present at the Engineering Education Conference in Sheffield, England.



BUILDING QUALITY EXPERIENCES FOR USERS IN ANY LANGUAGE

John Collins

It's easy to assume that writing English in a friendly, human tone and writing for localization are mutually exclusive activities, but they aren't. This paper looks at some content trends and then gives tactics, gained through the author's personal experience, for achieving both a friendly English tone and successful localization.

More and more companies strive for a friendly tone in their English content. At the same time, many companies are turning to international markets for expansion opportunities.

As a technical writer creating English content, you're setting yourself up for ridicule if you're not concerned about non-English readers.

You don't want your brand to be seen in the same way as you see Figure 1.



Figure 1. Let's assume that the translator of this sign meant well, even if results are a little passive aggressive.

Why it Matters

Many writers recognize Mailchimp, with its voice and tone website [1], as a leader in tech companies taking a more friendly, human tone with written communications.

It's not just young tech companies trying to be more human, though. Microsoft has also taken steps to replace its stuffy, computerized voice in written communications with a lighter, more colloquial voice. [2]

Seemingly, companies realize that tone—even in technical communication—is part of their brand. We see this as marketing communications and technical communications converge. [3]

“Your tone can help you stand out from competitors, communicate efficiently and effectively with your audience and share your personality,” wrote Robert Mills. [4]

“Like people, UIs communicate through their personality, tone, and attitude,” wrote Everett McKay. [5]

Technical communicators are crucial. According to one survey, over half of technical writers contribute to user interface (UI) content. [6]

Localization is the adaptation of websites and software for other languages and cultures. That's growing at the same time that there's increasing focus on tone and communication via UI.

For example, 56.2 percent of consumers say that the ability to obtain information in their own language is more important than price. [7] In 2013, 79 percent of users of the top 10 global websites came from outside America. By 2014, that percentage jumped to more than 86 percent. [8] Only 15 percent of the global population speaks English as a first or second language. [9]

You want your brand to reflect its personality in English and other languages. How can you achieve both a friendly English tone and successful localization?

Lay a Strong Foundation

Before you finalize your English content, you should already be building relationship with your language service providers. Try to screen translators and develop a “preferred” translator list.

As you work on your English content, you’re probably referring to a style. Before you send content for translation, you should give translators the same strong foundation. Work with your language service provider to create localized style guides for each language you’ll be translating to.

These style guides are similar to your English style guide, but they will be adapted for the special needs of each language. They will address linguistic issues as well as cultural considerations, like formality.

Develop a localized glossary so that translators will know how to handle key or common terms in their translations. Have a preferred translation for a term? Put it in the glossary. Have a term that you want to stay in English? Put it in the glossary. Are you coining a new term? Yeah, you guessed it. Put it in the glossary.

Like a style guide, you can base your main glossary on an existing resource—like the Microsoft Language Portal—and keep a list in a spreadsheet of your customizations.

Add Quality Content (Think like a Translator)

After you’ve put down a strong foundation of building relationships with language service providers, creating localized style guides, and creating a localized glossary, it’s time to create content.

Here’s a little secret: Translators can do a better job when they have quality source content.

One useful tactic is to think like a translator. To think like a translator, you need to:

- Watch your structure
- Write well

Watch Your Structure

The structure of your content—in the sense of content engineering—is critical to your ability to get good localizations. Consider the following structures:

- Variables and conditionals: Dynamic text is inserted within static sentences, paragraphs, or UI text strings. Err towards larger chunks of text for variables, like entire sentences instead of single words.
- Concatenations: Existing pieces of content—often UI text strings—get combined to create a new phrase or sentence. Don’t concatenate UI text strings.

Both are problematic because syntax differs across languages, and what works in English may fall apart in other languages. Sometimes translators can compensate, but that can often yield awkward results that aren’t their fault. Many times, though, there’s nothing a translator can do to fix problems introduced by variables or concatenations.

Another aspect of content structure to is how to re-use content. Content re-use is popular in technical writing, and it can save money on localization, but it needs to be done at the right granularity to avoid the same problems as you can run into with variables and conditionals. Make sure translators can understand the complexity of your source content.

Write Well

While solid structure enables easy localization, the quality of your source content paves the way for pleasing localizations.

You need to write clearly. Word choice is critical. Help your translators by using consistent terminology. Avoid synonyms.

Be wary of homonyms—words that have the same sound and spelling but different meanings. These are often, but not always, words that can be either a noun or a verb, like *date*, *engaged*, *leaves*, *type*, *stalk*. Similarly, watch out for homographs—words that look the same but change in

meaning with a change in pronunciation. Some homographs: *read*, *minute*, *does*, *wind*, *sow*, *bow*.

You can use homonyms and homographs, but try to realize when you're using them, and let translators know the definition you intend.

Finally, don't noun verbs or verb nouns. In business, you've likely heard *spend*, *ask*, and *invite* becoming nouns. In everyday life, you may have heard *fail* and *reveal* becoming nouns. The nouns *incentive* and *solution* are morphing into the verbs *incentivize* and *solutioning*, and *dialogue* is moving from noun to verb. [10] Verbing weirds language, as Calvin points out. [11]

Another secret: When your source content is high quality, it'll be friendlier too.

Provide Context

When it comes to translation and localization, context is king. The more context your translators have, the easier their job will be. Good context begets good translations. Bad context begets bad translations.

Think about a translator's job. They're often given content in a context-free format. When they get an item that says "Contact," how will they know if it's a verb for a button or if it's a noun for a label in an address book feature?

There are four main tools for providing context on a project (aside from the glossaries and style guides mentioned earlier).

- Localization project brief. Use the project brief to give a high-level explanation of your product or localization. It's also a good place for communicating requirements for vendors, such as: due dates, languages needed, specific character count requirements on translations, technical requirements for the delivery of translated files, and desired translator expertise (such as API translation, scientific translation, legal translation, or marketing translation).
- Reference materials. Provide translators with some means to visualize how the content will be used. You can send UI wireframes, mockups, screenshots, screencast videos, or existing documentation.
- Access to software. Ideally, translators would have access to the software being built. While the translators may bill you for that time, it will greatly improve their understanding of the

content they're working on. Many times, organizational or technological impediments may prevent you from giving translators access to your in-development software. If that's the case, your wireframes, screenshots, and screencast videos will become more valuable to the translation effort. Tools such as HockeyApp can help you distribute pre-release builds of your mobile apps. [12]

- Context comments. Give information to translators directly in your source files. If you're working with content in text-based code files like XML, HTML, JSON, .strings, .properties and so on, your best bet is just to use code comments to leave info for translators.

```
<!-- CONTEXT: This button takes the user
from the sign-up page to the home page.
-->
<string id="sign_in_button" text="Start
Now"/>
<!-- CONTEXT: This button takes user to
web-based form that they can use to send
Acme Corp. a message. -->
<!-- TRANSLATOR: "Contact" is a verb
here. -->
<string id="contact_button" text="Contact"/>
<!-- TRANSLATOR: The phrase "on the go"
is an idiom. It basically means you're very
busy, moving about, going from one activity
to another. We're using "on the go" in
English instead of "busy" because we prefer
how it flows. -->
<string id="easy_to_use_explanation"
text="The new vertical layout makes the
app easier to use when you're on the go."/>
<!-- CONTEXT: The variable (the curly
brackets) here is filled by only one value,
the product name. -->
<string id="about_link" text="About
{$productName}"/>
```

Figure 2. Example from an XML file

```
# CONTEXT: This button takes the user
from the sign-up page to the home page.
sign_in_button = Start Now
# CONTEXT: This button takes user to web-
based form that they can use to send Acme
Corp. a message.
# TRANSLATOR: "Contact" is a verb here.
contact_button = Contact
# TRANSLATOR: The phrase "on the go" is
an idiom. It basically means you're very
busy, moving about, going from one activity
to another. We're using "on the go" in
English instead of "busy" because we prefer
how it flows.
easy_to_use_explanation = The new
vertical layout makes the app easier to use
when you're on the go.
# CONTEXT: The variable (the curly
brackets) here is filled by only one value,
the product name.
about_link = About {0}
```

Figure 3. Example from a .properties file

```

/* CONTEXT: This button takes the user
from the sign-up page to the home
page. */
"sign_in_button" = "Start Now";
/* CONTEXT: This button takes user to
web-based form that they can use to
send Acme Corp. a message. */
/* TRANSLATOR: "Contact" is a verb
here.*/
"contact_button" = "Contact";
/* TRANSLATOR: The phrase "on the go"
is an idiom. It basically means
you're very busy, moving about,
going from one activity to another.
We're using "on the go" in English
instead of "busy" because we prefer
how it flows. */
"easy_to_use_explanation" = "The new
vertical layout makes the app easier
to use when you're on the go.";
/* CONTEXT: The variable (%@) here is
filled by only one value, the
product name. */
"about_link" = "About %@";

```

Figure 4. Example from a .strings file

If you're working with content in a spreadsheet, you probably have columns for English and your other localizations. It's easy to add a column for context notes next to the English column.

When possible, use all four tools together for best results.

Pulling it all Together

Now it's time to think about getting your tone right. Here are two approaches to getting friendly English while still prepping for localization:

- Write your English your way. Write your source content just as you would for your usual audience. Be human and casual. When you're done (or as you go), write plenty of notes to the translators, giving them context about what you've done. Explain the humor or idioms and provide alternate phrasings.
- Write and maintain two English versions. Start with a basic version in global or Simplified English. This is what you'll send to be localized. The trick for this method is that you're one of the translators. Working from the global English version, you'll translate into the human, idiomatic version that you want your users to see. You and your organization will need to decide how exactly to implement this workflow.

Translation or Transcreation?

Sometimes all you need from your language service provider is translation, where the goal is a faithful, accurate version of your content—in another language. Other times that might be too bland, so you might want to specify that

you want a transcreation, where the goal is an adaptation of your message that evokes the same reaction for readers in another language.

“The goal of transcreation isn't to say the same thing in another language. Indeed, it is often not possible to say exactly the same thing in another language.” [12]

This is when you'll be glad to have a strong relationship with your language service provider, healthy localized style guides, and plenty of contextual information.

Resources

- Cohen, Lior. “Localization – The key to success in the international market.” Net-Translators. <https://www.net-translators.com/knowledge-center/articles/localization-the-key-to-success-in-the-international-market/>.
- Collins, John. Strategies for Friendly English and Successful Localization. (Proceedings, STC Summit Conference, 2014).
- Globalization and Localization Association. “Why Localize?” GALA (2014). <https://www.gala-global.org/industry/language-industry-facts-and-data/why-localize>.
- Lionbridge. “How to Create a Translation Style Guide and Terminology Glossary.” Lionbridge (21 August 2014). <http://content.lionbridge.com/how-to-create-a-translation-style-guide-and-terminology-glossary/>.
- Mailchimp. “Voice & Tone.” Voice and Tone (2014). <http://voiceandtone.com/>.
- McElroy Translation. “Funny but Costly Localization Mistakes.” Translation Directory. <http://www.translationdirectory.com/articles/article2409.php>.
- McKay, Everett. UI is Communication (Waltham, MA: Elsevier Inc), 2013.
- Microsoft. “Microsoft Language Portal” <https://www.microsoft.com/Language/en-US/Default.aspx>.
- Microsoft. “Style and Tone.” [https://msdn.microsoft.com/en-us/library/windows/desktop/dn742477\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dn742477(v=vs.85).aspx).
- Mills, Robert. “Finding Your Tone of Voice.” Smashing Magazine (21 August 2012). <http://uxdesign.smashingmagazine.com/2012/08/21/finding-tone-voice/>.
- Multilingual Computing, Inc. “Bringing a new tone to CEE product localization.” Multilingual (March 2013). <http://cdn2.hubspot.net/hub/147991/file->

274228047-pdf/docs/bringing_a_new_tone_to_cee_product_localization.pdf?t=1460543732932.

References

- [1] Mailchimp. “Voice & Tone.” Voice and Tone (2014). <http://voiceandtone.com/>.
- [2] Multilingual Computing, Inc. “Bringing a new tone to CEE product localization.” Multilingual (March 2013). http://cdn2.hubspot.net/hub/147991/file-274228047-pdf/docs/bringing_a_new_tone_to_cee_product_localization.pdf?t=1460543732932.
- [3] O’Keefe, Sarah. “Trends in technical communication, 2014 edition.” Scriptorium (21 January 2014). <http://www.scriptorium.com/2014/01/trends-in-technical-communication-2014-edition/>.
- [4] Mills, Robert. “Finding Your Tone of Voice.” Smashing Magazine (21 August 2012). <http://uxdesign.smashingmagazine.com/2012/08/21/finding-tone-voice/>.
- [5] McKay, Everett. UI is Communication (Waltham, MA: Elsevier Inc), 2013.
- [6] Welinske, Joe. “Skills and Technologies Survey 2014.” WritersUA/Welinske.com (3 February 2014). <http://www.welinske.com/skills-and-technologies-survey-skills-14/>.
- [7] Globalization and Localization Association. “Why Localize?” GALA (2014). <https://www.gala-global.org/industry/language-industry-facts-and-data/why-localize>.
- [8] Chan, Vincent. “14 Compelling Statistics About the Importance of Localization.” One Sky App.com (7 August 2014). <http://www.oneskyapp.com/blog/14-compelling-statistics-about-the-importance-of-localization/>.
- [9] Cohen, Lior. “Localization – The key to success in the international market.” Net-Translators. <https://www.net-translators.com/knowledge-center/articles/localization-the-key-to-success-in-the-international-market/>.
- [10] Wasko, Brian. “Verbing Weirds Language: When Nouns Become Verbs.” The Write at Home Blog (13 September 2013). <http://blog.writeathome.com/index.php/2013/09/verbing-weirds-language-when-nouns-become-verbs/>.
- [11] Watterson, Bill. “Calvin and Hobbes” GoComics.com (25 January 1993). <http://www.gocomics.com/calvinandhobbes/1993/01/25>.
- [12] Bit Stadium GmbH. “HockeyApp: The platform for your apps.” HockeyApp (2014). <http://hockeyapp.net/features/>.
- [13] Arno, Christian. “Translation vs. Transcreation.” Articulate (30 November 2009). <http://www.articulatemarketing.com/translation-vs-transcreation>.

Author Contact Information

John Collins
Senior Technical Writer
Atlassian
www.linkedin.com/in/johncollins
jrc.collins@gmail.com
+1 540.246.9845

Author Biography

John Collins is a senior technical writer at Atlassian. Prior to that, he was at Rosetta Stone, the well-known language learning software company, for almost 8 years, including 5 years as the team lead of the technical communication and localization team. He’s written UI text, help systems, user’s guides, guided a move to single-sourcing, and led several major internationalization and localization projects, two of which included trips to Asia to train coworkers and translators. Collins enjoys speaking at conferences around the country.



SINGLE-SOURCING AND LOCALIZATION: MAXIMIZING CONTENT ACROSS LANGUAGES

Laura Dent

Single-sourcing means maintaining one source of content for multiple deliverables and audiences. Localization includes translation as well as the technical aspects of managing the content. When you localize single-sourced content, the benefits multiply: you translate the source files once, and create different versions from those source files. However, localizing single-sourced files introduces significant challenges. This presentation will demonstrate best practices for addressing the combination of single-sourcing and localization.

