



## Charlotte Brogden interviews Sarah Maddox

*Sarah, please tell us about yourself.*

I'm a technical writer with Atlassian, a software development company in Sydney. I've worked in software development environments in South Africa, the UK, the Netherlands and now in Australia. When I'm not writing you will probably find me striding through the Australian bush or gazing up at birds trying to engage them in conversation.

*How long have you been a technical writer?*

Fourteen years – five of them in my current job at Atlassian.

*How did you get into technical writing?*

I started my working life as a developer and later I became an analyst programmer. Then a couple of babies arrived and I wanted to work from home while they were small. So after a brief foray into the publishing world as book indexer, I discovered that I am actually a technical writer.

How did that happen? Well, one day my husband, who also works in systems development, said, "We need some documentation for a new system and would you like to write it?" That was ideal because I could still work from home for part of the time. The development company was called Face Technologies. They're in South Africa. They build really exciting software using biometrics and smartcard technologies. So I learnt a lot and I really loved working for them. They got me started in a great career.

*Did you attend university? How well did it prepare you?*

I did a Bachelor of Arts degree in English at University of Cape Town in South Africa. It was an excellent degree in navigating a wide range of experiences and interests. But it didn't prepare me for any career in particular and at that stage I hadn't thought of technical writing. So at the end of the degree I completed a four-month computer course and started out as a developer.

*What is your level of expertise including software packages and skills? How much was self-taught or learned on the job?*

I've used many software packages. For example, Confluence and Microsoft Word.

On the documentation management side: SharePoint and Documentum, Lotus Notes and Domino. I've used QuarkXPress and Acrobat. For online help: HTML Help, Help & Manual, and Robohelp. Also things like Visio, PaintShop Pro and Photoshop.

And a lot of other tools that are not specifically technical communication related. For version control I've used VSS, Git and Mercurial.

I've also attended short training courses in various technologies such as DITA XML and methodologies like Information Mapping. They're really useful in broadening my knowledge of what's available. But I think my in-depth skills came from learning on the job.

*Do you work on contract or full or part-time? What type of work do you prefer?*

I'm in a permanent role at the moment working full-time. I've worked part-time before and also on contract roles.

I prefer full-time rather than part-time because I think you miss out on a lot if you're not in the office at the same time as your colleagues. And as far as contracting versus permanent goes, I don't really have a strong preference. Permanent is really good because I feel a strong connection and a sense of pride in the company. But contracting has its good points too. It gives you the freedom to try a number of different roles in a short period of time.

*Were you ever unemployed for a long time?*

The only time I've been unemployed for a long time was by choice when my children were very young. So I did a lot of studying in that time to keep up with technology and broaden my skills. I learnt things as diverse as object oriented programming and book indexing.

Looking back it was good I had that chance to refocus my career. It is what got me started as a technical writer, which is an awesome career.

*What part of technical writing do you enjoy and never stop talking about?*

I think the best part is interacting with our audience, with the people who use the documentation we write and the other forms of user assistance we give them. I love hearing from customers on Twitter, or via comments on the documentation, in various forums and they even get hold of me on Facebook and LinkedIn. It's brought home to me just how much people need and value the help we give them.

I've also had the wonderful opportunity of attending technical communication conferences in various parts of the world. My very first visit to the United States was thanks to the WritersUA conference in Seattle a few years ago. I loved it and I've loved every conference since. Meeting other technical writers who know and love the job is really great.

*What is your most significant accomplishment as a technical writer?*

I think that has to be my book. I've recently published a book called *Confluence, Tech Comm, Chocolate: A wiki as platform extraordinaire for technical communication* and it's about developing technical documentation on a wiki. It's also about technical communication in agile environments, using social media and a whole lot more.

There are so many great things about writing on a wiki. I wanted to pass information on to people who haven't had the chance yet to develop and deliver content on a wiki, or to interact with documentation written on one.

I think there are three aspects of the book that I'm most proud of. I used the technique of progressive elaboration, which means you progressively reduce the level of detail in

the step-by-step instructions, while increasing the complexity of the concepts. So you start with fairly detailed instructions for the basic functionality in the first section of the guide and you become less detailed in later pages because the users will have become more familiar with what you're talking about. At the same time, they will be dealing with more complex topics. But that's okay because they're in their field of expertise.

The book has a hero, called Ganache. She's a technical writer, of course. Readers follow in her footsteps as she sets up her wiki. There's humour, fun, and lots of technical help. There's a lot of chocolate too ...

It is available on Amazon  
<http://www.amazon.com/Confluence-Tech-Comm-Chocolate-extraordinaire/dp/1937434001>

It's also available from the publisher, which is XML press  
<http://xmlpress.net/>

*What would be your ideal job?*

I think one where people share ideas and give generous recognition to the ideas shared by their peers. That generosity can be simply verbal provided that it's sincere.

*Are you a member of any technical writing societies, mailing lists or websites? Do you contribute to the technical writing community in any way?*

Yes, I love sharing ideas with everyone out there and I love receiving their ideas in turn. I think people are very generous. They give feedback very generously.

For example, I have a blog at  
<http://ffeathers.wordpress.com>.

I'm a member of Technical Writing World, which is a really great technical writing community.  
<http://technicalwritingworld.com/>

I'm on Twitter @sarahmaddox.

I'm a member of the ASTC(NSW) and I hold webinars every now and then, such as the recent one that I gave for the Technical Communicators Association of New Zealand (TCANZ).

*Do you have any advice for people trying to break into technical writing?*

I think it's all about getting some experience behind you. And one way to do that is to find an open source project that is looking for contributors. There are open source communities that run docs sprints every now and then. Mozilla or Open Stack, for example. A doc sprint is when people get together to write a specific set of documentation. They're a really good way to interact with

other technical writers and to write some documentation that you can then put on your resume.

Another idea is to monitor technical writing forums and mailing lists, such as the austechwriter list. Often there are short-term jobs advertised there that take just a week or two. They are often editing rather than writing jobs and that is a really great way of getting an 'in' into an organisation and to get some experience for your resume as well.

*Where do you see technical writing going in the future?*

I think the big things will be APIs and mobile devices. An API is an application programming interface that people use to build small applications or apps that hook into bigger services like Twitter and Facebook. So we'll be writing more and more guides for people who want to develop a mobile app. The APIs will become simpler and at the same time more diverse. So we'll need to explain what each one does and how people can use them to build the app they want.

Also, I think people will be building more and more mashups. Those are integrations of one web service with another, for example, or one desktop program with a web service. You may want to integrate Twitter with a wiki, or you may want to integrate an issue tracker with an online diagramming tool. We'll need to take those requirements into account more and more and take account of the fact that the thing we're documenting may look and behave radically differently on other users' sites. This is due to the growing ease of integration and customisation.

We'll also be spreading our help onto more and more channels. So instead

of writing a single monolithic document that people can come to and absorb, we'll need to scatter our gems of wisdom onto multiple channels. And also provide a location where people can find everything in one place if they need it. We'll need ways to bring them to that one place.

And I think that about covers it.



Sarah Maddox

**Check out recommended technical communication resources by Sarah and our other interviewees on the ASTC(NSW) website:**

[http://astcnsw.org.au/useful\\_links](http://astcnsw.org.au/useful_links)



*Recommended resources*