Engaging your readers in the documentation

How and why with social media

Tekom tcworld
23rd – 25th October 2012, Wiesbaden
But nobody reads the documentation!

Speaker’s notes: But nobody reads the documentation!

Blogs, websites, forums... There is so much information out there! How can we ensure people find our documentation?

“Nobody reads the documentation.” We have all heard people say that.

But we also know that our documentation is the best source of information available to our customers. We spend our lives making sure of that.

All we need to do, is to make sure that people know it too. The first step is to make sure that they can find the documentation.

One way of doing that is by clever use of social media.
Why social?

The social enterprise

Social is happening anyway

How to get our readers involved

Social media

Documentation as an emotional experience

Speaker’s notes: Why social?

We know that companies are becoming more social. Studies show that engaged customers buy more, are more satisfied with the product, and are more likely to help each other use the product. Similarly, I’m proposing that engaged readers read more, are more satisfied with the documentation, and will use the documentation as a tool to help each other.

Social is happening anyway.
People are talking about us and our products. They may be saying good or bad things. Let’s join them, find out what they’re saying, and answer them when we can. That way, we can make sure that we benefit from the social phenomenon rather than only lose from it. Where possible, we can influence what people think about us, by responding in a timely and appropriate way.

Documentation is the face of the company. It’s where people come when they’re looking for answers. It’s where they land when they search for information.

How can we make our documentation a living, interactive hub where people can come to find answers, talk to us, swap techniques and help each other?

I’m going to:
• tell you about engaging your readers in the documentation
• show you some of the tools you can use to do that.

We’re also going to get a bit emotional about documentation...
"Documentation as an emotional experience“

But wait...

Speaker’s notes: Documentation as an emotional experience

"Documentation as an emotional experience“: Isn't that a bit weird? OK, perhaps I’m saying it with tongue in cheek. But guess what... reading some documentation already is an emotional experience!

Instead of frustration, let’s aim for these emotions:
• Laughter
• Pleasure
• Satisfaction
• Fulfilment and even reward – “Yay, I did something right.”
"Documentation as an emotional experience"

But wait...

I felt the earth move when I read your docs, mate
Speaker’s notes: The death of SEO

A very interesting post appeared on Forbes.com in July 2012: “The Death Of SEO: The Rise of Social, PR, And Real Content”

SEO means “Search Engine Optimisation”. It’s the art of designing your content so that it ranks highly in search engine results, and people can therefore find your content. The term SEO is usually used to apply to external search engines, like Google or Bing. You can also apply it to internal search engines, such as the search provided by your particular wiki or content management system.

Technical communicators know a number of SEO techniques. We know we shoulds:
• Make the title meaningful.
• Add key words to the content, especially in the headings.
• Make sure the first paragraph on the page is meaningful and includes the relevant key words.
• Write succinct, clear, and relevant content.

The gist of the Forbes post is this:
1) Google is changing the algorithm used to rank pages in search
results.

2) Your “internal” SEO is becoming more important. That’s the way you write and organise your content. Primarily, it means good use and judicious placing of key words in the content, headings and title.

3) What about “external” SEO? Links are like votes. The more links that point into your content from external sites, the better. That has always been so. But now, the way Google evaluates “external” SEO is changing. Google is becoming smarter at recognising artificial links that people create all over the web, pointing into their content purely to improve their own SEO.

4) Most importantly, Google is giving a higher ranking to links coming in from social media. That includes the “shares” that people post on Twitter, Facebook, Reddit, and so on. If someone “likes” your content on Facebook, or adds a +1 on Google+, that counts too.

In other words, Google recognises content as valuable and relevant if people share it via social media.

How does this affect us as technical communicators? Just two points:
• Write great content. (Well, we know how to do that already.)
• Make sure people find your content and share it via social media, so that other people find it and share it via social media, so that other people... That’s called “going viral” and it’s a good thing!
Speaker’s notes: Tools

Let’s look at some things we’ve done with social media in and around our documentation.

I’ll show you some real-life examples and give you some tools you can take away and use.

Note: By “tools”, I mean techniques that you can use. I’m using a wiki called Confluence. But the techniques I’m showing here can be applied to most web-based platforms.

We’ll start with low-level engagement:
• Comments on documentation pages.
• Feedback via Wufoo forms.
• “Tips of the Trade” pages that link to blog posts and articles written by readers, customers and community developers.