Benefits of Single-Sourcing and Localization

When you combine single-sourcing and localization, the benefits include:

- Consistency—Translated content remains consistent across products, brands, markets, and other variations.
- Ease of maintenance—When you localize your updated content, you maintain all versions in sync, rather than having to update every version of the translations when features change.
- Cost savings—One of the biggest advantages of localization and single-sourcing is the massive cost savings over translating individual pieces.

In a single-sourcing system, you start from a central system to produce variations of your content. Tools for single-sourcing include Help Authoring Tools (HAT), DITA, and content management systems. Evaluate your needs and resources to develop a single-sourcing system that is best for you. You can develop the expertise to implement these systems.

Single-Sourcing Techniques

Single-sourcing lets you provide variations in your content in several ways:

- Format—different output forms such as web help, mobile help, and PDF documents.
- Content—variations on the material such as different brands, products, or audiences.
- Purpose—different groups in your organization can repurpose the same content for different uses, such as documentation, support, and training.
- Region—some content might be location-specific.

All these factors can overlap, and the more variations contained in your single-sourcing system, the more you'll realize the benefits of localizing your single-sourced content.

Techniques for single-sourcing content include:

- Topic-based authoring—creating modular content that can be reused and combined in different ways
- Conditional text—turning a set of content “on” or “off” for different outputs
- Variables—setting a discrete word or phrase, such as a product name, that changes with each output

Writing and Communication

- Snippets—creating a whole chunk of content for reuse

Depending on your single-sourcing tools, you can create some or all of these features.



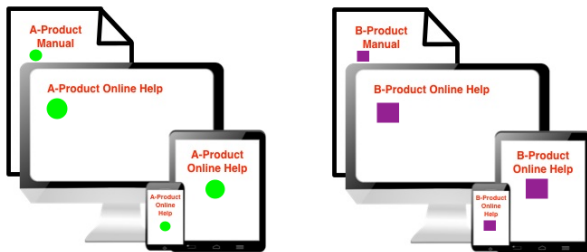
Figure 1. Use single-sourcing to create variations on the content and publish in multiple formats.

You can create a single-sourcing system at progressively more comprehensive levels of your organization:

- Within a document or set of files, usually within one group, such as the documentation department
- Among related groups, such as documentation, technical support, and training
- Company-wide, to maximize efficiency and consistency

Usually you start from one group and spread from there. The more buy-in you can get across the company, the

Documentation



Support



Training

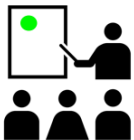


Figure 2. Repurpose single-sourced content throughout your organization.

more you'll realize the advantages of single-sourcing, especially when it comes to translating the content.

The efficiency and cost savings become even more advantageous when you localize the single-sourced content across groups or even your whole company.

Best Practices for Localizing Single-Sourced Content

If you are single-sourcing with localization in mind, you can follow these guidelines to help the process go more smoothly.

- Use variables with caution, paying attention to language differences. Variables such as <ProductName> can create language issues, as we'll see below.
- Use conditional tags on standalone content, such as a whole sentence or paragraph. For example, don't separate parts of the sentence such as "<adjective> <noun> <verb> <object>" because languages use different word order.
- Create snippets of standalone content, not for reuse within sentences where the grammar could change in other languages.
- Allow for text expansion. For example, your content in English will translate into much longer text in some languages, such as German. Changes in text length become even more of an issue when the text itself can change, based on variables and conditional text.
- Use layered graphics for localization of callouts (do not embed text in images—keep them separate to be localized).
- Review your content before and after localization to address language issues.

Linguistic Challenges

Some of the constructions you use in English to produce single-sourced content may cause issues in localization. Variables and conditionals that create variations in the outputs may cause grammar issues in localized languages.

Examples of linguistic challenges include:

- Articles and combination forms—for example, French combines the article “le” or “la” when the

following word begins with a vowel: l'exemple (the example). See the next section for more about variables with articles.

- Word order—for example, French often uses noun-adjective instead of adjective-noun; German and Japanese place the verb at end of the sentence.
- Gender—if the noun changes, the adjective form might have to change to agree with the noun.
- Number—in some languages, such as Russian and Japanese, the grammar changes depending on the number.

The Localization Process

Typically, you'll follow this sequence:

1. Create your single-sourcing system in English.
2. Have your single-sourcing system localized by your language services provider (LSP).
3. Produce the outputs in the new language(s), just as you did in English.
4. Update and maintain the localization as you expand your single-sourcing system.

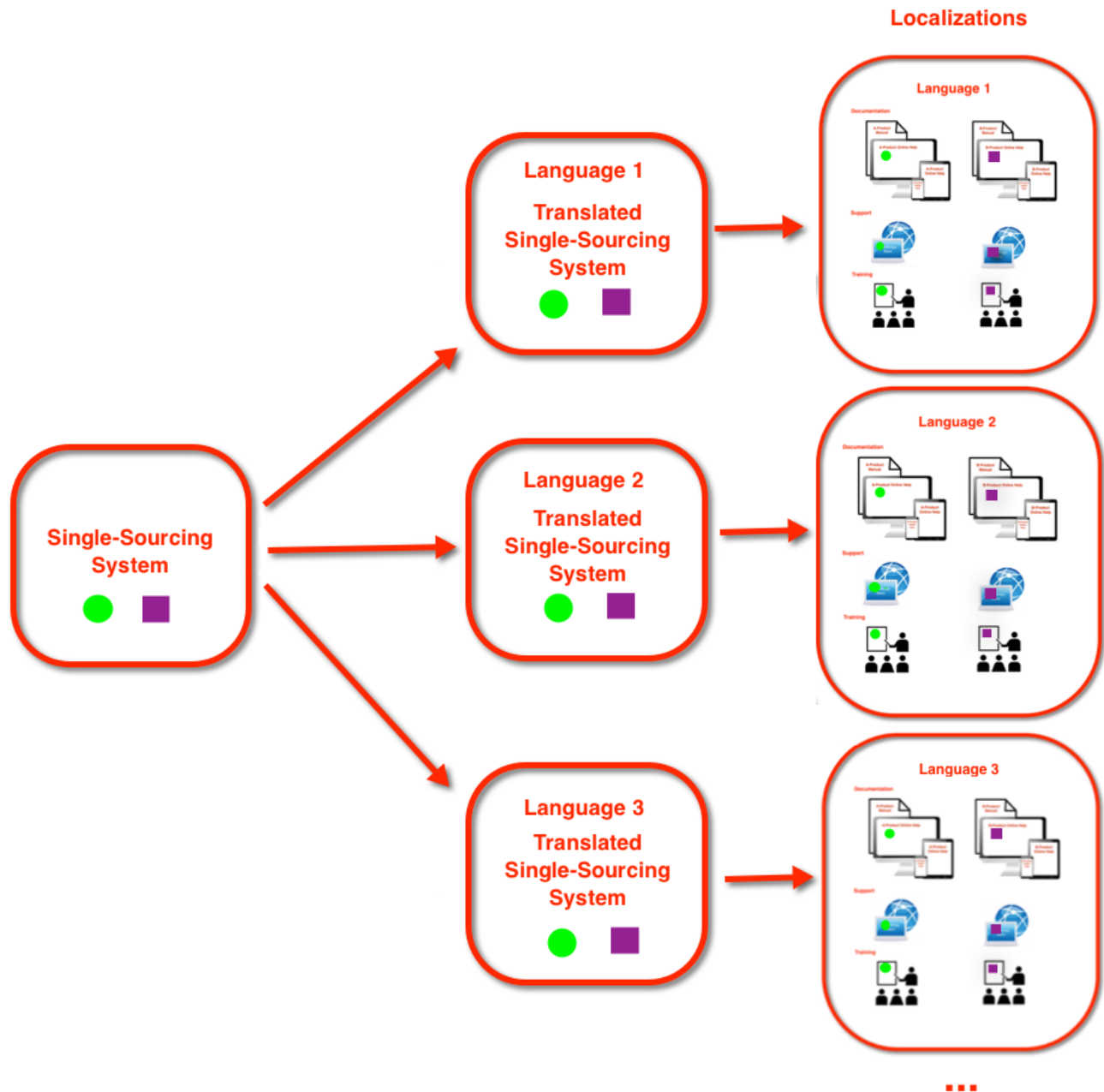


Figure 3. Localize your single-sourcing system to produce translated variations in content and format.

Writing and Communication

Once you have created a robust single-sourcing system, you can send your single-sourced files to your LSP.

Where before you might have translated each document and help system separately, now you can have your entire single-sourcing system localized just once. From that one localized system, you can produce all versions of your content in each language.

Author Contact Information

Laura Dent
Freelance Technical Communicator
156 Maryland Ave.
Harrisonburg, VA 22801
+1 540.383.7027

Author Biography

Laura Dent has several decades of experience working in technical communication, specializing in localization. As a freelancer, she provides her clients with localization of single-sourcing systems. Previously, at Rosetta Stone, she created systems that enabled SimShip (simultaneous shipment) of all software and documentation, an almost unheard-of achievement. Laura has a passion for languages, and speaks Russian, Italian, Spanish, French, and (a little) Japanese. She teaches technical communication to computer science majors at James Madison University.



SELF-PUBLISHING FOR PRINT AND EBOOK DISTRIBUTION

Richard Hamilton

Self-publishing, using Print on Demand (POD) technology and readily available eBook tools, has revolutionized traditional publishing, making it possible for anyone to publish a book or eBook. The same technology that will let you publish a memoir of your kindergarten years will also let you publish more down-to-earth content, including your technical documentation.

This paper provides guidance to companies that want to publish content in print or eBook form and sell that content through online retailers. It describes the steps you need to take to create deliverables from existing material and distribute them for sale in print and eBook form. You can also use these instructions if you want to distribute a book privately, for example, to offer a book exclusively to your customers.

The self-publishing process has four steps: creating content, creating deliverables from that content, distributing content, and marketing. This paper discusses the middle two steps.

Why Self-Publish Documentation?

The usual motivation for self-publishing is personal; you have a novel or memoir that you're sure a traditional publisher would reject, or you don't want to go the traditional route. However, the same techniques and platforms used to self-publish a novel can work just as well for a company that wants additional avenues for delivering content to customers.

Print on Demand (POD) technology removes the need to plan print runs, warehouse books, and fulfill customer orders. POD companies take orders directly from retailers, print the copies ordered, deliver those copies to the purchaser, and send you the net revenue (after retailer discounts and printing costs). Once you have provided your content in the correct form, the POD company handles the rest. This makes providing printed content to your

customers significantly easier than it has been in the past. Taking the process one step further and offering eBooks extends your reach with relatively little additional effort.

Selecting Material

The following categories of content are good choices for self-publishing:

- Content that was previously in print, but has been moved online for cost reasons, even though it might be better presented in print. Examples include reference content and conceptual content.
- Value-added content. Examples include programming guides, internal information, and other content that could be pulled together into a book. The Dummies Series, the Missing Manual Series, and the Complete Idiot's Guides are models of this kind of book.

In evaluating whether you should sell content, there is a fine line between offering added value at a fair price and charging customers for information that they really should have as part of your product or service. In my experience, customers who want content in printed form will pay for a printed book, even when the same content is available online for free. But they will balk if essential content is only available for purchase. You need to carefully consider the content you are considering publishing to make sure it delivers value beyond what a customer would expect from the standard documentation.

Selecting Vendors

If you anticipate selling a large number of titles, you should either work in partnership with a traditional publisher or select POD and eBook distribution companies that specializes in working with publishers. For print, XML Press uses Lightning Source/Ingram, and, for eBooks, we use O'Reilly Media. However, neither of these distributors works with non-publishers. To work with either one, you would need to convince them that you will be publishing more content than the typical self-publisher and that you have sufficient technical resources so you won't overburden their support.

Unless you are in that category, you should consider a consumer-oriented Print on Demand (POD) distributor. In nearly all cases, they provide the best balance of cost and return.

The following POD companies distribute print self-published books from individuals and companies:

- Lulu: <http://lulu.com>
- CreateSpace: <http://createspace.com>
- IngramSpark: <http://ingramspark.com>

There are tradeoffs in ease of use and cost between the three, but they are relatively similar. Lulu is generally the easiest to use, but returns the least revenue for sales. CreateSpace and IngramSpark provide somewhat better revenue and are a little harder to use. CreateSpace is owned by Amazon, but offers the option to distribute through channels outside Amazon. IngramSpark is owned by Ingram, which also owns Lightning Source; however, IngramSpark is designed for self-publishers. I recommend visiting all three websites and reviewing their offerings.

The three companies above also distribute eBooks. In addition, you should consider the following options:

- Smashwords: <http://smashwords.com>
- Individual retailers: Amazon, Barnes and Noble, Kobo, Apple, Google, etc.

Smashwords specializes in taking books written in Microsoft Word and creating ePub books. They distribute through most of the major eBook retailers, with the exception of Amazon.

Many individual retailers will allow you to distribute directly; for example, you can choose distribute through Amazon, Google, or Apple exclusively. If you prefer to distribute through only one retailer, this can work well. However, if you want to distribute through several retail-

ers, you will be better off working with a distributor; each retailer has its own requirements, and you will likely be overwhelmed with small, but critical, differences if you try to work individually with more than one or two.

Finally, you may want to investigate selling or distributing eBooks directly from your own website. In that case, you won't need a distributor, but you will still need to create deliverables and support users as they download and read your books.

I recommend that you consider eBook distribution at the same time you consider print and select a vendor that will handle both.

Lulu, CreateSpace, and IngramSpark also provide print services without distribution. This can be convenient if you need books for marketing or other purposes, but don't plan to sell them. And, even if you use a POD vendor's distribution services, you can still purchase copies at cost for marketing or other activities.

Setting the Price

Pricing for most distributors is relatively simple. All of the printers and distributors named earlier have calculators that will tell you how much you can expect in net revenue from a print book based on the following variables: retail price, discount to distributors, number of pages, interior color (full color or grey scale), size of the pages, hardcover versus paperback, and the quality of the paper. For eBooks there are fewer variables: retail price and size of the file are the main ones.

The only other variable to consider is whether you want this content to generate revenue or whether you would rather make it available as cheaply as possible to maximize sales volume.

The same calculators can also tell you what the cost will be if you are buying copies for internal use.

Obtaining ISBNs

Any book sold through a retailer must have an ISBN (International Standard Book Number). If you do not intend to sell your content through a retailer, you can ignore this section.

Some self-publishing vendors will provide an ISBN as part of their service. The upside is that you don't have to purchase an ISBN, but the downside is that the vendor will

be listed as the publisher, not you. If that doesn't matter, then you might as well use their ISBN. However, if you want your company to be listed as the publisher, you need to obtain an ISBN. In the United States, Bowker (<https://www.myidentifiers.com>) is the only official agent. In other countries, you can find the agent by searching the website of the International ISBN Agency: (<https://www.isbn-international.org/agencies>). In some countries, ISBNs are free, and in others they must be purchased.

