

Learning from testing

Tips for technical communicators from testers



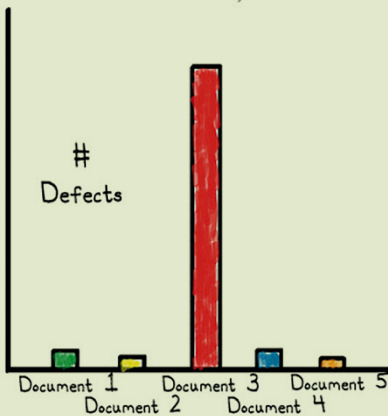
Communicator

The Institute of Scientific and Technical Communicators
Spring 2016



Reduce your long-term maintenance workload

Learn how Green Technology is relevant to you



DocOps and the technical communicator

The impact of Augmented Reality for us



Observations of a newcomer

As a new technical communicator, **Louise Fahey** shares her tips for newcomers to the field.

It was 5 January 2014, three days before I turned 24. I was preparing to leave Ireland for a new job as a trainee technical communicator with BridgeHead Software, a software company in Surrey. My mother tearfully waved goodbye to me at Cork Airport. The weather was stormy and the flight turbulent. I arrived at my accommodation in Epsom to find no one home. My phone wouldn't work. I didn't know a single person in the area.

I learned to use the active voice rather than the passive

And yet throughout all of this, I couldn't stop thinking, or rather worrying, about the new career I was about to embark on. I mean, I had no technical experience, what was I thinking going into software? My degree was in law so why couldn't I have attempted a legal career like my classmates? Oh God, what if I got fired for being terrible? I'd have to go back home. And then how would I face my friends after they threw me such a nice going away party?

All completely unfounded worries as it turned out. Looking back on the past two years, if I could have given myself or any other newcomer to technical communication advice, it would probably be these five tips:

1. Steep learning curve
2. Knowledge is power
3. Less writing than you originally imagined
4. Verbal skills are important
5. Documentation is part of the product

A good technical communicator knows the importance of keeping up to date with the latest industry trends

1. Steep learning curve but stick with it

As with any new job, there is a lot to learn at the beginning. Coming from a non-technical

background, there were three main areas I had to get up to speed on quickly: MadCap Flare which was the content authoring tool we used at BridgeHead, the software I was documenting, and also my own writing style.

A combination of training and practice using the tools and software helped me gain ability and confidence in the first two areas. In particular, I found the MadCap training courses and free webinars excellent. With regards to my writing style, writing had always been one of my strongest skills at university but as a technical communicator, I had to adjust my style completely. I learned to use the active voice rather than the passive and I also got to grips with topic-based authoring.

Looking back now, my first six months were not easy. However, I was fortunate to have an excellent mentor during my time at BridgeHead. It is thanks to the patience and knowledge of Judith Knowles that I gained competency in these three areas after only a few months on the job. Judith taught me that mistakes happen and that's okay. In my first few months, I did something with the version control system that took almost four hours to fix. It was embarrassing and I'm sure Judith was slightly irked but it certainly wasn't the end of the world! People learn from their mistakes and I never made the same mistake again.

2. Knowledge is power - keep striving to learn more

Despite being a year from retirement when I started at BridgeHead, Judith kept up

MORE INFO HERE: www.tceurope.org | colloquium@tceurope.org

co-organized by
apcomtec
 Associação Portuguesa
 de Comunicação Técnica

hosted by
 INSTITUTO
 SUPERIOR
 DE CIÊNCIAS DA
 ADMINISTRAÇÃO
 DO PORTO

to date with the latest trends in technical communication and would pass this knowledge to me. She watched webinars and read various writing blogs. This made me realise that even if I was able to use MadCap Flare and my writing style was up to scratch, a good technical communicator never stops learning, and knows the importance of keeping up to date with the latest industry trends.

This realisation prompted me to consider a formal training course in technical communication and so I applied for a place on the University of Limerick's Graduate Certificate in Technical Writing. I remained in England, continuing to work full time and study by distance learning. A glutton for punishment, I decided to go against the advice of the course director and take this course full time over one year. It was tough. Especially as I changed job and moved from Epsom to Reading in between! But the course was definitely worth it in the end. I found that what I learnt on the job, complemented what I learnt on the course and vice versa. As a result, I would strongly urge newcomers to technical communication to consider some sort of formal training course on top of what you already learn on the job.

I would also urge new technical communicators to further their knowledge of whatever industry they find themselves working in. If this industry is software, then it is useful to look at the development process used to create the software. When I started my current job at Clearswift, I found myself working in an Agile development environment. As part of the documentation team at Clearswift, this means our content creation process must fit the Scrum framework. This took some adjusting to at the beginning. But after months of attending

Scrums, Sprints and Retrospectives, I now can't imagine working any other way. In fact, I enjoy working in this environment so much that I find myself eager to learn more about Agile.

Fortunately Clearswift share this goal and have been very supportive. In September, the company sent me to the Agile on the Beach conference in Cornwall. This conference explored themes in Agile software, business, teams and product management and it gave me a lot to think about in terms of how our documentation team operates within a Scrum team. One of my next goals at Clearswift is to complete training to become a certified Scrum Master (or rather Scrum Mistress!).