Then greater involvement:
• Open editing of documentation pages by subject matter experts and even customers.
• Doc sprints - Organised events that get people together in the real world to write documentation.
Integration of external social tools in and around the documentation.

An example of a game wrapped around the documentation, plus interactivity via Twitter and forums: The “Dragon Slayer” documentation.

Hold on to your hats, because this presentation is packed full of information.
What is a wiki?

A wiki is a web application that runs on a server. It produces web pages that people can access in their browsers, such as Firefox, Internet Explorer, and Chrome. In essence, the wiki software puts an "edit" button on a web page. If you have the right permissions on the wiki, you can edit a page and publish your updates immediately, from your browser.

Click the "Edit" button, change the content, click "Save". That’s it.

You, as the author, and all your readers interact with the wiki pages in the same way, via your web browsers. People who have only read permissions will see the web page, but without the edit button.

There are many brands of wiki: Confluence, DokuWiki, MediaWiki, Mindtouch, etc.

I’m using Confluence, the enterprise wiki from Atlassian.

This screenshot shows the home page of the Confluence documentation. This is just one of the documentation spaces on the wiki.
Let’s jump straight into the first tool I want to show you: Comments from readers.

We allow our readers to add comments to our documentation pages.

We're acknowledging that technical writers are not perfect.
We make mistakes.
People find them.
We fix them.
Some readers give us a lot of hard-won information. Where relevant, we incorporate it into the page, and thank the person.

This is awesome, because it engages the person who made the comment, and also gives our readers extra information.
People like to let us know they’re keen on a new feature.

*Waits intently for Apache Connector 2.0 to be built on windows*

**Speaker’s notes: Comments**

People like to let us know they’re keen to have a new feature or improvement in the product.
In return, we can tell them we’re working on it, and point them to the issue in the bug tracker, so that they can stay informed of progress.
Speaker’s notes: Comments

Comments can be perhaps a bit off topic, but the readers are totally engaged.

[Demo now.]

Laughter is a good emotion to have in the documentation.

Demo:
https://confluence.atlassian.com/display/DOC/Cheese+Macro
Speaker’s notes: Types of comments received

Types of comments received over a one-week period:
• Offering information, hints and tips: 31%
• Suggesting an addition/correction to the documentation: 7%
• An unhelpful comment, possibly meant to be humorous: 3.5%
• Requesting help on functionality ("how do I..." or "is it possible to..."): 31%
• Requesting support (when something does not work as expected): 10%
• Suggesting a feature or improvement in the product: 17%

From a technical communicator’s point of view, all these comments are interesting and useful.

The ones we can apply most readily to the documentation are:
• Comments that supply information
• Additions and corrections

The requests for help are very interesting, in that they show us potential gaps in the documentation.

For the others, we direct the customers to a more appropriate channel:
• Support team
• Feature requests in the issue tracker
• User discussion forum

**Background information**

The number and type of comments received on the Atlassian documentation wiki between the 7th and 14th of January 2011.

Overall traffic across the entire wiki:
• 99,329 people visited the site, in a total of 157,312 visits.
• 45,549 pages were viewed a total of 540,488 times.
• Number of comments on all spaces: 98

Traffic in the Confluence documentation (DOC space) only:
• 1,793 pages were viewed a total of 99,310 times.
• Number of unique page views: 78,037
• Number of comments on this space: 39

Who made the comments:
• 18 comments were anonymous – added by someone who was not logged in to the wiki.
• 21 comments were from logged-in users.
Speaker’s notes: Dealing with feedback

Another new learning point for us is how to deal with customer feedback, and how to use our responses as a good way of engaging our readers. After all, we want them to come back to our documentation when they have questions, rather than finding an answer that may be incorrect or misleading, somewhere else.

In dealing directly with readers via comments on the documentation, we’ve learned a lot.
- We’re dealing with feedback every day, on the fly. So we learn not to agonise too long over a response. Tell them as much as we know, show them where to find more information, and move on.
- Be friendly, polite, and “real”.
- Each of us gains the confidence to respond.
- It’s great to see how people react mostly with thanks and further information.
- We also learn how to handle angry responses. Remain polite, and keep our own responses to the point. Always aim to help the customer.
Speaker’s notes: Wufoo forms for gathering feedback

We’ve looked at comments. Now let’s examine another way of getting feedback.
Wufoo is an online web service that allows you to design an HTML form and collect information from people over the Internet.