You will need one ISBN for print and another for the eBook; however, you do not need separate ISBNs for each eBook format. If you are publishing more than a few books, I recommend obtaining enough ISBNs to assign what you will need for the next few years; in the US, ISBNs are cheaper when purchased in bulk.

If you obtain an ISBN in the US through Bowker, you also have the option of listing the book in their database. I recommend doing this.

Preparing the Deliverables

For print, POD vendors will expect two PDF files: one for the front and back covers and one for the interior (everything except the front and back covers). For eBooks, distributors will expect a single file in one of several formats, the most common being ePub and mobi (for Kindle). This section looks at the steps required to prepare these files.

Preparing the Cover

For print, each of the vendors provides templates for creating your cover, and some provide automated tools that will generate a generic cover with some customization. The templates are typically generated from forms that take the page size, the number of pages, and the ISBN and generate a template that can be edited using a tool such as Inkscape or Adobe Illustrator. The automated tools typically take the same information, the text you want on the cover, and an optional cover image; they then generate a complete PDF for the cover in a form that their tools can use.

No matter how you create the full cover, if you are also generating an eBook, you will need to create a .jpg or .png image of just the front cover.

Keep in mind that the cover of your book, print or eBook, will be displayed in a thumbnail size on retailer's sites. Make sure the title text remains visible when you view the cover in that size.

Preparing the Print Interior

The two challenges in preparing the interior of your book are creating a book structure (front matter, including the table of contents, and back matter, including the index) and creating output formats for print and eBooks. In this paper, I will discuss the latter challenge. Tools such as Framemaker, Indesign, and Microsoft Word provide support for the former, as do the XML tool chains for schemas, such as DITA and DocBook, that are designed for technical communication.

Each vendor has guidelines for creating the interior. If you have access to a high-end page definition tool, such as Framemaker or Indesign, you should be able to generate a PDF that follows any of the guidelines out there.

If you use Microsoft Word, you will need to do more work, both to create a book with the necessary structure and to generate a PDF that will satisfy vendor requirements. The upside of using Microsoft Word is that Smashwords, Lulu, and CreateSpace all have guidelines that will make it easy for your Word file to be converted into an ePub. Here are links to their guidelines:

- Smashwords: <http://smashwords.com/books/view/52>
- Lulu: <http://connect.lulu.com/t5/eBook-Formatting-Publishing/eBook-Creator-Guide/ta-p/109443>
- CreateSpace: <https://forums.createspace.com/en/community/docs/DOC-1482>

If your process uses DocBook or DITA XML, you can generate a book, with all the required parts, using the tools associated with those schemas. You will probably need to do some post-processing using a tool such as Adobe Acrobat Pro to meet vendor requirements.

Preparing the eBook

For eBooks, the typical process is to generate an ePub file, then, if you need the Kindle version, use Amazon's kindlegen tool to generate a .mobi file, which is the format used by the Kindle. You can generate that ePub file in several different ways, including using an authoring/production tool, tools provided by your eBook distributor, or a dedicated tool.

If you want to distribute eBooks only through Amazon, then you can use Amazon's tools to generate a Kindle file from Word, HTML, or ePub. In that case, the simplest path is through Kindle Direct Publishing (<http://kdp.ama->

zon.com). If you want to distribute through other vendors, then your choices are to use Microsoft Word (Smashwords, Lulu, and IngramSpark all offer conversion services that take Word as input) or ePub. If at all possible, I recommend using ePub. You will have more control over your content, and because ePub is primarily an HTML-based format, you can use any existing HTML expertise in your organization.

If you generate an ePub, you can convert that ePub into the Kindle format using the kindlegen tool from Amazon or the open source Calibre tool (<http://calibre-ebook.com>). Calibre converts between a wide range of file formats and allows you to do a certain amount of editing, so it is a valuable tool to have if you work with eBooks in any format.

Once you've completed your deliverable files, you upload them to your vendor(s). The last step is checking proofs to ensure that the result is what you expected. The print vendors will send you a print proof, which you can check. In my experience, the most important things to check are the cover (are the colors correct and are there any typos), fonts (especially be sure to check any text that uses characters other than ASCII), and images.

Conclusion

Self-publishing technologies offer new ways to deliver content to your customers. There is still enough variation among competing vendors that you will need to do some research to make sure you pick the best one for you, but the investment shouldn't be too large, and the benefit of being able to offer your customers content in print and eBook form can be large.

Resources

Kindle Direct Publishing. Publish on Amazon Kindle with Kindle Direct Publishing (Amazon Digital Services), 2011.

Kawasaki, Guy, and Shawn Welch. APE: Author, Publisher, Entrepreneur: How to Publish a Book (Nononina Press), 2013.

Castro, Elizabeth, EPUB Straight to the Point, (Peachpit Press), 2010.

Castro, Elizabeth, Pigs, Gourds, and Wikis (Elizabeth's blog). <http://www.pigsgourdsandwikis.com>.

Shepard, Aaron, New Self Publishing (Blog about self-publishing). <http://www.newselfpublishing.com>.

Author Contact Information

Richard Hamilton
Publisher
XML Press
24310 Moulton Pkwy, Ste O-175
Laguna Hills, CA 92637
+1 970.231.3624

Author Biography

Richard Hamilton is founder and CEO of XML Press. He began his career at Bell Laboratories developing computer software and has worked at AT&T, Unix System Laboratories, Novell, and Hewlett-Packard in jobs ranging from software development to product management to documentation management.

In 2008, he founded XML Press, which is dedicated to producing high quality, practical publications for technical communicators, managers, marketers, and the tools builders and engineers who support their work.

Richard is the author of *Managing Writers: A Real-World Guide to Managing Technical Documentation*, and editor of the 2nd edition of Norm Walsh's *DocBook 5: The Definitive Guide*, published in collaboration with O'Reilly Media.



APPLYING MINIMALIST WRITING PRINCIPLES TO IMPROVE USER'S GUIDES: READERS AND WRITERS WIN

Michael Harvey, Fellow

At SAS, we reduced page count and improved clarity when we applied minimalist writing principles to documentation. We formed a team to promulgate these principles throughout the division. Our team provided strategies to follow these principles when revising existing and writing new material. With minimalist documentation, readers can use software more expediently. Applying minimalist writing principles makes content more suitable for consumption on mobile devices.

At a minimalist writing workshop that I and some colleagues attended, our first exercise was to critique one of our own documents. We were asked to determine how many pages it took to find information that enabled you to do real work. Not become familiar with the software, or understand concepts, but actually do a task. Sad to say, it took 41 pages before I could find a real task in my document. At that point, I knew that I was going to benefit from the workshop. Little did I know that I would become an advocate for minimalism within my division.

What is Minimalism?

Based [John M. Carroll's](#) research at the IBM Watson Research Center, minimalism facilitates work on real tasks. It promotes learning by doing rather than learning by reading. To apply minimalism to writing, you use fewer words in action-oriented text. You cut back on descriptions and eschew useless information. Useful information shows rather than tells. Minimalist writing eliminates ROT – the redundant, outdated, and trivial.

The four minimalist writing principles are:

1. Choose an action-oriented approach. Emphasize tasks before you explain concepts and ensure that tasks follow a real workflow.
2. Focus on users' real goals. Anchor everything that you write in the task domain rather than the tool domain. For example, do not lead with tabs or menu items (“the Home tab enables you

to change fonts”); lead with what you can do through them (“Change fonts through the Home tab”).

3. Support error recognition and recovery. Provide steps to recover from mistakes whenever errors are likely to occur, and provide carefully labeled troubleshooting steps within procedures.
4. Support information access. Readers view technical content as a series of answers to questions that they have about the product. Craft your headings, labels, and topic sentences accordingly.

An important goal of minimalist writing is to create information tools rather than content containers. The best way to achieve that goal is to optimize content for search: organize topics around user roles and activities.

This does not mean that we should completely abandon descriptive or conceptual material. Instead, we should allow readers to opt-in to it, providing it in sidebars or appendixes. Motivated readers opt-in whenever they think they get something that they want. Similarly, chapter tables of content are useful only when they provide the reader a relevant workflow. Design TOCs as if they were quick reference cards. This means using action verbs and nouns that are relevant to the task domain, rather than the product architecture.

Applying Minimalism to My Writing

I found opportunities to apply minimalism to my writing almost immediately after finishing the workshop, some small and some large. For example, I had written the following passage in one of my documents:

For more information about assigning roles to nodes, see <xref>.

The <xref> resolved to “Assigning Roles to Nodes.” So in the finished document, the following appeared:

For more information about assigning roles to nodes, see “Assigning Roles to Nodes” on p. 26.

That passage was the result of rote technical writing habits that I had developed over the years. But now I was keenly aware that no one needs that type of repetition. So I made the following change:

“For more information, see <xref>.”

It was a small change, but when I repeated it a dozen times throughout my document, it had a large impact on its scannability.

I also was able to apply minimalism in a more substantial way. In my one of my User’s Guides, I had written several sections in an early chapter that explained how to explore something called a “risk cube.” In a subsequent chapter, I provided a step-by-step example to create and explore a risk cube. Applying minimalism, I removed the explanatory sections from the early chapter. I retitled the later chapter “Working with Risk Explorations: An Example” and wove in helpful material from the early chapter. After all, I was covering the same material in both places. I kept the example because it showed in a direct, task-oriented way, most of what the sections in the early chapter had explained. What I did not cover in the example, I covered through EUA or made sure that it could be intuited by exploration of the UI. Thus, I eliminated around 20 pages. Readers got to learn while doing, rather than simply reading.

Promoting Minimalism within Our Division

With management support, some of us formed a team to promulgate minimalist writing principles throughout our division. One of our biggest challenges was guiding writers away from the almost academic style inculcated over years of writing about statistics. That approach was understandable when users willingly read books about topics such as logistic regression and the programming elements

to use in order to run statistical analyses. These days, users are impatient, rarely if ever investing time in reading a technical book from cover to cover. If after scanning a few lines these users cannot discern whether information is giving them what they need, they move on to something else. Minimalism is the best approach to write prose for the too-busy-to-read user.

To anchor our work, we produced a web page to serve as a resource for information about minimalist writing, which we determined should be emphasized as “customer-focused” writing. The page provided information about the following points:

What minimalist writing is – an approach to create documents that quickly provide customers what they need in order to do real work.

How to think like a minimalist – for example, when your document has a getting started chapter, make it the first chapter instead of the product description. What is the product’s workflow? (What do you do first? Then what? And then what?) Make the organization of your document follow that workflow.

How to write like a minimalist – for example, move conceptual information to the end of a task or chapter, or maybe even to a separate document. Remove information about industry standard buttons and common tasks. Distill instructive narrative into numbered steps.

How to work with SMEs - they might resist focusing on how to do something with the software rather than on what the software does – for example, when a SME asks you to put technical details about product operations in an awkward place, ask them “could we provide a link or cross reference to that information instead?”

External resources – for example, [ISO/IEC CD 26513 Annex A, User-centered test and review guidelines](#).

We held a number of training sessions after finalizing the web page. One session was captured on video for ongoing reference. After that, the division’s management team made implementation of minimalist writing principles a performance goal for the coming year.

As our division continues to move away from a linear, book model to a topical, help center model, we know that we need to create information that can be readily consumed on multiple platforms. Those who consume information from tablets or other mobile devices are especially keen to find critical information quickly. They are perhaps the least patient of our impatient readers. These consumers benefit from the simplification, content

reduction, and emphasis on relevant tasks that we achieve through minimalist writing.

Readers and Writers Win

In an interview in STC Intercom, John Carroll suggests that we think of readers as “learners” rather than “users.” He said that “(readers) need to act, they need to be engaged” as they learn. As we write, “we have to exert a certain amount of self-control, empathy, and...user-testing...to make sure we do not lapse into decoration and obstacles” to this learning. In order for readers to act, we writers need to provide clear directions about what to do, and not much (if anything) else. Readers win when we do.

A quote attributed to Pascal captures one way that writers can internalize minimalism: “if I had more time, I would have written a shorter letter.” At first, the self-control that Carroll encourages requires more time than usual. With practice, though, minimalism becomes second nature. It becomes easier to provide readers only what they need in order to do relevant work. As a result, our prose becomes crisper and more direct. We writers win when we become better at our craft.

References

- Bleiel, Nicky. “Minimalism Revisited: An Interview with John Carroll.” *Intercom* (February 2013): 7-13.
- Carroll, John M. ed. *Minimalism Beyond the Nurnberg Funnel* (Cambridge, MA: The MIT Press), 1998.
- Minimalism: Creating Information People Really Need*. Comtech Services, Denver, Colorado. 2013.

Author Contact Information

Michael Harvey
Principal Technical Writer
SAS
100 SAS Campus Drive
Cary, NC 27513-2414
+1 919.677.8000

Author Biography

Michael Harvey is a principal technical writer at SAS, serving as documentation project leader for financial risk management solutions. He also teaches Information Architecture for the Duke Continuing Studies Technical Communication certificate program. 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. His LinkedIn profile is <http://www.linkedin.com/in/mtharvey>

Michael has served in various leadership positions for the Carolina chapter of the Society for Technical Communication (STC) and has presented at local and international STC conferences. He was honored to be named an STC Fellow in 2011.

Michael is an avid runner, having completed 15 marathons and 10 half-marathons as of May 2016.



FRENEMIES: TECH COMM AND TECH SUPPORT WORKING TOGETHER

Rick Lippincott, Associate Fellow

Technical Communicators and technical support team members have the same goal: ensuring that users get the information that they need, and that it is correct. But at times, it seems as though the two organizations are working at cross-purposes. This paper takes a look at the problem, and offers some hope. It discusses the differences and similarities between the two organizations, the advantages of being a technical communicator embedded in a tech support organization, and questions whether or not they are truly “frenemies.”

They’re the Same, Only Different: Tech Comm and Tech Support, Compared and Contrasted

Technical communications and technical support share a common goal: assisting the user. However, the two organizations use different methodologies because their relationship to the product and their methods of contact with the users are entirely different.

Similarities

- Both organizations may be more likely than engineers to see the users as real people, with real problems.
- Both are driven by a desire to improve the customer experience.
- Both strive to deliver clear, accurate, and useful information about operating the product.

Differences

- Technical communicators have to present all of the possible contingencies and cover all aspects of the product, because they do not know which specific task a user will need to do at any given time. They are aware that some of the procedures

(for example system failure recover) they write may never need to be used at all.

- Technical support teams are required to deal only with immediate and urgent topics, because their action has been triggered by a customer problem. They know exactly what task the user needs at the moment. They may frequently deal with the same topic over and over, and never get questions at all concerning some topics.
- Technical communicators are normally always looking at the next release, in order to have documentation ready when the product ships.
- Technical support is always dealing with the last release, because that’s what is in the field and is what the customers are using.

Bus vs. Taxi

Mark Baker has compared the tech comm/tech support differences to a bus vs a taxi. Both are methods of transport available to people as needed, but the concept of operations differs.

- The tech comm bus has to accommodate many passengers going to different destinations. When passengers board the bus, the driver does not know the passenger’s destination. The bus makes a series of stops, based on the designed route, and

has to hit all of them. Passengers may or may not get off at those stops, but the bus has to stop anyway. The route the bus takes is designed for the efficiency of the route, but may not be the fastest route from a give Point A to Point B.

