

Chapter 5: Writing for Software Developers

I have read a number of books about writing, from Nathan Barry's *Authority* to Stephen King's *On Writing* to recent editions of *Writer's Market*. In my experience, books about writing (except for style guides) tend to dedicate only a small fraction of their length to what happens when the writer is actually writing. Until I sat down to write my own book about writing, I found this disappointing. How could you write about writing without, well, writing about writing? It turns out that there are two primary causes for this widespread phenomenon. First, writing is a very individual process that is hard to describe in the abstract or teach conceptually, unless you are addressing specific aspects like grammar or usage. Second, most of what happens around writing is the opposite: easy to describe, but not intuitive to figure out. Thus, I am joining a long tradition by dedicating only one chapter to writing.

I often get asked about my writing process, and I also ask other writers who I respect about their writing process. The writing process tends to be intensely personal to each writer, and I don't necessarily know that it is generalizable. I don't write in the way that some of my favorite writers write, nor have I been able to teach people successfully how to write in the way that I do. I have generally been unable to operationalize things that people have attempted to explicitly instruct me on with respect to making my writing process more effective. —
Patrick McKenzie

The writing process is the hardest part of this whole endeavor. David McCullough, a Pulitzer Prize and National Book Award winning author, famously said, "Writing is thinking. To write well is to think clearly. That's why it's so hard." Adapted to software

engineering language, writing about programming forces us to operate at a different level of abstraction than we normally do while programming, and to do so along an axis that is perpendicular to the levels of abstraction that we more regularly traverse. However, sufficient work on outlining and, when needed, sample code can make the writing itself feel achievable.

5.1 The Actual Words

With completed sample code (if appropriate), research, and an outline completed, your actual writing process should feel steady and achievable. It is still the hardest part, in my experience, but with enough time and practice, it should begin to feel like a regular part of your work rather than a recurring miracle.

The testing and gathering phase continues for a while, until I feel like, “Okay, in broad strokes I know the approach I’m going to take on this piece.” Then, I get to the extremely non-replicable portion: I sit down at a desk, drink a lot of coffee, and write like a man possessed for six hours. — *Patrick McKenzie*

For a book on writing, I have been remarkably short on metaphors. I seek to rectify this oversight by asking you to imagine that you are building a bridge for your readers to cross. On one shore stand your readers, secure in the land of existing knowledge. Your task is to construct a safe passage over a wide strait of uncertainty to the island of new understanding.

Now, you cannot simply lay planks for miles and hope that they stay rigid. To cross wide spans, you must construct additional pylons to support the overall structure. Sample code, cited facts, diagrams and images, and other discrete certainties are the pylons spanning the strait. The outline determines the order and distance

between these structures. Your job in the article itself is to span each gap with explanation and intuition. Matt Levine, for example, writes his newsletter using block quotes as pylons:

A lot of my audience is in some way technical and is interested in the technical details of these things. The block quote is the equivalent of the code; they want to see the guts of the thing. My commentary will be some sort of deep structure explanation or some sort of joke or something, but I'm not going to tell the guts in a way that is more accurate or specific than what the guts are, so I might as well just show them. Again, people can skip it. People who are interested in the technical details are going to read the block quote and people who are looking for the jokes can skip right over the block quote and not miss the jokes. It's a nice way to segment the readership. — *Matt Levine*

The two main tools for your task are the questions “Why” and “How.” Before each pylon, establish why it is needed. After each pylon, describe how it functioned to get you closer to the destination. Motivating and explaining code helps avoid the trap of simply repeating yourself by explaining concepts in code and again in words. Other classic questions from journalism, “Who,” “What,” “When,” and “Where,” are usually best answered in the introduction and conclusion. In our metaphor, these are the on and off ramps of the bridge.

5.1.1 Craft and Style

At the craft level, there are a few things to keep in mind. These are general heuristics that make sensible defaults. You will know when it is right to deviate from these guidelines. The most common reason to deviate is to conform to a given publisher's style guide.

I prefer to write in the first person (I write this) and second person (you write that), as I have been doing for this whole book. I especially like the first person plural (we do this, we now understand that). To my ear, this establishes a conversational tone and familiarity that mimics explaining a concept to someone one-on-one. “I” and “you” establish distinct characters in your text. As a general rule, I try to avoid using the third person (one writes that). However, for articles about more abstract concepts, I will sometimes write in the third person. This helps when synthesizing and analyzing sources or conveying other concepts that I do not have direct experience with.