You can then grab the HTML for the form and embed it into your own web page.

Confluence has the Widget macro, which you can use to embed a Wufoo form into a wiki page.
The screenshot shows such a form, which we used to collect feedback about the new Atlassian Plugin SDK.

Also on the screen is the code for the macro in Confluence wiki markup:

```markdown
```
Speaker's notes: Links to readers’ blogs

We have pages in the documentation called “Tips of the Trade”. These pages contain links to blog posts written by customers, full of tips and techniques about how to use our products.

This is awesome, because it engages and rewards the bloggers and it also gives our readers information that we could not otherwise provide.

When we add a link from our documentation to a blog post, we add a comment to the post, letting the blogger know we’ve linked to their site.

It is a good thing to have the support of bloggers, since many of them are influential in your community of readers and customers.

This is also a useful addition to the documentation

- Plugs a gap we could not fill. Often, technical documentation is forced to ignore the small, specific use cases. We can’t cover them all, and maintaining them would become a nightmare. But if other people have documented them for us, we can point our readers to the external information, with suitable warnings that it may not be current
- Requires little maintenance.
- Expands our own knowledge.

Background information:

Research before adding the link:

- Must be a "how to" article containing good information well presented.
- Must be correct.
- Must be useful – an edge case but not too much of an edge case.

Communication:
• Warn readers that the links point to external blogs.
• Let the bloggers know what the requirements are.
• Inform people via blogs that we’re doing this.
• People can suggest their blog posts by dropping comments on the page.
• People can even add their own links --- see ACLA coming up next.
Speaker’s notes: Open editing of wiki pages

Our vision is to provide a high-quality, dynamic, interactive set of documentation, watched over and shaped by the technical writers.

Is it safe to allow readers to update the documentation itself?
Answer: It depends on the documentation.

I’m in:
• The software development industry.
• An agile environment.
• An “open” company.

Our authors include:
• Technical writers.
• Employees: developers, support, marketing and sales.
• Community developers.
• The general public.

The technical writers monitor and fix the updates:
• RSS feeds.
• “Watching” the pages and receiving email notifications of all updates.
**Permissions and ACLA**

Comments – anyone can comment

Product documentation

- All employees can update
- Other contributors sign a licence agreement first

**Atlassian Contributor License Agreement (ACLA)**

- Based on Apache Contributor License Agreement
- Guards the rights of all

---

**Speaker’s notes: Wiki permissions and ACLA**

We use wiki permissions to control who can edit the various types of documentation.

Anyone can comment, including anonymous users (that is, people who have not logged in to the wiki).

All staff members can update the documentation.

We ask other people to sign a contributor licence agreement before we give them update rights. It was a very interesting experience formulating this agreement. It’s based on the Apache Contributor License Agreement. It protects the rights of Atlassian, of the contributors, and of any third party whose content may be unwittingly included into the documentation.
Speaker’s notes: Any technical writers in the room?

You probably spotted the two spellings of “licence” on the previous slide. Relax! The name of the licence uses American spelling. I don’t. 😊
Speaker’s notes: Creative Commons copyright licence

We have a Creative Commons licence on all pages:
• Anyone can use our documentation, provided they acknowledge us as the source.
• Anyone who contributes to our documentation knows that their contribution falls under the same licence.
I’d like to show you some examples of how giving update rights to subject matter experts really pays off.

This example shows some product release notes. They’re the release notes for the latest version of Confluence, in September 2012.

Release notes are an interesting document. They’re part technical, high in information, and also part marketing.

Let’s take a look at the page history, to see the number of updates made and who made them.

**Demo:**
https://confluence.atlassian.com/display/DOC/Confluence+4.3+Release+Notes
Contributors include the technical writer, software engineers, product managers and marketing team.

**Demo:**
https://confluence.atlassian.com/display/DOC/Confluence+4.3+Release+Notes
Speaker’s notes: Collaboration – database setup

This example shows the MySQL setup page in the Confluence configuration guide. This page has seen many updates over the years, in response to comments from customers, and problems and solutions found by our support team, and so on.

- Updated by Matt Doar, an external partner and community developer.
- Updated by staff engineers: Niraj, Matt R, Anna, Michael.
- Updated by technical writers.

It’s much more efficient for the subject matter expert (engineer) to update this type of page directly, rather than raising a change request for the technical writer.