- The tech support taxi need only accommodate one passenger at a time. The passenger tells the cab driver exactly what the destination is. The taxi makes only one stop per passenger, and the passenger gets off. The route the taxi takes is designed for efficient travel and ideally is the fastest route from start to destination.

Frenemies: Competitors or Collaborators?

There is a conventional wisdom that tech com and tech support, even though they both exist to help the customer, are competitors in a certain way. Tech comm teams may have a concern that a customer reliance on tech support may sway management to minimize documentation efforts: “Let them just call the help line, we’ll have support answer the questions.” On the other hand, tech support team members may be concerned that perfect user documentation may reduce the number of incoming calls, and thus reduce the need for tech support, or reduce the income to the company if we are charging for tech support.

This is not supported by real-world examples, and quality documentation is actually an aid to the tech support team. Reasons include:

- Technical support may be included in the product purchase, effectively paid for up front. Anything that can be done to reduce the number of support calls actually makes more financial sense for the company.
- The tech support team may be overworked, and clearing up some common, frequent problems can help reduce their stress.
- Reduced calls by eliminating the most-common, easily solved problems allows tech support more time to focus on more difficult user problems.
- For hardware systems that require field service and repair support, complete and accurate documentation is vital to troubleshoot the problem, determine what replacement parts may be needed, and reduce the repair time.

A second concern is that, due to repeated calls on a specific subject, tech support may be aware of issues in the documentation or may have developed a clearer way of explaining a specific process, but does not send that

information back to tech comm. Tech comm may perceive this as “wanting to keep errors in the docs to increase the need for support.” The reality, however, may be that the company’s corporate structure may make communication between the two organizations nearly impossible. The support team and the writers may not even know who each other are.

Embedded: Working on Tech Comm from Within Tech Support

The most common structure for a technical communications organization is to be embedded as a part of the engineering development organization. This presents obvious advantages: writers are closer to the subject matter experts, writers are in the loop regarding latest changes and additions to the product, writers are clearly aware of the product development plans.

The disadvantage is that the writers are further removed from the users, the writers may not be aware of specific user or even customer support/service needs, and the writers have a more difficult time getting access to direct customer input.

Organizational Structure Comparisons

In the graphics that follow, the symbols mean the following:

- M: Upper Management
- E: Engineering Division
- S: Tech Support Division
- Red bar: Engineering organization
- Green bar: Tech Comm
- Blue bar: Tech support

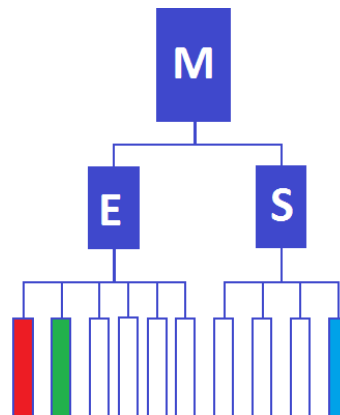


Figure 1. Tech Comm in Engineering

When the tech comm group is located in the engineering division, contact with engineers is easy and frequent. This makes it easy to remain current on new developments and changes, but formal communications and requests for changes from tech support may have to go through upper management

When the tech comm group is located in the technical support organization, it's easier to get relevant information on issues with the fielded products and also issues with the existing documentation. This aids in the process to correct errors, organize the documentation flow to reflect real-world usage, and focus on the activity that is most useful to the user. But this comes at the expense of frequent communication with the engineering group, and the danger of being left out of the development group.

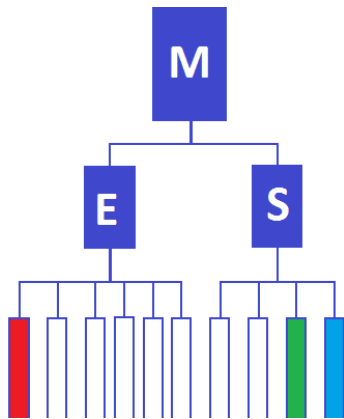


Figure 2. Tech Comm in Tech Support

What Can Be Gained?

There are many advantages to having tech comm and tech support working within the same organization.

The first and most obvious advantage is that this type of organization puts tech comm closer to the users, and thus able to more easily accommodate the needs of those users. Writers can learn directly from tech support about the portions of the documentation that are good, bad, wrong, or missing.

A second advantage is nearly a corollary of the first: knowing the problems that tech support teams are encountering can be used as information to improve the next product design. For example, one company's customers had requested addition of a file archive utility. The writer, embedded with field support, got in on the discussion and offered "The feature already exists, it's just hard to find on the GUI." The customer got an immediate answer, and the

field service manager immediately started pressing software development to come up with a better GUI design.

Tech support is the organization on the front line of customer contact, and sometimes it is the first to learn that customers are using products successfully, but in ways other than the engineers or marketing teams had planned. Tech comm, as part of the team, can take the information and incorporate it into the documentation so that the techniques devised by some can become the knowledge used by all.

What Risks Being Lost?

The biggest risk of loss is, of course, frequent and daily contact with the people designing the next series of products. Unfortunately, locating writers with one group will always mean that they lose some contact with the other. On the other hand, it's reasonable to believe that in successful companies the management will always recognize the importance of tech comm and engineering group communication, and will help preserve communication even if the groups are separate. In other words, when the writers are in the support group it's easier to get info from engineers than it is to get information from support when the writers are in engineering. "I need to know the latest information from the engineer to make the product release deadline" is more easily understood as a problem than "I need the most common system failure from tech support to update a service manual" when the problems are presented to management.

Support documentation for hardware is unfortunately often placed low on the list of priorities for doc development. From the engineering perspective, the emphasis is on the operation of new system, not methods to support or maintain it. Then, once product rollout is over, budgets for documentation gets slashed, and there's no funding to create service documentation. However, when writers are assigned under tech support, their manager has not only the goal of supporting the product rollout with the operating documentation but also the incentive to push for development of support documentation.

Despite the good news mixed in with the bad news in the two paragraphs above, tech comm teams assigned to tech support should be prepared to do extra digging and questioning to ensure that nothing gets lost. Techniques to ensure this include:

- If your company is using the Agile method of development, be sure to get involved in the process. If necessary, invite yourself to the daily scrum.

Writing and Communication

- Maintain friendly contacts with engineering in order to find out what new items are coming up.
- Be sure to find out when preliminary or critical design reviews are taking place, and be at the meetings.
- In short, be visible...or be forgotten.

Hardware Writing: Additional Issues

Larger hardware systems that require field service and maintenance have additional considerations that can be better resolved when the writers are working within the technical support organization. These include:

The field service/repair team members are likely part of the support organization, so this is another group of documentation users located just a few desks away.

Field service/repair team members are often eager to present ideas to tech comm regarding ease of use and accuracy. Suggestions can include improvements to presentation, layout, organization of the material, and ideas about “We need more of this but we don’t always need as much of that.”

Field service/repair team members are your SMEs for creating troubleshooting and failure analysis documentation. Everyone involved wants the field service team to diagnose the problem as quickly as possible, and get to repairs. Their input can aid your task to build troubleshooting and fault isolation documentation.

Field service/repair team members may have good reasons for embracing new technology. For larger and complex hardware systems, field service manuals may be large, bulky, and multi-volume. The field service engineers could have good reason to push for transition to mobile devices as the documentation media. One iPad is easier to carry than a half-dozen three-ring binders.

All of these make the field service engineer’s job easier and faster, thus reducing cost and benefiting the budget.

Conclusions

Tech comm and tech support can and should work together more closely. Both share common goals of explaining process to the customer, both strive to make the user’s experience less painful and more productive. Perceived friction and failures in communication often exist mainly due to corporate organizational structures which tend to inhibit the transfer of information between

two corporate groups that have actual contact with the users. Overcoming these barriers may be difficult at times due to management misunderstanding and a resistance to change, but when done the outcome is improved documentation, improved customer technical support, and happier customers.

References

Baker, Mark. “Users’ Advocate: You Can’t Please Everyone” Techwhirl-1 <http://techwhirl.com/users-advocate-you-cant-please-everyone/>

Grey, Jim. “Software Technical Writing Is A Dying Career (but Here’s What Writers Can Do to Stay in the Software Game)” Stories from the Software Saltmines <https://softwaresaltmines.com/2015/06/16/software-technical-writing-dying/>

Author Contact Information

Rick Lippincott
Technical Writer
120 Pilling Street
Haverhill, MA 01832 USA
+1 508.662.6428
RJL6955@gmail.com

Author Biography

Rick Lippincott is an Associate Fellow in the STC, and former President of the STC New England Chapter (and its predecessor, STC Boston Chapter). He was the guest editor of the April 2016 issue of Intercom (“Grand Hardware” theme). He normally documents hardware systems (often very very large hardware systems) including aerospace, telecommunications, ion implantation, networking systems, large-scale X-ray inspection systems, and CT medical imaging systems. Rick has an MS in Technical Writing from Rensselaer Polytechnic Institute, and a BS in Journalism from Ohio University.

SIMULTANEOUS PRODUCT AND INFORMATION DEVELOPMENT

Janice Manwiller

During any development cycle, technical communicators are challenged to have the information piece of the product prepared as quickly as possible after the code is complete. Using a more iterative, incremental process that begins with the product design, you can significantly reduce the lag time, simplify the review process, generate a more polished information product, and provide valuable resources for other teams.

Highlights of the Process

So at a high level, the key elements of the process are:

- **Creating most content using a less formal, more widely accessible platform**—I do most of my real writing and editing work using a shared platform such as wiki pages or Google Docs. Only when it's time to finalize the feature do I start working in the authoring tool.
- **Editing content incrementally and iteratively**—I start with an outline based on the initial product requirements, then add and update details gradually as the product development continues.
- **Participating actively in the development process**—I attend design and review meetings, and

track design documents, bug resolutions, and code check-ins.

Tracking Information Development to Product Development

Grooming Requirements/Planning the Information Updates

Each release starts with the product team deciding what new functions and features are being added. This then triggers a set of meetings to groom those requirements. The grooming process sets the boundaries for each new feature, and reveals possible issues and roadblocks.

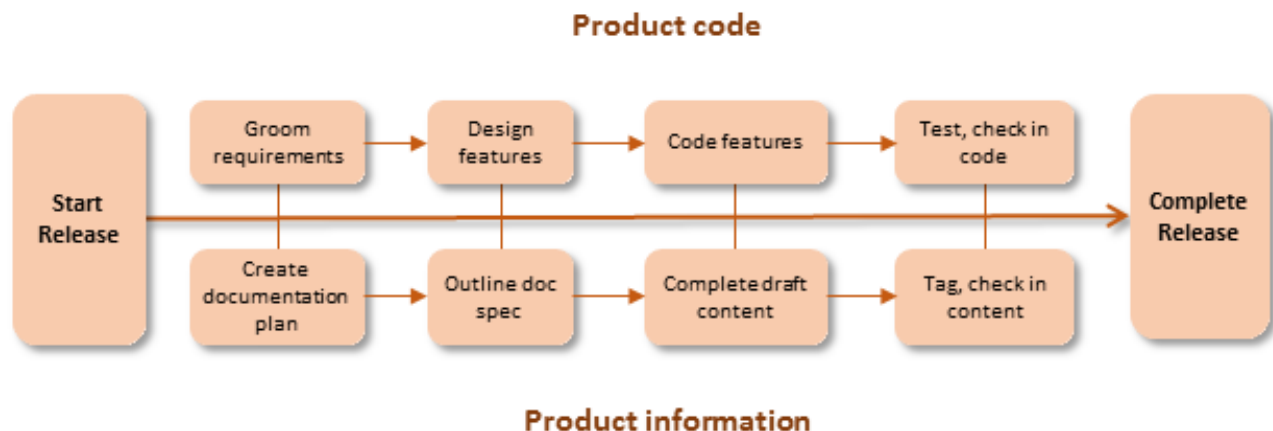


Figure 1. Corresponding release process flow for product code and information

Writing and Communication

For the information part of the product, each release starts with a wiki-based documentation plan, which I update throughout the release.

In the documentation plan, I list the planned product changes, and then for each change:

- Provide a brief description of the new or revised function or feature
- Summarize the corresponding information updates
- Add links to design documents, issues in the tracking system, and draft content

See Figure 2 for example.

Designing New Features/Starting the Draft Content

When grooming is complete, the next step is to complete the design for each feature. Developers write design documents, and the UX team creates wireframes and visual designs. Review meetings confirm that the design matches the requirements, and catch any missing or incomplete details.

On the information side, I start creating the draft content (also referred to as a doc spec) for the new features and functions. While I currently use Google Docs, in the past I've also used Word documents and wiki pages.

When I first create a doc spec, it's just an outline with some notes.

In addition to the content itself, each doc spec includes:

- A summary of the new feature or function
- Links to the associated issues in the tracking system

- The location of the new or revised content in the documentation set
- A summary of the content changes

Developing the Code/Fleshing out the Draft Content

As the development continues, I flesh out the information in the doc spec. This includes:

- All of the text content
- Any new or revised diagrams

By the time the code is ready to be checked in, the corresponding information is also ready to go, except for the final screen captures.

If possible, I also have developers use the doc spec for their reviews, using the Google Docs commenting function to provide feedback.

Checking in the Code/Tagging and Checking in the Information Updates

When the code changes are about to be pushed into the code repository, it's time to finalize the information.

I drop the draft content from the doc spec into my authoring tool, tag it using the appropriate tagging language (such as HTML, DITA, or DocBook), and add the final images.

I usually check the final tagged content into the documentation repository within a day of the code being pushed.

Item	Doc Status	Dev Status	Description	Doc Impact	Doc Task
Using regex for model mapping (PRODUCT-34)	✓	✓	<p>New regex mapping ability. Condition expressions can include a regex - this also means that users will also be able to use regex expressions in any SELECT query WHERE clause.</p> <p>The regex expressions can include capture groups to capture values, which can then be mapped to entity features or relationship values.</p> <p>Regex mapping design note</p>	<p>Configuring and Loading Sqrrl Data - Add information on using regex to map model data.</p> <p>Exploring and Querying Sqrrl Data - Add new regex capability for SELECT query WHERE clauses.</p> <p>Reference Guide - Update the entity JSON to reflect the new regex mapping for syslog data.</p> <p>Draft content</p>	PRODUCT-855

Figure 2. Example entry from a release documentation plan

Resources for Keeping up with Product development

Traditionally, “interviewing SMEs” is considered a primary resource for information about new product features and functions.

In my experience, while I of course need to ask questions or get clarification from developers or product managers, actual “interviews” have been fairly low on the list of ways to keep informed.

The following are some timely, direct, and in some cases automated methods for getting information.

Attending Daily Standups

If you work with a development team that uses some version of agile methodology, then it’s likely you already attend the daily standups.

These daily updates provide the current status of work in progress, to get an idea of how close development work is to completion.