Default to present tense. Of course, use the past tense to describe the past (Python 3 was released in 2008) and the future tense to describe the future (Django 3.0 will be supported through 2021). Otherwise, stick with the present tense for the bulk of most articles because the reader is following along with your tutorial in the present moment. Thus, when you write instructions like “copy the following HTML into *index.html*,” you are describing the action that they are taking in the moment that they are taking it in.

Most writers and editors tell you to avoid passive voice. Passive voice is where the object of a sentence is being acted upon by the subject. In active voice, Sarah writes the program, in passive voice, the program is written by Sarah. In general, I agree with this advice. In technical communication, it is critical to identify the correct cause and effect in a system. Active voice combats ambiguity.

Keep your syntax and usage accessible but interesting. Do not throw around ten-dollar-words just for the sake of it, but be precise in your diction; if there is a word that expresses your exact meaning, use it. At the same time, be generous to your reader. If you suspect that the precise term may not be familiar to your

reader, offer a definition of the word or acronym after you use it for the first time in your article. Keep your target reader in mind; define any terms, technical or otherwise, that might trip them up. For example, in the article in Chapter 8, I define “iambic pentameter” but assume that readers know what “URL” means.

Vary your sentence length. A long sentence can be helpful in expressing a complex connection, its multiple clauses conveying substantial detail. Be careful not to sacrifice readability to length. Solid, clear prose with a bit of flair where appropriate will establish your presence as a writer and will communicate your ideas effectively.

I think there’re fine lines between too much personality and just enough personality and almost none. Some people will use their personality and apply it to the article, but then it takes away from the actual content. There’s a lot of fluff that I remove from guest-author articles, and that just goes to skimmability. People want a couple lines of personality and then they want to get to the actual good parts of the technical stuff of the article. Definitely try to test out how things are. When I say test out, I mean definitely try different writing styles, push it out, see what the community thinks and get feedback from that. Writing is such a dynamic thing; it’s a give and take between you and your readers. Just see what your readers like and go from there. — *Chris on Code*

5.1.2 Voice

There is a perfect secret for developing a clear, natural voice in your writing. Using this technique, you will learn to channel your personal, inimitable style into your writing. I have not yet mastered

this approach to voice. This mystic wisdom is easy to write but hard to read.

Here it is: there is no point in attempting to develop a voice in writing because you already have one. That is the secret. That does not mean that you should not pay attention to how your writing sounds. When evaluating your own work, look for phrases that do not sound like something you would say in real life. This is what needs to be fixed for your voice to naturally reveal itself over time. Imitating other writers' style is an incredibly useful exercise for developing your craft and specific techniques, but outside influences should melt into the background that *you* bring to each written word. While this is a process that will happen, it does take time and consistent practice.

If you just write for publication every day, you'll develop your own voice naturally. My perception was that blogging, writing on the internet, was like writing an email to your friends. There's really no better way to write naturally and in a conversational style that sounds like yourself than thinking about your writing as writing an email to your friends. I had experience writing emails to my friends at Goldman Sachs and making jokes and trying to be funny while also assuming a pretty high level of specific financial knowledge. That was a useful experience to draw upon when writing for a publication. In fact, I remember six months or a year into my job at Dealbreaker having dinner with some of my Goldman friends and one of them saying, "You know, you really sound like you now. You didn't used to sound like you when you started; now you sound like yourself. I read you and I'm like yeah, that's Matt." That was a validating moment for me. When I write, it clearly is a persona, but

the main aspects of developing a voice is not so much developing that persona but stripping away artifice and things that you've tried to pick up from elsewhere so that the writing sounds more like your conversational style. — *Matt Levine*

More than a byline or author bio, your voice is your signature on the page. It is why teaching writing is so difficult. It is why you like some authors more than others. It is more than just words, it is how those words occur to you that matters. Practice, but trust yourself. Trust your voice.

For differentiating yourself in writing, I think it's really just about finding your own voice. That comes with practice. I graduated from Iowa State University. At Iowa State I blogged for the university on cyclonelife.net. I had to write three or four blogs every week for them and so it was really just a matter of practice. The more I wrote for them the more I developed what my tone sounded like and what my voice sounded like.... I think that's what a lot of people are hesitant to do, they're like, "Ah, well, I want to blog, but I don't really have a voice. I need to develop that voice." I think that just comes with putting pen to paper, writing stuff down, and actually writing out blog posts and developing it over time. — *Cassidy Williams*

5.1.3 Your Writing Practice

Each time you sit down to write, focus your attention on one discrete part of your article. If the article's sections are too big to tackle, break them down. With only your outline, research, and relevant code in front of you, write a section in simple, straightforward words. Try not to get hung up on making the

language pretty or elaborate, just get the words on the page. It will be much easier to adjust your work when you have a piece to edit.