Demo:
https://confluence.atlassian.com/display/DOC/Database+Setup+For+MySQL
Doc sprints

Shut ‘em up in a room together
Don’t let ‘em out until the documents are written

Speaker’s notes: Doc Sprints

A very good way of engaging people in the documentation is to get them together to write tutorials and other documentation. We have held a four doc sprints to date.

A doc sprint is a short period of time when a group of people collaborate to write a specific set of documents.

Our first doc sprint focused on plugin and gadget tutorials – documentation for developers. Our second doc sprint focused on user documentation, and specifically on quick-start guides for our products. The third and fourth doc sprints focused on developer documentation again.

So, we put 30 developers in a room with computers for 3 days. We didn’t know whether to expect some classy tutorials or... the complete works of Shakespeare.
Speaker’s notes: Doc sprinters

So, what did we get?

We found that if you give people a focus, an opportunity, and something to be excited about, they jump in and do awesome stuff. And they enjoy it.

23 people took part in our first doc sprint: 19 Atlassians and 4 community developers

Our second doc sprint had around 30 sprinters: 14 from outside the company!

The third sprint was internal only. The fourth had more than 30 sprinters, again with a number of external participants.

We had people from all over the world, including far-flung places like Russia and Israel. And Australia. 😊
People worked remotely as well as in the company offices in Sydney, San Francisco and Amsterdam.

Background information:
Why do people take part in doc sprints?
• I read a study on why people are willing to contribute their time to community documentation projects: Why do people write free documentation? Ours is not exactly free documentation, but I think the same sort of motivations apply.
• People enjoy learning cool stuff from the other experts on the sprint.
• External developers enjoy the contact with your company, and employees enjoy and learn from the external developers who are using your tools.
• People like helping other people.
• People get a sense of satisfaction from fixing documentation that is out of date.
• People feel good about developing a new tutorial that is as near perfect as possible.
• Personal email invitations were very powerful. One of our external attendees mentioned this specifically, during our retrospective session.

Who did we invite?
• Developers from within the company.
• Community developers from outside the company, via targeted invitations.
• Technical writers from all over the world, via targeted invitations and tweeting and blog posts.
• Anyone who wanted to join us, via a couple of blog posts.
Speaker’s notes: Doc sprint results

The results of the first sprint:
• We produced 19 tutorials on how to develop gadgets and plugins.
• A few developers decided to spend the first day of the sprint looking over the existing reference documents and doing a mass update, before starting the tutorials.
• Some even decided that this was the most valuable exercise for the entire sprint.
• One person was heard to exclaim, “The biggest sign of victory is how many pages I managed to delete.”

The results of the second sprint:
• We produced 23 guides, some of which consist of many pages and were worked on by multiple authors.
• It was great collaborating with technical writers all over the world, as well as all sorts of people within the company.

In the third sprint, we produced a number of new developer tutorials.

For the fourth sprint, we decided to focus primarily on updating the existing documentation. We had sprinters testing and updating the existing tutorials. We also created some new tutorials, for the latest technology introduced into the product.
Feedback:
• Overwhelmingly positive feedback from the participants. (We were expecting some negative feeling about doing documentation, especially from the developers, but that didn’t happen at all.)
• Positive feedback from the company.
• Plenty of interest from the international technical writing community.
• We’ll do it again.
Speaker’s notes: Doc sprint wiki

A wiki or other social and collaborative platform is essential.

This is how we used the wiki:
- Planning – the wish list, schedule, signup and all necessary information.
- Templates.
- The tutorials themselves – the sprinters developed their documents in a private area of the wiki, and we later moved them into the official documentation after review.
- Retrospective – we published the results of the retrospective on the wiki. Remote sprinters added their feedback directly on the retrospective pages.
- Fun stuff – a haiku competition, anagrams and more.
- Our hall of fame, containing photographs of the sprint and the people. These help to engage both the people who were there and the people who may be there next time.
This is a nice segue from the doc sprint to social media. Let’s see how we used Flickr in our doc sprint hall of fame.

Flickr is an online service where you can upload photographs, videos and other images and share them with other people. The Flickr API also opens up opportunities for “mashups”, such as embedding Flickr images into other web pages.

Many CMSes offer a way to embed a Flickr photo stream onto a page.

In Confluence, you can use the Widget macro to embed a Flickr image, or a set of images, onto a wiki page. If you embed a set of images, it shows as a slide show.