Attending Design and Review Meetings

While meetings may be the bane of most people’s existence, as an information developer, you want to attend as many meetings as possible, including:

- **Design meetings and reviews**—Design meetings don’t just tell you the nuts and bolts of where the fields and buttons are going to go. They also provide insight into the thinking behind the design, deepening your understanding of the product. These meetings are also an opportunity to contribute suggestions for usability and clarity. I’m often the one to answer questions about the current functionality, and to verify existing standards for text and labels.
- **Functional reviews and handoffs**—At the other end of the process, attending reviews when the work is complete or close to complete allows you to verify and correct the draft content you developed based on the original specifications and design meetings.

Tracking Changes to the Code

Another valuable source of information is the code repository.

If you get automatic notifications when code is pushed to the repository, you’ll know to look over the latest changes so that you can finalize your draft content.

Ideally, you have been contacted during the development process to review messages and labels,

If that hasn’t occurred, then you can also review new text as it’s checked in in order to suggest updates to clarify the text, use standard terminology and wording, and correct basic errors.

Tracking Progress of Bugs/Tasks

Your bug and task tracking system is another source of up-to-the-minute information.

For larger stories and tasks, the initial requirements are a good place to start when putting together your first outlines. If you use the same tracking system for both code and documentation tasks, you can link the documentation tasks and stories to the development tasks.

For agile environments, you can use the tracking boards to keep up with progress.

For bugs and smaller enhancements, which may not have full design specs or reviews, the bug tracking system may be your primary source of information.

In my case, all issues go through a doc review process to verify whether a resolved issue requires any changes to the documentation.

Monitoring Team Chats

If your development team uses an IRC or other chat tool, be part of the conversation.

In addition to using the chats as a source of information, you can also become an information resource yourself.

The more you know about your product, the more you can make yourself someone to go to for answers.

Why Work This Way?

So how can this process benefit both the product and the information you develop?

- **Timely information assets**—Completing the information piece of the product at or very close to the same time as the code prevents bottlenecks in the release process.
- **Easier reviews**—Reviewers can easily get to and comment on the draft content. Depending on the tool you use, they can even see each other's comments and ask each other questions. Completing the review outside of the main authoring tool or final output allows them to focus on the actual content, and to not be distracted by style and formatting.
- **Built-in editing pass(es)**—Building up the content gradually gives you multiple opportunities to fine-tune it. As you add each next layer of information and become more familiar with the feature you're working on, you may realize that you may need to tweak the organization or correct an example. The final transfer to the authoring tool also provides a built-in editing pass. As you're dropping in and tagging the text, you can catch some final typos or awkward phrasing.
- **Available assets for other functions**—When new features become available, the next step is usually for QA to start testing them. If the information is completed at the same time as the code, QA can immediately test that the information is accessible, clear, and correct. Training groups can also take advantage of having initial content available early as they work on training plans and materials.

Since then, I've worked as a technical writer at mostly smaller software companies, including a couple of interactive television startups.

Over the years, I've come to focus on:

- Technical communication as a visual profession, both in the sense of using visual elements (graphics, diagrams, videos) to better present information and ensuring that text can be visually parsed more easily.
- Seeing myself and acting as a developer who creates the information piece of the product, as opposed to creating the documentation that goes with the product.

Author Contact Information

Janice Manwiller
Principal Technical Writer
Sqrrl Data, Inc.
125 Cambridge Park Drive, Suite 401
Cambridge, MA 02140

Author Biography

After a couple of years of using my English degree to eke out a living in legal publishing and television listings, I quit and got my MS in technical communication from Rensselaer Polytechnic Institute.



EDITORS AND SMEs: BUILDING PRODUCTIVE WORKING RELATIONSHIPS

Ann Marie Queeney

Your success as a technical editor depends on the quality of your working relationship with SMEs (subject matter experts). Creating and maintaining a productive relationship takes effort and planning. However, it is worth it. This paper describes practical, proven techniques and tips the author has used as a procedures editor in the medical device industry.

Like other relationships in your life, building a productive working relationship with SMEs takes time, patience, perseverance, and an open mind. Your efforts will be rewarded with an enriched working environment. You may also discover opportunities to expand your influence in your organization.

Every editor's work environment is unique. Therefore, not all the tips in this paper may be applicable or feasible for your situation. Use this paper as a starting point for strengthening your working relationship with SMEs.

Adapting to THE SME's Preferred Working Style

We all have our preferred working style that helps us perform at our best. Adapting to the SME's working style does not mean abandoning your personal working style or trying to be someone else. The goal is to create a mutually beneficial, productive working relationship.

Identifying the SME's Working Style

Use your observation skills to notice how SMEs interact with you and others.

Here are some factors that contribute to a SME's working style:

- Personal characteristics. These items, such as personality, learning preferences, and cultural

background, make each person unique. A self-described morning person is more likely to be alert and eager to discuss a project at 7:00 a.m. rather than 4:00 p.m.

- Profession. Individuals who work in specific professions share similar personalities and communication styles. When I started working in the medical device industry, I soon noticed that engineers would sketch out diagrams and process flows when explaining their projects to me. I used their preferred approach in my meetings. As a result, I often received all the information I required.
- Company culture. Although mostly unwritten, a company's culture strongly influences how individuals in an organization interact. If you are new to a company, take time to learn the company culture.

Adapting to the SME's Working Style

Develop an approach to working with SMEs, based on your observations. A simple and effective approach is to ask SMEs about their work preferences.

It is important to work with your company's established processes and culture. Depending on your company, these processes may be difficult to change. Therefore, your changes may need to be incremental.

I was fortunate as I had autonomy to develop an effective approach. I edited document submissions as part of a change control process. The SMEs were responsible for

Writing and Communication

the content, so I did not edit directly on their documents. Instead, I listed the results of my review in an email, citing the step number. While this may sound like extra work, it actually was an efficient method.

The SMEs used my email as a checklist, making the necessary changes and recording their responses in the return email. The SMEs found this format and process helpful. The emails also served as a helpful checklist for me as I could easily review the SMEs' updates.

The important point is that I developed a process of working with SMEs – emails for most reviews and in-person meetings for complex or new projects – with SMEs. Since your goal is to build a partnership, including your SMEs in the process can be beneficial.

Adapting to SMEs' working styles is an ongoing process. Changes in your company's environment or personnel can affect your approach. Review the processes you have established, and adjust them as needed.

Prioritizing Editing Remarks

We have heard the saying, "choose your battles." In the editing world, that means prioritizing your review. It does not mean that you compromise the quality of your work. Instead, make a distinction between issues that must be addressed – those that affect understanding or compliance – and issues that would improve the document, but are not necessary for its understanding.

To use my working method as an example, I used my editing email to prioritize my comments into the following three categories:

- **Required—For Your Review.** These items included wording that affected the use and integrity of the document, such as ambiguous and unclear wording. The SMEs knew they must address these items, either by correcting the wording or explaining why the text did not require change.
- **Optional—For Your Consideration.** The SMEs could choose to accept or not accept my comments. While the SME's writing was clear, these suggestions would improve the document.
- **Typos and Minor Corrections.** This section included any typographical errors and misspellings.

Other approaches include writing a summary that emphasizes key points of your review or adding brief explanations with your edits.

SMEs can feel overwhelmed or frustrated when faced with blocks of seemingly unrelated edits, which leads to the next topic – SME fatigue.

Identifying Signals Of SME Fatigue

Whether it is a face-to-face meeting, email, or another method of communicating your edits, the keys to addressing SME fatigue are preparation and being aware of the SME's responsibilities, deadlines, and workload. This section focuses on face-to-face meetings and email communications.

Face-To-Face Meetings

Your work begins before your meeting. Come to the meeting with the document(s) thoroughly edited and prepared with questions.

While it is common to have follow-up meetings or discussions, do not have multiple meetings because there were loose ends and you were not prepared. Many SMEs are generous with their knowledge and time, but they become frustrated – rightfully – when an editor is not adequately prepared.

In a face-to-face meeting, you have the advantage of observing body language as an important cue signaling fatigue with the topic. Signs that the SME is losing interest include restlessness, a stressful or tired look, and a glazed-eye look indicating they are overwhelmed. Do not ignore your instinct, your gut feelings. If you feel that you are losing the SME's attention – you probably are. When you see signs of fatigue, reprioritize your agenda or suggest another meeting to complete your review.

There are several ways that editors can proactively avoid SME fatigue.

First, briefly outline the meeting's purpose and topics at the start of the meeting. Engineers and other SMEs feel more comfortable with structure.

Second, discuss the issues that require or benefit from a personal discussion upfront when everyone (including yourself) is fresh. If you end the meeting without covering all your topics, you can provide the SME with the remaining comments that do not require personal discussion.

Finally, complete meetings within the scheduled time. If you do not complete your review, you can ask the SME if he or she is available to stay beyond the scheduled meeting time. However, that should be the exception and not the rule.

Email Communications

Again, preparation is key. Complete your review before emailing your comments. Just as SMEs become frustrated with multiple meetings and follow-ups, they also find multiple emails counterproductive and annoying.

Organization of your email is also important. Use your technical communication skills to prioritize your review. I list required information first, followed by optional information. I also organize the information so it is not a block of text. Opening an email that seems to go on forever is overwhelming. Organize the content in a way encourages SMEs to read your comments.

Other Communication Methods

If your company uses other communication methods, the following tips can ensure a productive review, regardless of the method:

- Perform a complete review before contacting the SME.
- Use your technical communication skills to prioritize and organize your review.
- Complete meetings within the scheduled timeframe.

Expanding Your Contributions

Editors have skills they can use successfully in other initiatives. Look for opportunities where you can use your communication, organizational, and analytical skills to help SMEs reach their goals.

The following suggestions can help you strengthen your working relationship with SMEs:

Become a resource beyond your editing skills.

- Example: If you work with a broad spectrum of areas, use your knowledge of your company to become an organizational resource.
- Example: Your knowledge of specific software programs can position you as the local specialist.

Offer to present skills training.

- Example: Create a class on good writing practices.

Share STC appropriate Intercom and Technical Communication articles and blog posts with non-editors.

- Example: I shared an STC article on internship programs with the Human Resource Manager who was responsible for managing our company's internship program.

Conclusion

You can transform your working relationship with SMEs into a mutually beneficial partnership. The goal is to identify the SME's working style and find ways to compliment and work with his or her preferred style.

Author Contact Information

Ann Marie Queeney
Owner
A.M. Queeney, LLC
P.O. Box 513
Annandale, NJ 08801
+1 908.892.5156

Author Biography

Ann Marie Queeney of A.M. Queeney, LLC specializes in procedures for the healthcare industry. She has over 18 years of document specialist experience in the medical device industry. Her services include technical writing, editing, compliance assessment, and skills training.

Ann Marie, a senior STC member, is currently co-manager and past quarterly program manager of the Technical Editing SIG. She is also a member of the Consulting/Independent Contracting and Policies & Procedures SIGs.



USING PDF AS SOURCE

Liz Roscovius and Mike Sawyer

What do you do when your client provides only a PDF as “source” for your project? Run with it! PDFs are more usable than many of us realize, and clients can provide better quality than they may think. This article covers methods of extracting usable text and graphics from what’s usually considered a read-only format. We also cover how to determine the typefaces used, how to export or extract text, how to isolate bitmap and vector artwork, and even how to determine what tool was used to create the actual source file. PDF may not be the ideal source format, but it’s definitely workable.

Why are PDFs used so much? The utility of a PDF lies in the name itself: Portable Document Format. It’s portable, displays layout, fonts, and graphics perfectly well on any platform, and can be produced by any application that prints on those platforms. It’s also size-efficient and adaptable. Where an entire file tree (and its links) must be maintained for valid HTML, a PDF is self-contained and easy to archive. And, with the latest version of Acrobat (Acrobat DS), a PDF is even becoming much more editable.

Also, PDFs are also simply much more available when working with several-levels-of-separation mergers and acquisitions or overseas OEMs. They are a rock-solid deliverable format that can be easily replicated and transported on websites, by email, on flash drives, etc. For good or ill, they have longevity. As such, they are often the only file version of the document available for re-work.

Challenges

As versatile as PDF output is, it can present some challenges if that’s all you have to start with for revision work. They’re difficult to use because, like image files, they’re not really made to be editable. They’re a target deliverable format, generated from another piece of software that actually handles the layout and that may or may not be available. When you begin a document revision, you want to start with the source file set, but when you think “source,” you just don’t think “PDF.”

Assumptions

This paper’s instructions are going to assume you have some familiarity with PDFs, desktop publishing, and have access to a recent version of Adobe Acrobat. Keystroke instructions use Windows PC-based nomenclature, but we acknowledge that many in the reading audience use Macs.

Password Protection

A PDF can be password-protected against viewing, editing, copying, printing, changing form fields, and more. Various security methods are used (certificate- or server-based verification), but for standard password security, a quick removal of some of the PDF’s code will provide unfettered access. Several online tools are available for bypassing the password (Google is your friend, here, not us). The rest of the instructions in this article assume you have full, unrestricted access to the PDF contents.

Collecting File Information

Having detailed information about the PDF you are working with can help you know what to look for, what to ask for, and to make informed decisions about how to get the most out of it. For example, knowing who generated the PDF can help you track down the working file, particularly if the document was created by someone in another department within your company. Knowing what software was used to create the file could be useful. Knowing what

Writing and Communication

fonts were used could help you recreate the document's visual look accurately. You can find out all that and more. Simply open the PDF and right-click on the document, then scroll to Document Properties, or press Ctrl+D.

From the Description tab, you can see the author, the document creation date, and when it was last modified. You can also see what software the document was created in. The Security tab tells you what, if any, security may have been applied to the PDF. The Fonts tab lists fonts used in the document.

Exporting Entire Document

Options

Acrobat DS allows export into several formats, including spreadsheet, PowerPoint, image (JPG, TIFF, PNG), HTML, RTF, PostScript, plain text, and HTML. Because many variables affect the exported result, the best way to extract content will ultimately need to be determined by trial and error.

Pros and cons of Acrobat exports include:

- Microsoft Word and RTF: The most adaptable (and immediately usable) format. Could include formatting that you may or may not want to utilize in your new document. Header and footer information may be retained as body paragraphs.
- Spreadsheet: Best for exporting tabular data. Otherwise, it's not very useful (or pretty).
- Image: Export looks identical to the source page, but, as a raster image, it leaves you with the same problem as before--it's not really editable.
- HTML and XML: Quality varies widely, and it retains page headers and footers. Some cleanup in an HTML or XML editor will likely be necessary.
- Plain text: Strips all formatting, so useful as a source for "clean" text ready to be formatted with no legacy format issues.

Common problems with any whole-document export:

- Layout might not be retained.
- Column breaks could be interpreted as single lines with tabs instead separate vertical flows.
- Each line break could be changed to a hard return, which might look fine in an identically formatted document, but most of the time will break lines

in the wrong place in your new document. You'll have to strip out most of these hard returns.

- Acrobat tries to do something with every element on the page, which may result in random-looking placement of page numbers, page headers, and graphic captions and callouts.

How To

Perhaps the most adaptable export format is to Microsoft Word. In Acrobat DC, click File, Export To, Microsoft Word, then Word document. Open the Word file, then copy the desired text or graphic to paste into the new document.

Importing Entire Document

Some desktop publishing programs, including CorelDraw and the ubiquitous MS Word, can import PDFs directly.