3. Less writing involved than you originally imagined

Whenever I'm asked what I work as by friends or new acquaintances, the majority of them look a bit uncertain. So I now follow up my response with the line "I write instructions that tell people how to use the software that my company makes." This usually clears things up for people. However, this over-simplified description doesn't really do justice to what a technical communicator actually does. Yes, there is writing involved. But you shouldn't underestimate the "technical" part of technical communicator.

I remember a few occasions in my first job when Judith would end a day's work with the phrase: "Maybe tomorrow I'll actually write something!" As time went on, I realised just how correct this statement was. As technical communicators, we spend a lot of time planning our content and getting the information we need before we can actually write something. In the course of this planning and research, at least when working for a software company, be prepared to set up virtual machines, install software, raise bugs,

"Maybe tomorrow I'll actually write something!"

If you approach people politely and with respect and appreciation for their work, generally you can get all the information you need

"We have been very happy with the service provided by 3di. They have the know-how, the relevant experience and the project managers to deliver our projects and adhere to our high standards,"

Joseph Nosbuesh, Head of Documentation Management, Roche Diagnostics



Complexity made clear

Expert documentation & localization services

Technical Communication

- supply of technical authors
- managed outsourced teams
- information & document design
- modular writing training
- tools & process strategy
- software usability testing

Localization & Translation

- software products & online help
- website & e-learning content
- interactive & audio content
- technical & compliance documents
- multi-lingual translation
- scalable localization testing

☎ 01483 211533

✉ contact@3di-info.com

🌐 www.3di-info.com

close bugs, decipher technical terms that sound like gibberish at first, check out projects from online repositories, and much more! The good news is that none of these tasks are impossible. You'll have support and in time you'll even pick up how to do most of them yourself. And if, like me, you entered the profession with little to no technical experience, when the day comes that you can do these tasks yourself, it is extremely satisfying!

Be confident that what you create is part of the product

4. Verbal skills are as important as written – don't be afraid to ask

As a technical communicator you will deal with subject matter experts (SMEs) on a daily basis to get the information you need to do your job. This seems pretty obvious. However, when I started working as a technical communicator straight out of university, I was of the mindset that I could get the information I needed myself through the Internet or past documentation. Not so! How can you research information for a product that has just been developed and hasn't been released before? There might be internal documentation or similar products on the market but ultimately you are going to have to ask someone about it. Probably a few times too.

At the beginning, I found this a bit daunting. There is a stereotypical image of software engineers that says they lack social skills and I must say I was reluctant to bother them with my "stupid" questions. Of course both parts of this sentence turned out to be false – the majority of software engineers are as sociable as the next person, and there is no such thing as a stupid question. If you approach people politely and with respect and appreciation for their work, generally you can get all the information you need. As time passes and you build professional relationships with colleagues, this all becomes much easier. In the meantime, here are a few tips that I've found work well:

- For smaller queries, I prefer sending an email or instant message. I go directly to an SME's desk if I have a bigger question or if I don't get a response to an email.
- Take advantages of meeting people by chance in the kitchen. General chit chat like "How's the release going?" can provide you with an update on the status of a project.

- Don't be afraid to set up meetings with SMEs – you will have their undivided attention and even a short half hour meeting can provide you with invaluable information.
- The whiteboard is your friend. If something doesn't make sense in words, ask an SME to draw a picture. I use my camera phone to take photos of this picture so I can reference it after the whiteboard has been cleared.

5. Documentation is part of the product

Based on what I've heard from more experienced colleagues, in the past technical communicators sat on their own in a room and were not seen as part of the development team. Documentation was seen as an add-on, a non-essential part of the product. This is changing. One of the many advantages of working in an Agile environment is that documentation is now rightly seen as part of the product. Technical communicators now stand alongside programmers in daily Scrums and have much more visibility than they did in the past.

At Clearswift, documentation is a deliverable and an item that must be checked off in review meetings before a product can be released. As Clearswift respects documentation and the hard work that we technical communicators do, we respond by providing the best content we can. So my final tip to new technical communicators is to be confident that what you create is part of the product. This applies to technical communicators working in any field – it doesn't matter if your end-user is building a chair or setting up a policy to block spam emails, without your documentation, they are unlikely to get very far!

Conclusion

So there you have it. The findings of two years' experience. There have been ups and downs but I wouldn't change any of it (not even that version *control* mistake that took four hours to fix!). I now find myself in a career where job security is strong thanks to a constant demand for my skills, and career progression is fast depending on how hard I work. Do I ever regret not going into law after my Bachelor's degree? Never! I'm having too much fun ~~playing with~~ documenting software. **C**

References

- Bardez, J.P. (2014) 'Training in verbal skills' *Communicator*, Autumn 2014: 42-43.
- Fahey, L. (2015) 'Studying in the virtual classroom' *Communicator*, Autumn 2015: 18-20.
- Leahy, C. (2013) 'It's a career, but not as we know it' *Communicator*, Spring 2013: 26-27.
- Westfold, A. (2015) 'Writing in agile environments' *Communicator*, Spring 2015: 12-15.
- Manifesto for Agile Software Development*
www.agilemanifesto.org (accessed January 2016)



Louise Fahey is a Technical Author at Clearswift, based in Reading, UK. She holds an LLB degree in Law Plus and a first-class honours degree in Technical Writing, both from the University of Limerick, Ireland. She is also a big fan of all things Agile.

E: louise.fahey@clearswift.com

W: www.clearswift.com

L: <https://uk.linkedin.com/in/louisefahey3>