Format of the Widget macro in wiki markup:
{widget:url=http://www.flickr.com/photos/31065906@N08/sets/72157623503923898/}

Demo:
• Go to the doc sprint hall of fame:
https://confluence.atlassian.com/display/DOCSPRINT/Doc+Sprint+Hall+of+Fame
• Play the Flickr slide show.

The hall of fame shows a photograph of each sprinter, as well as a slide show of photographs drawn from Flickr.

Photographs are really great:
• For the people who attended the event, photos bring back fond memories.
• For people who didn’t make it this time, the photos show how much fun it was. People can see the sprinters who took part, realise that the doc sprint is not scary, and be encouraged to sign up next time.
Speaker’s notes: Incorporating social media - videos

YouTube and other social video services offer a great way of making your release notes more attractive, informative and engaging.

YouTube offers a few ways of embedding a video in your web page. Just go to the video on the YouTube site and click “Share”.

Many CMSes offer a safe way of embedding videos from well-known sites, such as YouTube, Vimeo or Daily Motion. For example:
• WordPress has an option to “Add Media”.
• Confluence has the Widget macro.

In other CMSes and documentation development tools, you may have the ability to insert raw HTML. You can grab the HTML from YouTube, for example.
Speaker’s notes: Example of a video in a document

This example shows the release notes for a Confluence release. The marketing team creates an overview video for every major release of our products. They put the videos on YouTube. The technical writers take advantage of this by including the overview video at the top of the release notes.

The rest of the page contains technical information and detailed descriptions of the features. Some people prefer to scroll down the page and focus on the points that interest them. Other people prefer to sit back and watch a video. We cater for both types of consumer, and have a pretty page to boot.

Demo:
https://confluence.atlassian.com/display/DOC/Confluence+4.3+Release+Notes
Speaker’s notes: Links are magic

Links are shareable.
Links are the currency of the Web.
Links bring people to your documentation.

Sprinkle them around like fairydust.

Add a link from the video page on YouTube, pointing to the release notes or other page in the documentation. The same applies to photos on Flickr, presentations on SlideShare, etc. People will share the link if it’s good.
Speaker’s notes: A little bird told me

Let’s talk about Twitter, and some of the ways you can use it to draw people into your documentation.

• Twitter as a medium for release notes.
• Encouraging customers to tweet their hints and tips about your products.
• Rewarding the community for contributing their tips.
• A configuration guide that looks like a game, and uses Twitter to add interactivity.
• How you can embed tweets and Twitter streams into your documentation.
Speaker’s notes: Twitter as medium for release notes

We send out a series of tweets for each major release of our products.

Each tweet:
• Reflects a major point in the release notes.
• Links to the "real" release notes.
• Contains a hash tag to tie the tweets together.

Why use a #-tag?
• It provides a way for readers at any time to see a collection of such related tweets.
• A collection of release highlights is... the release notes. That’s how we use Twitter as a medium for our release notes. Ta da ♫

Also in the above slide:
• People have re-tweeted our tweet.
• Other people have added extra tweets using our hash tag – one in Dutch!

Tweets are small chunks of information that are:
• Easy to digest.
• Shareable.
• Interactive and conversational.
• Useful for drawing people to the documentation.

Is there a chance people will add unfavourable information to our Twitter stream?  
Oh yes! But that’s all part of it. It’s up to us to reply, and keep the conversation going.
We encourage readers to submit tips and hints via Twitter. In other words, customers tweet about a technique they have discovered when using our products. Sometimes people write a blog post about the technique, and then tweet about the post.

The hash tag is the key. We encourage people to use a special hash tag, to tie all the tweets together.

For our product called JIRA, the hash tag is #JIRATips.
For Confluence: #ConfluenceTips

Taking it a step further: We have embedded the Twitter stream into a documentation page. So not only can people see the hints and tips from everyone via Twitter, they can also see their own names (Twitter IDs) appearing in our documentation.

Demo:
http://https://confluence.atlassian.com/display/DOC/Tips+via+Twitter
https://confluence.atlassian.com/display/JIRA/Tips+via+Twitter
Speaker’s notes: Badges for Tip Tweeters

People who tweet the tips often like to let other people know that they do that.

We’ve designed some pretty, cute badges that people can add to their blogs or other social sites. Along with the badges, we supply some HTML that links back to the “Tips via Twitter” page in the documentation.