How To

To import a PDF into Word, simply open it using Word's File menu. You'll see a dialog warning you that some documents won't import very accurately, then it'll do its thing. It works well on basic content, but doesn't handle watermarks well, it may change fonts, and it rasterizes and downsamples vector artwork. As with most conversions, the more complex the layout of the original, the more likely some fidelity will be lost.

To import a PDF into CorelDraw, click File, then Import. Select AI - Adobe Illustrator from the Files of type list box, select the PDF file, then click Import. In the Import PDF dialog box, select Text to import text as text (possibly resulting in font substitution problems), or select Curves to convert text as non-editable outlines that preserve the original appearance.

Extracting Usable Text

Extracting Editable Text

The easiest way to export only editable text (text stored as text, not as a graphic) is with basic text selection operations: clicking and dragging to select, or pressing Ctrl+a.

Pros: Quick and easy. Later versions of Acrobat will copy some of the text-level formatting, as well, which may or may not be what you want.

Cons: PDFs may not store your text flow exactly the same way as it was originally. Clicking and dragging over a selection of text can select the intended text along with snippets from all around the page. Also, line breaks will likely be interpreted as hard returns, which must be stripped out for the text to flow right in the new document.

Extracting Noneditable Text

Non-editable text in the PDF, or text that was scanned, isn't usable in that format. You can re-type, but you're not here to learn how to type. Luckily, Acrobat has a built-in OCR (optical character recognition) tool for handling just that.

In Acrobat DS, select the Edit PDF tool. Acrobat immediately sets to work and tries to recognize separate text and graphic blocks, and then runs OCR on the text blocks. The default interpretation language is Acrobat's current default, so after it runs OCR, it gives you the opportunity to change language and character dictionaries.

Its OCR also has another neat trick. If the scanned page isn't quite straight, it will read and convert the text, then straighten it so it's level. It doesn't just recognize the text—it actually replaces the image with editable text and cleans up the appearance, making it easier to select. Thereafter, the PDF is no longer a scanned document—it's actually been reconstructed, and can now be exported in any of the several formats discussed earlier. The drawback: It significantly lowers the quality of the graphics, and exported graphics quality is lowered even further.

To test the limits of its OCR, we took a photo of text on a computer screen, tilted, and at a skewed angle. It didn't do too well.

Extracting Usable Graphics

Graphics can be extracted from a PDF in a number of ways. Some utilize the same method you might use to extract editable text. The quality and usability of each graphic will depend on a number of factors, including how they were created, what desktop publishing software was used to create the original document, and how the PDF itself was generated.

Perhaps the easiest method of extracting a graphic is to export to Word. Whole-document extracting was covered earlier. That process extracts all content, both text and graphics. You can also save each graphic as a separate file, then import into the new document.

PDFs can also usually be opened directly in a desktop publishing or graphics program such as Adobe Illustrator. Further modification may be possible, but again, that will depend on how the graphic was created, how it was pulled into the working file, and how the PDF was generated.

You may also be able to copy an image directly from a PDF, but this is not always an option as availability may depend on the graphic format (raster vs. vector) and how the graphic was inserted into the original document.

How To

Select the Selection tool (solid arrow). You can right-click individual graphics, then select Save As. Raster (bitmap) images are more likely to be selectable than vector images. Early versions of Acrobat, though, won't let you select a specific graphic this way.

You can also export the page into a usable format, then select and save the image from the exported tool (such as MS Word). Note: Although a vector image exported into Word will retain most of its vector attributes, curves can get divided into several small, straight segments, so further cleanup will still be required.

For the highest quality export of vector images (especially EPS and AI graphics), open the PDF in Adobe Illustrator. Specify the page to open (it must be done page-by-page), then select the graphic you need and isolate it into its own graphic file. Note: Clipping paths are often created on a vector graphic within the PDF, so you might have to release multiple clipping paths to begin editing the graphic.

You can also export the page as an image, then crop it, or crop the PDF page itself within Acrobat to isolate the graphic, then export.

As a last resort, if you want the high-quality vector image but can't get it exported as a vector, you can zoom in, take screen caps of the image, then stitch them together using a raster editing program like Photoshop to create a high-resolution proxy of the vector image. It won't be editable as a vector graphic, but it will print and display very well.

Identifying Vector vs. Raster

Being able to extract a graphic from a PDF does not necessarily mean it is editable. If you need to change a graphic, it's important to understand what kind it is.

Graphics can be either vector or raster (sometimes referred to as bitmaps). Raster images are composed of pixels (tiny squares) that can each be a different shade or color. Common raster images are photographs and screenshots. Popular raster file formats include JPG, PNG, TIFF, and GIF. They can provide rich detail and offer precise editing. At the same time, because of the color information contained in each pixel, they also tend to have large file sizes and can become pixelated or blurry when enlarged.

Vector images are made up of lines, paths, and shapes. One example of a vector image is a product line drawing. Common vector file formats include EPS and AI. The strongest feature of vector images is that they can be resized without losing image quality because the shapes can be recomputed at any zoom level. They are usually editable and typically offer a smaller file size. On the other hand, they offer limited color detail and effects.

Converting Raster to Vector

Several tools exist for converting raster images to vector images. One of the most common is Illustrator. Place the raster image into a blank Illustrator artboard, select the image, then click the Object menu, Image Trace, then Make and Expand. Results vary widely and will depend on many factors, including what file format was used in the original document.

Recap

A PDF, although not considered “editable,” can actually provide a smorgasbord of content that can be sliced, diced, de-boned, and layered back together to help you cook up a new, nearly identical version you can edit using true desktop publishing tools. All it takes is some patience and perseverance. Even more techniques exist, which we'll cover in the STC Summit session, as well as add to the online version of this article stored at www.sawyerhome.net/stc2016. To see visual examples of these methods, please join us at our presentation at the STC Summit 2016.

Author Contact Information

Liz Roscovius
Director of Global Communications
D2
P.O. Box 644
North Sioux City, SD 57049
+1 605-232-4170 x117
liz.roscovius@d2worldwide.com
<https://www.linkedin.com/in/lizroscovius>
[@LizRoscovius](#)

Mike Sawyer
Documentation Manager
Control4
92 West 1565 North
Orem, UT 84057 U.S.A.
+1 385.209.7164
<https://www.linkedin.com/in/mikesawyerprofile>
[@Akambe](#)

Author Biographies

With nearly 20 years of experience in the communications field, Liz Roscovius has managed information development efforts, such as technical documentation, marketing support and communications research, for a variety of global retail and industrial clients. Her background includes expertise in visual design and marketing, content strategy, visual design and marketing, along with project management and process compliance. She's earned nearly 40 awards in professional competitions, with several advancing to the international level.

As director of global communications for D2, a US-based company providing documentation services for customers around the world, she leads a diverse team of technical writers, editors, graphic designers, copywriters, videographers, and project managers. Their shared goal: create communications materials that reflect the brand, target the right audience, and fit the product or service.

Liz holds BA degrees in art, from the University of Wisconsin, and in mass communications from Briar Cliff University. She also holds a MA in communications and journalism from the University of Nebraska.

Mike Sawyer has worked as a technical writer and manager for over 20 years. He'd list all of his accomplishments, but he's so humble that he won't. He loves technology and language, and sees technical communication as a “translation” between the technical and user worlds. He has a BA in communications from Brigham Young University and

completed coursework for his masters in human factors engineering from the University of South Dakota. He lives in Utah, enjoys sleeping in, and has five kids and too many cats.



IMPROVING THE REVIEW CYCLE: CONCURRENT EDITING

Mike Sawyer

Sending drafts out for review is relatively easy, but managing the overlapping—and sometimes conflicting—feedback of multiple files is a challenge. Making reviews concurrent makes them truly collaborative. Session covers workflow and best practices for concurrent editing in general, and specifically using Acrobat tools (online and local server-based), Word (using track changes and sharing), Google Docs, SharePoint (using online app version), and a little-known web app. You’ll also learn how to consolidate feedback from separately marked-up files.

Scope and Overview

The purpose of this article is to get you up and running quickly with collaborative reviews, so only common tools and best practices are covered. It covers types of collaboration, the basics of these interactive reviews, and some specifics for the most common tools. Links are provided for further research.

What are Collaborative Reviews?

Collaborative reviews bring team interaction to a review. At its best, it can be considered a team-based, interactive, high-visibility effort.

- Team-based: Individuals doing things together instead of separately.
- Interactive: Not a one-way conveying of input, but a dialog.
- High-visibility: Difficult to ignore. Accountability for completing a review, notifications of deadlines, and notifications of additional comments to your annotations.

Types of Collaborative Reviews

Like empowered, bleeding edge, and scalable, “collaborative” is a term that holds more promise than it usually delivers. It’s not the fault of the concept; it’s the fault of its misuse. In a quest for more accuracy, we can think of collaborative reviews as being done in three general approaches:

- Distributed (or “shotgun”)—This is the easy (and popular!) email blast. Starting one is a cinch. Processing it and synthesizing the results is a hair-pulling stress inducer. This approach complicates the Q&A between team members because, since most reviewers respond to the sender instead of the body who can answer the question, you are thrust into the role of moderator. You are asked questions you cannot answer, and even more communication burden is placed on your already drooping shoulders. In short, reviewers using this method can appropriately ask each other questions and resolve any issues. But they usually don’t.
- Consecutive (or linear)—Whether hard copy or an email attachment, this draft is passed along a chain of reviewers until all give the thumbs up. This approach naturally applies additional pressure on reviewers, since the next reviewer may be (hopefully) chomping at the bit for their turn. It also invites more collaboration, since each succes-

sive reviewer must encounter each of the previous annotations (and, hopefully, evaluate them). The most substantial advantage is that a single copy of the draft is passed along, and you (or the writer) will have a single set of comments to process and apply. Drawbacks of this approach include a very lengthy review cycle and early reviewers not having visibility to annotations of the successive reviewers.

- Concurrent (or parallel)—This is the ideal, and it involves all reviewers being able to review and discuss the copy at the same time (or within a given time span), making this potentially the fastest review type. It requires the help of a software interface to avoid collisions in a single file that has multiple humans reviewing it at the same time. It enables you, the administrator, to “freeze” the feedback at any time. Because it uses a file stored centrally, it reduces email burden and allows dialog to happen within the document instead of in outside email strings or chat windows. Its analog equivalent is a very long meeting that includes all reviewers sweating it out in a stuffy conference room, each with highlighters and pens, working with a single printout in the middle of a long table, with steady progress punctuated by arguments and spitwads. It’s not a group hug, but it gets the review done with maximum involvement, and the results will speak for themselves.

Making Distributed Reviews More Effective

Even the shotgun approach can be made more effective. If you must use this, try these tips:

- If you send the email using Microsoft Outlook, attach a reminder to prevent it getting lost in reviewers’ In Boxes. When composing the email, click the **Message** ribbon, click **Follow Up**, then click **Add Reminder**. You can select a **Start date**, **Due date**, and a **Reminder** date.
- Make sure Track Changes is on for any MS Word document you send for review. When you receive the multiple markups back, click the **Review** tab in Word. Then, in the Compare group, click **Compare**, then click **Combine revisions from multiple authors**, and follow the prompts to finish. Do this process for each of the returned documents, combining all feedback into a single master file. Then you can compare and evaluate the changes more easily. You can also re-distribute the combined file so all reviewers can see each others’ comments.

- When you receive multiple copies of a reviewed PDF, use Acrobat to combine all comments into a single master file so you can evaluate the annotations more easily. In Acrobat Pro, open the original file, then click **Comments**, then **Import Comments**. Follow the on-screen prompts to select and combine the several reviewed PDFs into one file. You can also re-distribute the combined file so all reviewers can see each others’ comments.

Making Consecutive Reviews More Effective

Going consecutive is a step in the right direction. It’s not ideal, but it does encourage more participation than the shotgun approach and allows some reviewers to easily see markups of those who have already completed their task.

If you’ve ever plopped a hard copy draft onto the desk of a reviewer, you know the value of a cover sheet/checkoff list. It’s in their face, they see it’s their turn, and they know who’s next in line. It’s accountability, and it’s a progress tracker.

You can do the same with soft copy. For example, you can create a single, full-page checkoff sheet as a PDF. This sheet could have your logo at the top, along with blank rectangles, underscores, or table rows for project name, product name, date, name of the originator, and for each reviewer. In Acrobat, create form fields for each of those information blanks, along with “Completed” check boxes for each so that you can use and re-use the cover sheet. Attach it to the front of each draft you send for review, fill out the information in each of the fields, then attach that review packet to the email you send to the first reviewer. They can annotate as usual in Acrobat (or Acrobat Reader), check their “Completed” box, and forward it on.

Managing Concurrent Reviews

First, a dose of reality: Although this review method is theoretically the most ideal, it is a new process outside the comfortable email realm, so widespread adoption may be difficult to pull off. It will take persistent pressure, and buy-in and vocal support from management. Also, planning and setting up tools for your concurrent reviews will take more effort than simply typing an email, so there’s a lot of staging work that has to occur before the big rollout.

This may sound Machiavellian, but to help jump start adoption, consider using a high-pressure, short-turn-

around project that must be reviewed quickly to introduce your process. (Then don't provide an alternative.) If you use this approach, make sure you've tested the process enough that you are confident it'll go smoothly, and you must be on hand to answer immediate questions from the review team. It's the ol' "throw-the-kid-into-the-creek-to-teach-swimming" form of teaching, but you'll be there to lifeguard.

Using Concurrent Review Tools

What You'll Need

You'll need three things for concurrent editing:

- Network connectivity
- A central storage location for the file
- A tool for sharing (and annotating) the file

Google Docs

This is the easiest way to get started, and many teams don't need any more than this. The network connectivity is the Internet, central storage is the Google Docs repository, and the tool is Google Docs itself. Each reviewer needs a Google Docs login to make changes. Primary advantages: Ease of use, no software install required.

To start a Google Docs review:

1. Create the document in Google Docs (docs.google.com), or import it (from MS Word, for example).
2. In the document view, click the Share button in the upper-right corner. The Share with others dialog box opens.
3. Enter the email addresses of each reviewer and select the level of editing rights they have ("Can comment" allows reviewers to suggest, but not change the source).
4. If you want to change the edit rights of reviewers after you send the invitation, you can always click Share again and change each individual's settings.
5. In the upper-right corner below the Share button, reviewers will see the editing mode button read "Suggesting."
6. As the document owner, you will see each reviewer's comment as "redlining" of various colors. General comments will also be visible. You can

click on changes and accept or reject them. You (and other reviewers) can also add replies to the markups, with notifications arriving in the user's Gmail In Box.

Annotate.co

Annotate is a web-based annotation tool. Like Google Docs, its network is the Internet, the website takes care of storage and access, and of course the website is the tool. The current version is found at annotate.co, but the earlier iteration, a.nnotate.com, is still up and running. The free account allows a limited storage space and user count, but premium subscriptions allow more leeway. Advantages: Ease of use, no software install, graphics notation support.

