March 2022

**Docs for developers**

An engineer’s field guide to technical writing

By Jared Bhatti, Zachary Sarah Corleissen, Jen Lambourne,David Nunes, Heidi Waterhouse



Docs for developers: book review

Reviewed by Natalia Dowding

Unlike many other publications in the Technical Writing field, this book is intended for developers who face the necessity to create their own documentation rather than writers. The aim of the authors was to equip developers with a comprehensive guide to creating successful and effective documentation.

This relatively small volume covers several topics, leading the aspiring writer through all steps of documentation creation, from audience analysis to maintenance and deprecating. Each of these topics is a big subject on its own. The authors opted forextracting the bare essentials and leaving the details behind. In addition, extra references and a list of useful resources are available in the Appendix.

To illustrate the process and the pitfalls that developers who are undertaking writing tasks encounter, the authors include a story of a fictional software company named Corg.ly, thatdevelops a creative toolkit used tor translate dog barks into English. At the end of the development process, the company faces the need to create the required user documentation for their product, and each chapter demonstrates through daily situations, how the writing process should be tackled.

## 

## What is between the book covers

The volume shares the detailed process of creating documents,from planning to performing user research and handling content maintenance. The great bulk of the first part of the book deals with the writing aspect, taking us step by step through each stage of writing and publishing. The second part introduces the practicalities of gathering feedback, establishing quality metrics and planning the maintenance.

Throughout the book, the authors show lots of sympathy towards developers who end up having to wear a writer’s hat, and wherever appropriate, emphasise the similarities between coding and writing. They -mention on several occasions that the stumbling blocks in the writing process are not caused by the inability of developers to write, but because writing is difficult. Indeed, writing is not what developers do on a daily basis and there are significant differences between coding and writing.

Facing a blank page can be daunting to anyone, including experienced writers. The book claims that a good amount of planning is key, hence writing is part of a process.

Getting into the details

As mentioned before, audience analysis plays an important role on this process: in terms of content suitability and to empower the developer, so they can understand the real purpose of the documentation. The authors provide a few practical methods of user research.

Chapters 2 to 4 cover the planning, drafting and editing stages of document creation. I particularly like these chapters because they equip the developer with a strategy. For developers, who might see writing as a vague, creative process, this will help them to start off on the right foot.

Types of content

It is interesting to see different types of content being categorised along with examples of common types of documents that go along with each type of content.

For example, tutorials areprocedural while design notes areconceptual . For each category authors discuss the structure of the document: a how-to guide will likely contain a number of steps to achieve a necessary result.

Trial and error

Introducing structure is important for a beginner writer, and having a pattern or template to followeliminates the difficult task of designing some sort of structure from scratch, while allowing devs to learn from theexperience of others and having more time to concentrate on writing alone.

As drafting can be challenging,

It can be very tempting for non-writers to publish their document straight after creating the first draft. Comparing editing to testing must be the most convincing argument for developers. Where possible, authors draw parallels between coding and writing processes, be it conceptual similarities or mere processing environment.

As before, the authors propose a strategy to follow, they show a few points to keep in mind when editing and reviewing, to help the developer to go through editing tasks easier.

Visual elements

Unlike with writing, developers feel more athome with code samples. Facing a relentless stream of new technologies, apps, tools and languages to deal with, developers know the relevancy of code samples to better grasp new material.

The reader is presented with good practices for creating effective and reliable code samples. Thoroughly testing of samples against the current version of the product is also encouraged.

The authors explain the practical steps of creating efficient and appealing visuals. This approach allows them to put priorities in the right order: structure comes first, aesthetics - second. The authors also leave some interesting resources for developers to draw upon.

Unlike code samples, the publishing stage might be less known to developers. Here developers will find out that publishing is not so much about the choice of tools, but about planning, which makes all the difference. This is one of my favourite chapters because again, all the basics are laid out in a succinct, clear way.

Content freshness

Ensuring that the documentation is up-to-date at all times helps when building trust with users. In a few short sections, the authors manage to distil the best practices for maintenance and deprecating documentation.

Content evaluation

Gathering feedback and measuring the quality of documentation are aspects that are often overlooked. The authors present a number of practical ways for collecting feedback, introducing improvements and evaluating the quality of the documentation using various metrics so that the task appears more manageable.

Similarly, the authors propose several practical ways for the developer to evaluate the quality of the documentation. They explain the concepts of functional and structural quality and provide methods to translate them into meaningful values.

For many developers, documentation metrics would be a novel concept which is why this chapter may be of special interest to them.

Information Architecture

Making sure that users can find their way through documentation, and that the information is appearing logically and intuitively, plays an important part in establishing a good relationship with the reader and product user.. Nowadays many developers would have some exposure to IA, but here the authors provide a concise guide to how developers can use it for their own documentation without going into the details.

## Reviewer’s comments

The first impression is that in the attempt of simplifying the presentation and concepts presented, the authors went a little too far, making the book a kind of “plan of attack” for beginner writers, but lacking in detail However, if you look at it from the developer’s perspective, you start to appreciate the author’s approach more.

Take, for example, an explanation of a paragraph. Presenting a paragraph as a section of text which provides a context for a procedure or that it may contain a story illustrating a concept, rather than as a logical subdivision of text, seems to be confusing and hardly logical.

Overall, I feel that the book might benefit from more examples. The story of Corg.ly does not fill that gap for me. There is a feeling of imbalance between the overly cheerful Corg.ly company and an abstract presentation of subsequent material.

At times, I felt the text is overly simplistic

But having said that, I liked the overall style of the book: clear, cheerful, sympathetic and welcoming to a diverse community of devs

I noticed the lack of software references, which seems to be a wise approach when taking into account one of the lessons shared in the publication, the importance of keeping the content as fresh as possible, and considering the variety of products available and the speed with which they go obsolete, this is a veryrational approach.

In my opinion, writing is difficult because you can’t always follow a prescription, and I would have appreciated if more resources about style and language were available.

Overall, I think that development teams facingthe prospect of creating their own documentation should read this book. It is full of valuable advice and exemplifies the very style in which the documents should be written: clear and concise. The book might help you toovercome hurdles on the way to creating documentation.