It’s a win-win opportunity, like most social tools:
• The tweeters get some recognition for their work.
• They feel part of a community, and feel an identity with the company.
• We get promotion of our “tips via Twitter” campaign, and of the products too.

Background information: How does it work?
We host the badge images on our documentation site. We provide HTML code that people can copy and paste into their blog “widgets” or other locations.
The code grabs the image and links it to our “Tips via Twitter” documentation page.
When people click the badge on someone’s blog, they come to our page.

Reference:
Encoder’s notes: Gamification - the Dragon Slayer

We have a complex and lengthy configuration guide. We wanted to make it more enjoyable for people to work through all the steps in the guide. So we added some aspects of a game.

The game:
• Many stages
• Charlie, the hero, starts out with very little clothing or other accoutrements. Actually, he’s naked, and has just a staff.
• As the reader works through the steps, Charlie acquires pieces of armour or a bigger weapon.

The rewards:
• Charlie acquires clothing and armour.
• The person feels part of a community of “dragon slayers”. They have a sense of camaraderie and contact with Atlassian.
• The person gets a difficult task done by following step-by-step, top-down instructions. They have a sense of achievement, conquering the dragon and telling everyone else how much they have managed to do.
• Of course, there’s a prize of a T-shirt when you get to the end of the guide. A real T-shirt.
It’s amazing what people will do for a T-shirt!
Speaker’s notes: The Dragon Slayer pages

Format:
• Fun stuff is at top and bottom of page.
• In between is solid, well-tested “how to” information.
Speaker’s notes: Dragon Slayer interactivity

Interactivity:
• Tweet directly from the wiki page.
• Real-time display of tweets on the page.
• Link to Twitter for people who want to follow the other dragon slayers via the Twitter web interface.
• Link to forum, where people can meet other dragon slayers and help each other. The Atlassian support team also monitors the forum and helps with problems.

Let’s see some dragon slayers in action...
Speaker’s notes: Dragon Slayer Twitter stream

People love a game.

The suggested tweets are funny, in context and a call to arms. Other people join in. Some of them hijack the Twitter hash tag to add to the fun.

When I got into the office on the first day after publishing the Dragon Slayer documentation, I saw the Twitter stream and thought, “Wow, people are actually doing it. They’re doing the dragon quest and tweeting all over the show!” It was an awesome feeling!

Demo: https://confluence.atlassian.com/display/ATLAS/Here+Be+Dragons
How to embed Twitter into your page

This page in the Dragon Slayer documentation includes 2 ways of embedding Twitter.

• A link that prepopulates a tweet.
• A dynamic, real-time list of tweets that contain a given hash tag.

Let’s look at both these techniques.

The first is to display a link which people can click if they want to tweet some suggested text. In other words, you are prepopulating a tweet for them. People can choose whether to tweet your words, or adapt the words and then tweet.

Prepopulating a tweet sounds complex, but it’s easy. Just use an HTML link with a URL parameter called “status”:
http://twitter.com/?status=My status

This page also displays a stream of tweets that contain the hash tag #AtlassianDragons.

To get a stream of tweets that use a given hash tag, you can use a Twitter search URL:
http://twitter.com/#!/search?q=My query
Example:
http://twitter.com/#search?q=%23AtlassianDragons

You need to put the search into a widget, to get a pretty display...
**Twitter link and widget in Confluence**

The link is just an HTML link.

Especially for people who love wiki markup, I’ve included the Confluence wiki markup on this page!

[tweet your status][http://twitter.com/home?status=My status]

To embed a stream of tweets, you can use the Widget macro in Confluence.

Here’s the Confluence wiki markup:

{widget:url=http://search.twitter.com/search?q=AtlassianDragons}

Other CMSes offer similar options. WordPress, for example, has a number of Twitter plugins.
Speaker’s notes: Building a Twitter widget

You can also go to the Twitter widget builder, build your own widget, then grab the HTML to insert it onto your page.

Demo:
https://twitter.com/settings/widgets/new/search
Speaker’s notes: Which emotion would you rather have?

Which emotion would you rather see in people who are reading your documentation? Frustration or exhilaration?

We have both. We’re constantly working to move from the one to the other.
Speaker’s notes: My book

I’ve recently published a book about developing technical documentation on a wiki. The book is called, “Confluence, Tech Comm, Chocolate: A wiki as platform extraordinaire for technical communication”.