To start an Annotate review:

1. Upload a document. Supported formats include PDF, plain text, MS Word, and various image file formats.
2. In your workspace, double-click the file you uploaded. It opens full screen.
3. In the upper-right corner, click **Share**, then select the users (who you've previously set up in your account). Users are sent an email containing the link for the review. Or, they can log in and access the document list.
4. When users click on text (it automatically highlights whole words) or graphics, a note box opens. They can enter their feedback, change the color, position, and format of their comment box, and even reply to others' markups. Notifications are sent to commenters when a reply is added to their comment.

Acrobat

Adobe Acrobat provides several methods for reviewing PDFs. Based on comment and drawing tools, it allows easy combination of markups by multiple reviewers. The network connectivity is your own company's network or, in the case of offline file storage, the Internet. The storage location can be a shared network drive or cloud storage. The tool is Acrobat, but this summary specifically covers "Send for Shared Commenting" and "Send for Comments by Email" using Acrobat DS. Primary advantages: Rich markup tools, ubiquitous software including the free Adobe Reader.

To Send for Shared Commenting:

1. Store the PDF on a shared network location, then open the document in Acrobat DS. (Previous Acrobat versions support shared commenting, but the procedure is different.)
2. Under Review and Approve tools, click **Send for Comments**, then click **Next**.
3. Specify the storage location (reviewers must have login access to the location), then follow the on-screen prompts to add usernames to the review list. If you use MS Outlook, you may be able to select reviewers from the Outlook address book. Invitation emails containing a link to the document are sent to each reviewer. As long as network connectivity is maintained, each user will see all reviewers' changes in their copy of the PDF.
4. As the administrator, you can track multiple documents being reviewed using the Track Comments button on the toolbar within the document.

To send for email review: (This method is not dependent on a shared network drive, but the collaboration is diminished)

1. Open any PDF on your computer (or the network) in Acrobat DS.
2. Under Review and Approve tools, click **Send for Comments by Email**, then click **Next**.
3. Enter the email addresses of each reviewer, or select the addresses using the **Address Book** button, then click **Next**. A preview of the invitation message appears.
4. Edit the invitation message, then click **Send Invitation**. Users receive an email with the document attached, but linked with the original file stored on your computer, along with instructions on how to review.
5. Users open the attachment straight from the email, as-is (don't download it to the hard drive first), add their comments, then click **Send Comments**. The comments are sent to the initiator (you).
6. As the initiator, when you receive emailed comments and open the attached file, the Merge Comments dialog box appears. Select **Yes** to merge the reviewer comments into your master copy. Repeat this for each set of comments you receive.

Tracking Tools

Project tracking tools can help with accountability for your reviews and keep the review process moving forward. They also provide visibility about who is participating and who has completed their initial review. Too many exist for any comprehensive listing, but on the Internet search for terms such as "Issue tracking," "bug tracking," "Help ticket," "help desk," and "project management." As with other collaborative tools, adoption of a tracking tool may be an uphill battle, but one that is winnable.

Other Tools

Many other collaborative review tools exist, and information on these and more, including links, will continue to be added to the online version of this article stored at www.sawyerhome.net/stc2016. For visual walk-throughs of these processes, join me at this presentation at the STC Summit 2016.

Author Contact Information

Mike Sawyer
Documentation Manager
Control4
92 West 1565 North
Orem, UT 84057 U.S.A.
+1 385.209.7164
<https://www.linkedin.com/in/mikesawyerprofile>
@Akambe

Author Biography

Mike Sawyer has worked as a technical writer and manager for over 20 years. He'd list all of his accomplishments, but he's so humble that he won't. He loves technology and language, and sees technical communication as a "translation" between the technical and user worlds. He has a BA in Communications from Brigham Young University and completed coursework for his masters in Human Factors Engineering from the University of South Dakota. He lives in Utah, enjoys sleeping in, and has five kids and too many cats.



FAQS SHOULD SUPPORT DOCUMENTATION, NOT REPLACE IT

Stephanie Saylor

Many organizations use Frequently Asked Questions (FAQS) as a replacement for thorough documentation. This paper emphasizes the pitfalls of relying on a list of questions rather than delivering solid help resources. Then, it explains that FAQS—when done well—can serve as useful additions to stellar documentation. Finally, it shares best practices for organizing FAQ questions and capitalizing on their metrics to understand users' needs.

Bandaging a poorly designed or under explained website with a Frequently Asked Question section (FAQ) has become an acceptable replacement for fixing underlying problems. This paper addresses ways to truly improve those websites (or products) and deliver solid documentation that is accentuated by useful FAQS—not replaced by them.

Why

With many budgets for outreach and communications shrinking, the burden of explaining or marketing websites and products falls to overworked or inadequately qualified staff. Many of those people realize the importance of communicating with users but lack the time or experience to convey information in a way that users can easily digest. Enter the FAQ. Frequently Asked Questions can provide clarity on a number of confusing topics by a means that many authors assume will be easy to understand and easy to write. Writing a first draft of these questions can take less than an hour. And in some cases—specifically when the alternative to a poorly constructed FAQ is providing no information at all—this effort is better than nothing. However, a long list of miscellaneous questions will confuse people and mar user experience. Authors can remedy this construction by following the guidance in this paper to create useful, thought-out FAQS and help resources that address users' real concerns and shape the messages the organization wants to share.

Challenge

No one plans to create useless help resources. Nevertheless, many technical communicators pour time, energy and money into resources that meet the needs of only one type of stakeholder—the person approving it. If this situation occurs, the technical communicator must become an advocate for the other stakeholders and champion an effort to provide information that benefits all stakeholders. To do this, the technical communicator must investigate the wants and needs of all stakeholders and use that data to develop a comprehensive list of information that the organization will address in their help resources and FAQS.

Investing research time before you create or update help resources will strengthen those resources and identify weaknesses in the overall website or product. This investment can be a tough sell to the manager who must accept a delay in deliverables. Explain that it is a worthy investment that will save future redesign costs as well as improve the overall product.

The first step: Identify the needs of each stakeholder. Many people stop there. Instead, they should use this list as the basis to determine which planned FAQS need to be included and which planned FAQS are symptoms of problems with their website, product or existing help resources. Consider meeting with other technical communicators and decision-makers to discuss the list. The next section talks about identifying this information and developing strategies to filter it.

Method

This section outlines a process to create useful FAQs and improve existing help resources. Specifically, it explains stakeholder identification and questions why confusion exists on a website or within a product. Then, it reviews how to craft questions for each type of stakeholder and filter them into usable categories.

Stakeholder and User Awareness

Understanding who needs your help resources builds the foundation for developing useful resources. Early in my technical writing career, I concentrated on only new users' needs. I was wrong. Focusing on only new users ignored other stakeholders and users with different needs. To create comprehensive help resources, first identify all the types of people who may use them.

Stakeholders hold a vested interest in your website or product. They range from investors and employees to reporters and users with every level of experience. To begin identifying stakeholders, start a list. You can probably name your main types of stakeholders without too much thought. While this initial list usually includes the most outspoken and often the most important stakeholders, take caution to include underrepresented and quiet stakeholders too. If you have a usability person, they should have already gathered this information and developed personas around it. If you need to gather it yourself, ask people within your organization 1) how they use your website or product and 2) who from outside the organization contacts them. You may find that a company's own employees use their website to share directions and locate forms or that reporters use it to expand on company-issued press releases.

Next, find out who is asking questions. If your website includes any sort of "contact us" feature, analyze every message it receives. Once a co-worker handed me a three-ring binder of questions people had submitted to our "Contact Us" form. I used that binder to figure out who was asking questions and what they were having problems locating on our site. If your website or product collects analytics that measure where people are clicking and how long they are staying on each page, add that data into your evaluation. If the "When is the grant deadline?" FAQ gets the most hits, you should probably make the grant deadlines easier to find. As we will discuss in the next section, this information becomes the basis for improving the website and developing useful FAQs.

Identifying Potential FAQs

In the last section we discussed identifying stakeholders and gathering information about what they are looking for on your website or in your product. This section explains what to do with that data.

Categorize: Use, Problem, Symptom

At this point, you have amassed a lot of information. Start identifying what types of categories each piece falls into-- you should know types of stakeholders, types of uses and types of problems. Again, borrow a trade technique from the usability folks and card sort to find overlap between the groups.

Reserve a conference room with a lot of walls and whiteboards. Then write each type of use on a green sticky note and each type of problem on a pink sticky note. List stakeholders on the board, mostly as a reference to remind yourself (and anyone helping) to think about the different ways people use your site. Then, group the types of uses and problems into common themes. For example, group responses "I use the website to learn about new grants," with "Where can I find info about grant deadlines?" If you have time, do the activity by yourself first. After taking a picture of your groupings, invite input from your team— either ask them if they would move any of your groupings or ask them to make their own groupings.

You might not have much input into team formation. If you do, consider all the people you interviewed to gather your data. Include internal staff members that interact with different types of stakeholders and users. If all your representatives come from the communications department, you will have a very one-dimensional perspective. As you know, asking for the time and help of coworkers from other parts of your organization takes away from their primary work. Time box your meetings. Consider asking for 1.5 hours midday or first thing in the morning (and provide pizza or donuts). Remind them (and whoever gives the OK for you to use their time and buy pizza) that your work to improve the usefulness of the help resources will save time for these folks in the long run.

By the end of this exercise you must understand the uses of your site or product and the problems your users have trying to complete those uses. The next section discusses how to use that information to improve your help resources.

Review Existing Help Resources

The last section discussed grouping and categorizing the uses and problems your stakeholders have with your site or product. This section explains how to apply your findings to the existing site and help resources. For a website, print each page of the site and write the use or problem that relates to that page on the back of each printed page (some of the uses and problems relate to more than one page). For a software product, take a screenshot of the part of the product that relates to each use or problem and the existing help resource(s) that correspond to it. Highlight the sections that relate to the use or problem identified on each sticky note from the previous exercise. This tedious exercise does three things:

- Revision plan—It forms a tangible plan for exactly what you need to revise. Use this to estimate your time commitment and plan the revisions. If you use a project tracking tool, this is where you transition from the “figuring out what to fix” tasks to the “planning your fixes” tasks.
- Emphasizes problems with the product—It draws attention to problems in the website or product. Often, FAQs and complaints about help resources are symptoms of problems with the actual website or product. Sometimes your organization knows about them and sometimes this kind of exercise unearths surprising issues. Consider sharing your findings with your usability team and decision-makers. Keep in mind, this could be a touchy discussion if the features your stakeholders are complaining about are ones designed or implemented by those same usability folks or decision-makers. Use tactful discretion to share your research.
- Identifies useful FAQs—Finally, the FAQ! By this point, the real FAQs should stand out so much they will almost write themselves, as we will discuss in the next section.

Writing Useful FAQs

Using the research you did to identify your stakeholders and users and their uses for your site or product, you will have a strong understanding about the types of information people expect and a plan to fix your website and help resources. Use the FAQs to guide users to those updated resources. Here are some recommendations:

- Most importantly, do not write one big list of questions and answers. Separate them into grouped, digestible chunks. You can do this with

sections on one page or separated into smaller pages or tabs. Consider your organization’s ability to gather analytics when you make this decision. You want to organize your FAQs in a way that allows you to track their use so that you can continue to evaluate the needs of your users.

- Write your FAQ for the reader, not the writer. Follow basic tech writing guidelines: plain language, active voice, avoid jargon and advocate for the reader. Sometimes technical communicators work with a website or product for so long they forget things that confuse new users. Think about different types of users and how they will search for answers in your help resources.
- If all the questions start with “How do I...” do not write them in question form. It wastes space and slows the reader. For example, change “When is the grant deadline” to “Grant deadlines.” If you do use the question format, consider only bolding the most important word or phrase, “When is the grant deadline?”
- Most stakeholders are reading your resources quickly while they try to troubleshoot or meet a deadline. Do not camouflage your answers with unnecessary words or giant blocks of text. Consider what type of information you want to share. Will it be easier to digest as an infographic, fact sheet, or tutorial video? Provide the information in the medium that will benefit your users the most and be easiest for them to understand and access. If bandwidth is an issue for most of them, videos are not a useful help resource.
- If the answer is longer than one concise word or sentence, shorten it to one concise sentence that drives traffic to a help resource that explains the topic in detail. If the question can be answered concisely, using this method creates a risk of redundant information on your website. People have mixed opinions about this risk. I favor repeating some information if it saves a user an unnecessary click. For example, if all grants are due April 15, say that in an FAQ. If your site lists 10 grants with different deadlines, create a “Grant deadlines” FAQ that links to an easy to read table listing all 10. If you do answer your question in the FAQ, note that this adds the responsibility of updating content in more than one place and it does not drive traffic to additional information that you may want to highlight. If you decide to repeat information in more than one place, make sure you do not deliver contradictory answers or create situations that cause inconsistencies and confusion.

The next section focuses on marketing FAQs which serve a different purpose than help resources.

Marketing FAQs

The suggestions from the last section can apply to all FAQs—marketing FAQs included. That said, marketing FAQs require further explanation because they serve a very different purpose than “help” FAQs. While, marketing FAQs can help readers understand your product or website, they can also be completely useless if the reader identifies them as marketing materials. Most users can tell when someone is trying to market to them. Creating a “how do we make such an amazing product,” FAQ does not help your organization or your users. Revise that FAQ to drive traffic to a “reasons to choose us” or “why we’re better than our competitors” section. Give people the facts that help them make their decision (or justify their decision to the purchase approver). You have to make those facts easy to understand, evaluate and digest. Write marketing FAQs that are clear, concise and relatable. I call it the infomercial test. If my copy sounds like something my 5-year-old repeats from a toy commercial, I revise it. Even if the potential customer does not buy your product this time, you will build trust and credibility that may pay off later.

Conclusion

After you draft your revised FAQs, separate them into sections and make them searchable. Then, test your work. Go back to that first list of FAQs—the one you created when you started this exercise. Make sure that every question on that first list of FAQs:

- Concisely answers a confusing topic in a way that drives traffic to your help resources
- Is grouped into digestible chunks of questions that relate to similar topics
- Or is not applicable because you have fixed the problem the FAQ was working around

Finally, ask some different types of stakeholders and users to test your revised resources and FAQs. Do not be disheartened if they head straight for the search box or FAQs, we have trained people to troubleshoot this way. Your intense evaluation and overhaul of your help resources will pay off. Your succinct, organized, well-written FAQs will shepherd them to improved help resources that answers their questions clearly and concisely instead of frustrating them with a long list of verbose author-centric questions.

Resources

- Alred, Gerald J., Brusaw, Charles T., Oliu, Walter E. Handbook of technical writing (Boston, MA: Bedford/St. Martin's), 2009.
- Redish, Janice. Letting go of the words (Waltham, MA: Morgan Kaufmann), 2007.