A large part of the book is a step-by-step how-to guide to building a documentation set on Confluence wiki.

In addition, there are a number of tips and a number of detailed sections on more general topics that concern technical communicators. In these sections, I’ve focused on how we can make use of the unique features that a wiki brings, to solve the problems a tech comm team faces. There are also a number of ideas on how we can raise the profile of the tech comm team within and outside our organisations.

There is also more detail on the topics covered in this webinar.

These more general areas of the book include:
• Unique ideas for working in agile environments.
• Detailed guidelines to planning and running a doc sprint.
• Engaging your readers.
• Using social media.
• Crowd-sourcing documentation.
• Handling feedback on documentation.
• And more.
References 1

- Social media
  - “How to nurture an online community” by Rusty Williams on iMedia Connection: [http://www.imediaconnection.com/content/17758.imc](http://www.imediaconnection.com/content/17758.imc)

- Confluence wiki
  - Confluence documentation: [https://confluence.atlassian.com/display/DOC](https://confluence.atlassian.com/display/DOC)
References 2

- Feedback via comments and online forms
  - Wufoo Online Form Builder: http://wufoo.com/
  - Confluence documentation on using the Widget macro: http://confluence.atlassian.com/display/DOC/Widget+Macro
  - Comments can be fun: https://confluence.atlassian.com/display/DOC/Cheese+Macro
References 3

- Linking to external blog posts
  - The “Tips of the Trade” page in the JIRA documentation: http://confluence.atlassian.com/display/JIRA/Tips+of+the+Trade
  - The “Tips of the Trade” page in the Confluence documentation: http://confluence.atlassian.com/display/DOC/Tips+of+the+Trade

- Open editing of wiki pages
  - Creative Commons licence: http://creativecommons.org/licenses/by/2.5/au/
  - Atlassian Contributor License Agreement: https://confluence.atlassian.com/display/ALLDOC/ACLA+v2.0
  - Apache Contributor License Agreement: http://www.apache.org/licenses/icla.txt
References 4

- **Doc sprints**
  - The results of the doc sprints:
    [http://ffeathers.wordpress.com/2010/03/07/we-held-a-doc-sprint-and-it-was-awesome/](http://ffeathers.wordpress.com/2010/03/07/we-held-a-doc-sprint-and-it-was-awesome/)
  - How to plan a doc sprint:
  - Our doc sprint wiki:
    [http://confluence.atlassian.com/display/DOCSPRINT](http://confluence.atlassian.com/display/DOCSPRINT)
References 5

- Embedding Flickr images into a documentation page
  - Flickr: http://www.flickr.com/
  - A page showing an embedded Flickr photo set – the doc sprint “Hall of Fame”: http://confluence.atlassian.com/display/DOCSPRINT/Doc+Sprint+Hall+of+Fame
  - Confluence documentation on using the Widget macro: http://confluence.atlassian.com/display/DOC/Widget+Macro

- Embedding YouTube videos
  - A page with a video – product release notes: https://confluence.atlassian.com/display/DOC/Confluence+4.3+Release+Notes
References 6

- **Twitter**
  - Twitter: [http://twitter.com/](http://twitter.com/)
  - A guide to Twitter for technical communicators by technical communicators: [https://wikitechcomm.onconfluence.com/display/DOC/Home](https://wikitechcomm.onconfluence.com/display/DOC/Home)
  - The “Tips via Twitter” page for the JIRA documentation: [http://confluence.atlassian.com/display/JIRA/Tips+via+Twitter](http://confluence.atlassian.com/display/JIRA/Tips+via+Twitter)
  - Building a Twitter widget: [https://twitter.com/settings/widgets/new/search](https://twitter.com/settings/widgets/new/search)
References 7

- Documentation as an emotional experience
  - The Dragon Slayer documentation: http://confluence.atlassian.com/display/ATLAS/Here+Be+Dragons
  - Ellis Pratt’s post on the Cherryleaf blog: http://www.cherryleaf.com/blog/2010/02/turning-technical-documentation-into-an-emotional-experience-for-the-customer/
Contacting me

- Email: sarah@atlassian.com
- Twitter: @sarahmaddox
  http://twitter.com/sarahmaddox
- LinkedIn: http://au.linkedin.com/in/sarahmaddox
- Blog: http://ffeathers.wordpress.com
- Other blog: http://travellingworm.wordpress.com/

Let me know if you've done something similar, or something totally different

The end