References

- Czerniawski, Stefan. “Easily Answered is not the same as Frequently Asked.” Public Strategist (21 May 2008). <http://publicstrategist.com/2008/05/easily-answered-is-not-the-same-as-frequently-asked/>
- E-WRITE for the Energy Information Administration. Frequently Asked Questions Style Guide. (2006) <http://www.ewriteonline.com/faq-style-guide/>
- Kelly, Kevin. “NAQ – Never Asked Questions.” Kevin Kelly blog Conceptual Trends Current Topics. <http://kk.org/ct2/naq-never-asked-questions/>
- O’Flahavan, Leslie. “FAQs Upgrade: How the Library of Congress Regrouped to Improve Self-Service Content.” E-WRITE (28 September 2010). <http://www.ewriteonline.com/library-of-congress-faqs/>
- O’Flahavan, Leslie. “If your web content is good, you don’t need FAQs.” E-WRITE (9 June 2010). <http://www.ewriteonline.com/if-your-web-content-is-good-you-dont-need-faqs/>
- Richards, Sarah. “FAQs: why we don’t have them.” GOV. UK Government Digital Service, Cabinet Office (25 July 2013). <https://gds.blog.gov.uk/2013/07/25/faqs-why-we-dont-have-them/>
- Shapcott, Teale. “10 Reasons why FAQ’s [sic] are bad for Intranets & Websites.” Teale Shapcott (9 January 2014). <http://www.tealeshapcott.com/10-reasons-faqs-bad-intranets-websites/>

Author Contact Information

Stephanie Saylor
Technical Writer
Next Century Corporation
2701 Technology Drive
Annapolis Junction, MD 20701
stephanie.saylor@nextcentury.com
[@plainlang](#)

Author Biography

Stephanie started her career as a technology and transportation reporter. Her work has been nationally syndicated

by the American City Business Journals. Since 2007, she has worked as a technical writer for the U.S. Department of Health and Human Services and the Department of Defense. She earned a Master's Degree in digital communication from Johns Hopkins University.



YOUR SECRET WEAPON: THE DOCUMENTATION HANDBOOK

Kate Schneider

Whether you work on a team or alone, it's a good idea to organize your documentation procedures and style guides so you can refer to them later or use them for training. The Documentation Handbook is a one-stop shop for this information. It is written using your current authoring tool, so new writers learn procedures while editing topics. It is also a useful reference for the entire team.

It is never easy starting a new technical writing job, but it's even worse when you sit down and realize that your predecessor left little or no instruction for you, especially if you are new to the industry or the company. You might have teammates you can use as a resource, but they might not have time to train you one-on-one. You might be stuck weeding through different training manuals and resources, often filled with conflicting or outdated information. Worse, you might even start working on a Help project before you understand the company's voice or writing procedures, only to have to correct your mistakes later.

Whether you work alone or on a team, one thing is clear: incomplete training information and strained resources can be a serious problem when training new writers. This is where the Documentation Handbook comes in.

The Documentation Handbook is an ideal tool for lone writers and teams that want to use a central repository for their onboarding information and writing procedures. This information is kept in your authoring tool so while new writers are reviewing information, they can update existing topics and document additional procedures. This cycle of review and addition ensures that the Handbook stays current and is filled with the information your team needs to perform its daily tasks. Both lone writers and teams can use the Handbook not only for training, but also as a handy reference for workflows and tasks. Teams especially benefit because the Doc Handbook helps establish a consistent voice and workflow among the writers on the team while also decreasing the time needed for one-on-one training. The Documentation Handbook can help you combat several problems that arise as employees

are hired and leave, including knowledge loss, inadequate training time, and redundant training materials.

The Documentation Handbook: What it Is...

So, what is the Documentation Handbook exactly? Simply put, the Documentation Handbook explains how your team writes its documentation. However, it is more than simply an instruction manual; it serves several purposes.

- **Training Tool** The biggest benefit of the Documentation Handbook is its value as a training tool. By putting all of your training materials and procedures in one place, new writers can spend most of their training time reviewing and adding to the content in the Handbook. By doing this, the Handbook helps new writers learn your team's workflow and voice, as well as the ins and outs of your authoring tool. The Handbook also facilitates training by providing content to writers in a self-guided format. This enables the writer to work at their own pace and revisit information as needed. It also allows you to save one-on-one training time for more in-depth topics.
1. If you are a lone writer, the Handbook can be helpful for training should you ever expand your team. However, it is most useful if you leave the company so your successor can pick up where you left off; your knowledge does not leave with you. The same idea applies to contractors and freelancers: you have something to hand off to the company that hired you, and they can use it

Writing and Communication

to maintain the documentation going forward by referencing the procedures you put in place.

- **Reference Tool** Although the Handbook's primary purpose is to teach new writers your team's procedures, even the most seasoned writer on the team forgets things now and then. The Documentation Handbook is a great resource for all team members, especially for infrequently used procedures.
- 2. If you are a lone writer, the Handbook keeps your documentation information in a central location so you can refer to it as necessary.
- **Sandbox** From time to time, you might want to test out something new—like a stylesheet or homepage layout—before adding it to your main Help content. The Documentation Handbook is a great place to test these things because you have content to experiment on and access to your teammates for feedback.

...and What it Isn't

A Documentation Handbook can be a lot of things, and it can be different things to different documentation teams. But there are some things that it shouldn't be.

- **Your Help Project** Although you might mention some information that is found in your main Help project, your Documentation Handbook is merely a subset of the information that you'll find there. Keep the information separate so new writers don't get confused.
- **A Copy of a Published Style Guide** If you use a published style guide, like the Chicago Manual of Style, it's better to refer to or link to it than to rewrite the entire thing in the Documentation Handbook.
- **Authoring Tool Help** You don't need to add information in the Handbook explaining basic authoring tool concepts (like how to create topics). This information already exists in the authoring tool's Help.

Creating the Documentation Handbook

Before you start creating content in your Documentation Handbook, you want to set up your authoring tool so the Handbook reflects your main Help project. Be sure to use the same authoring tool for the Handbook as you do for your Help content. You may also want to set up your authoring tool to import shared resources, such as

Welcome

Welcome to the MadCap Software documentation team. This Documentation Bible is to be used by new and existing team members. It details our primary functions and shows new team members how to perform the various documentation tasks.

▼ **GETTING STARTED**

These topics introduce you to MadCap Software. They detail the essential knowledge, processes, tools, and templates used by the team to create and update our product documentation:

- » **Getting Started** Advises new team members how to get off to a good start. See [Getting Started](#).
- » **Style and Usage Guides** Outlines the writing styles and usage guidelines for the team. See [Style and Usage Guides](#).
- » **Software Tools** Summarizes our software tools and provides setup instructions and helpful tips for our programs. See [Software Tools](#).
- » **Templates** Gives you insight into the templates that we use. See [Department Templates](#).
- » **Step-by-Step Documentation Process** Summarizes the basic steps to follow when creating or updating documentation. See [How to Write Product Documentation](#).

▼ **WHERE DO I BEGIN?**

As a new team member, there is much to learn. That's why we've developed these suggestions to help you make a smooth transition into your new role:

- **REVIEW THE TEAM BASICS**
- **LEARN ABOUT THE COMPANY AND TEAM STRUCTURE**
- **REVIEW THE PRODUCT DOCUMENTATION**
- **GET THE LATEST VERSION OF THE DOCUMENTATION BIBLE PROJECT**
- **GET THE LATEST VERSION OF THE SHARED PROJECT**

▼ **IMPORTANT THINGS TO REMEMBER**

While there are many important concepts and processes for you to learn, there are a few things that are important to keep in mind as you get started on the team. First, it is not expected that you will be an expert at how everything is done right away. Second, if you have suggestions, you are always welcome to discuss them with the team.

Other things to note are as follows:

- **WE ALWAYS USE COMPANY PRODUCTS**
- **WE ALWAYS PRACTICE HELP AUTHORING IN THE DOCUMENTATION BIBLE**
- **WE DO NOT MODIFY CONTENT IN THE SHARED PROJECT UNTIL ASSIGNED TO DO SO**

Figure 1. The Welcome topic includes the most important information a writer should know on their first day.

stylesheets and layouts, to the Handbook so the projects have a similar look and feel. These things will help writers learn their way around the authoring tool and your documentation. You may also want to bind your Documentation Handbook to source control so all of your writers have access to the most recent version of the Handbook. This allows your entire team to view and edit the Handbook as necessary.

Once you have your authoring tool set up with your basic files and resources, you can focus on the content you want to include in the Documentation Handbook. Think about your team’s most important information and documentation procedures. These are the types of things you should document right out of the gate. You should include your basic documentation workflow, departmental and company style guides, lists of company resources, and anything else that is helpful to teach new writers how to do their jobs.

While there are many kinds of information that you can include in the Documentation Handbook, you might want to focus on the following three kinds of topics: Welcome, How-To, and Style Guides.

- **Welcome Topic** This topic serves as the gateway to the Documentation Handbook and points new writers in the right direction during their first few days on the job. Use this topic to compile basic information about your team’s most essential job tasks, as well as reminders or guidelines and links to topics that writers absolutely must read first.
- **How-To Topics**—These types of topics will probably make up the bulk of your Handbook. They focus on how your team writes its documentation. Include information about your documentation workflow, review procedures, and other processes. You should also add information about how your team uses specific authoring tool features (e.g., when to create reusable content, how to structure tables of contents), third party tools, and your company’s software or products (if applicable). You may also consider adding websites or social media accounts that your documentation team accesses or should subscribe to.
- **Style Guide**—Your Documentation Handbook’s style guide should focus on internal or department style information, such as unique capitaliza-

How Do We Use Notations?

Our team uses a variety of notations throughout our documentation and every notation type has different usage rules. See the table below to learn more:


Notation Type	Applied Via	Learn More
Emphasis	CSS	See Emphasis . EXAMPLE » This is an example of the span class named "span.SetOffFirstLevel" It is used to call attention to terms or important ideas in first-level bullet points. See Emphasis . » This is an example of the span class named "span.SetOffSecondLevel" It is used to call attention to terms or important ideas in second-level bullet points. See Emphasis .
Examples	CSS	See Examples . EXAMPLE This is example text.
How Did You Do That?	CSS	See How Did You Do That?  This is a how did you do that box. See How Did You Do That?

Figure 2. How-To topics give procedural information about documentation procedures, or show how to use specific functions of your authoring tool or software.

tion, product names, hex or RGB colors used in the documentation, or deviations from your main style guide. You can also use the style section to note common errors or things you're always looking up. If your authoring tool supports linking to external documents, you can link directly to company style guides on your network, so they will always remain updated.

Maintaining the Documentation Handbook

The bulk of the maintenance on your Documentation Handbook will occur when you are training new writers and they are reviewing and revising topics (which we will cover in the next section). However, there will be times that you need to add something outside of these training periods. In this case, maintenance should follow a simple principle: keep it current. Add information as necessary and do your best to update items if you know something has changed. This will make it easier—and less confusing—for new writers who later use the Handbook for training. It will also help you and your team if you need to look something up. Keeping the Documentation Handbook current is especially important for lone writers, since you never know when you may suddenly need to expand your team.

Since it is not always realistic to add or update a topic every time something changes, it's a good idea to keep a running To Do list in the Documentation Handbook. This list is not only a reminder for you, but is also a great jumping-off point for new writers when it is time to add new content to the Handbook.

Training with the Documentation Handbook

New writers will spend quite a bit of time working in the Documentation Handbook during their first few weeks on the job, becoming acclimated with your procedures and workflows. During this time, training with the Documentation Handbook falls into four major stages: Reading, Updating, Writing, and Reviewing.

- **Reading** This is where new writers should start. Reading through the entire Documentation Handbook helps writers to learn your procedures before they start to make changes. Have writers ask questions of you and their teammates, and use your authoring tool's annotation features to mark topics that might be good candidates for editing.
- **Updating**—Updating topics helps writers learn common voice and practice your documentation workflow. Writers should focus on helpful changes, such as updating out-of-date topics, adding useful notes or screenshots, revising poorly written topics, and fixing obvious typos or errors. Have the writer interview other team members to see where information is missing (or have them consult the To Do topic) and update where necessary. It might be helpful to enable track changes during this stage so you can make more effective comments when reviewing these topics.
- **Writing**—Using the techniques and workflows they've learned while reading and updating topics, writers can now start adding new topics to the Handbook. Again, have them consult the To Do list for more in-depth items that need to be added, or simply give them tasks. Writers can also make suggestions based on their own experiences; they

The screenshot shows a documentation page with the following content:

- Punctuation**
- Select a drop-down below to learn more about punctuation rules.
- NOTE
- ▼ BRACKETS
- Brackets have a special use in MadCap Software documentation. Use brackets to enclose placeholder text when documenting a value that the user must enter.
- EXAMPLE**
- Here is an example from the Shared project that shows how to apply brackets to enclose placeholder text.
- `File name should be Substitutions, you can use the appropriate number of the open and the close curly braces "()" to indicate the number of values (1-99) that can be used.`
- Notice that the action in bold can apply to any file. Instead of listing all of the possible files, the placeholder text "Name of Style" has been enclosed in brackets to represent the variable style names.

Figure 3. Rather than re-stating a published style guide's conventions for a style, style guide topics in the Handbook explain how your team uses specific styles.

might have performed tasks in previous jobs that aren't documented in the Handbook, but should be.

- **Reviewing**—Throughout the training period, you should spend time with the new writer to review their new or updated content. Use whatever method makes the most sense to you, but it should align with your typical review process so the writer becomes familiar with your review workflow. When reviewing content, check for the accuracy of the content, but also remember to check for obvious errors in voice and in workflow, so you can correct these problems. You may also want to have a teammate perform peer reviews or periodic status checks before you meet with the writer to free up your schedule and encourage team collaboration. And of course, if you uncover a style error or a question that has not yet been documented in the Documentation Handbook, be sure to add it as a new style guide note or process.

Best Practices

Keep the following best practices in mind when working with your Documentation Handbook.

- **DO** have new writers work exclusively in the Handbook until they have a good grasp on procedures and voice.
- **DO** make your full Help project available, so writers can refer to it while they add new topics to the Handbook. This also lets them read through the Help project and familiarize themselves with it.
- **DO** encourage collaboration between new and existing team members. Existing team members should be a source of information for new members, and can also do peer reviews.
- **DO** let new writers ask questions and make suggestions. There might be a better way to do something that only an outside eye can see.
- **DON'T** neglect the Handbook, especially if you are a lone writer! Schedule some time to work on it if you need to. When then the entire thing needs to be updated, everyone gets overwhelmed.
- **DON'T** try to document everything when you first start building your Handbook. Stick to the basics and add as you go.

Author Contact Information

Kate Schneider
Senior Technical Writer
MadCap Software
7777 Fay Avenue, Suite 201
La Jolla, CA 92037
+1 858.875.4228

Author Biography

Kate Schneider is a senior technical writer at MadCap Software in San Diego, CA. She works closely with other team members to write and maintain the Help documentation for MadCap's software products, including Flare, Lingo, Capture, and Mimic. She also has been responsible for training new team members using the team's documentation handbook.

Prior to moving to San Diego, Kate worked as a technical writer and proposal writer for several startup companies in the Midwest. She has spent nine years in the computer software industry, with experience in building online Help systems from the ground up, single-sourcing and restructuring existing documentation suites, and creating video tutorials.

Kate holds a Master of Science in technical communications from Northeastern University in Boston, MA, and a Bachelor of Science in business administration and public relations from Coe College in Cedar Rapids, IA. She enjoys playing video games, exploring craft breweries with her husband, and snuggling with her two West Highland White Terriers.