You know if you are a marathoner or a sprinter, and you can set your writing schedule accordingly. Planning to write a little bit of the article, or one plank of the bridge, at a time will take some of the worry out of writing. Plan your writing time; these writing blocks do not need to be long, twenty to forty-five minutes, but they should be as sacrosanct as any other appointment on your calendar.

In his booklet *Learning Technical Writing Using the Engineering Method*, Professor Norman Ramsey presents eight practices of successful writers, which he has generously allowed me to reproduce here.

1. *Pause mindfully, frequently.* Mind your body, thoughts, and feelings — and the stage of your work.
2. *Write in brief daily sessions.* Ignore the common myth that successful writing requires large, uninterrupted blocks of time — instead, practice writing in brief, daily sessions.
3. *Focus on the process, not the product.* Don't worry about the size or quality of your output; instead, reward yourself for the consistency and regularity of your input.
4. *Prewrite.* Don't be afraid to think before you write, or even jot down notes, diagrams, and so on.
5. *Use index cards.* Use them to plan a draft or to organize or reorganize a large unit like a section or chapter.
6. *Write a Shitty First Draft™.* Value a first draft not because it's great but because it's there.
7. *Don't worry about page limits.* Write the paper you want, then cut it down to size.
8. *Cut.* Plan a revision session in which your only goal is to cut.

Keep these principles in mind as you develop a steady, reliable writing practice. As I learned while writing articles and learned again while writing this book, time in the chair is the only way that words get written.

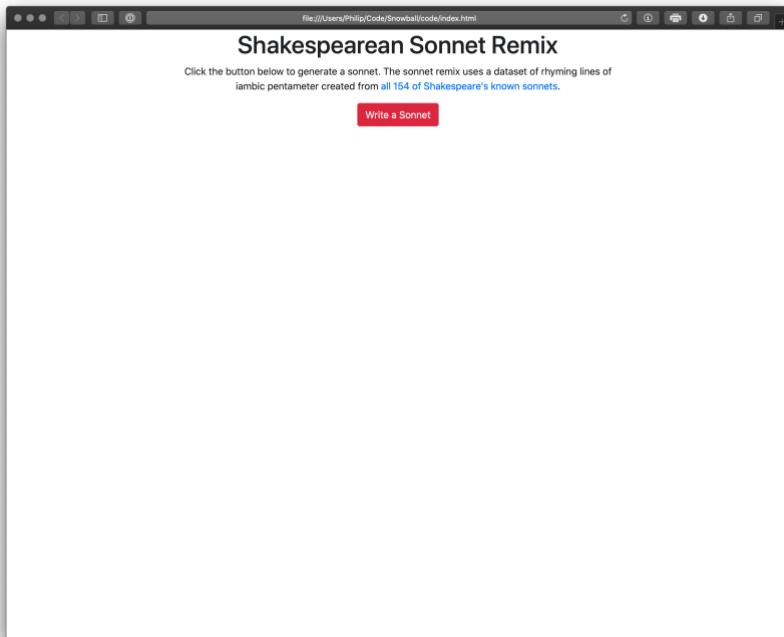
5.1.4 Markdown

Write in Markdown. Unless you are a wizard with LaTeX or your client specifically requests a different format, Markdown is the smart default: you can write Markdown in a different buffer of whatever editor you are using for your sample code; it has all of the formatting you need for both words and code; it is compatible with version control; and it is interoperable with most publishing systems, or at least it is easily convertible into a format that is. If you do not know Markdown already, it is among the most powerful tools you can learn in five minutes.

5.2 Graphics

As you are writing, you may realize that your words and sample code would be even clearer if supported by graphics. Screenshots, pictures, and diagrams are a supplement to, not a replacement for, good writing and complete sample code. Make notes to yourself while writing your first draft as you realize a need for any form of graphic. Once the draft is complete, move on to creating them. If you have trouble finding or creating graphics, reach out to your editor. They can help you with the process, and sometimes will find or make graphics for you.

There are three kinds of graphics that I use in articles: screenshots, pictures, and diagrams. They each have their own use cases, advantages, and drawbacks. Graphics are a great way to break up walls of text and keep readers engaged, but when overused can overburden the writing.



Screenshot from Chapter 9's Tutorial

Screenshots are useful when you are creating a tutorial on building or interfacing with anything that presents a graphical user interface, be it an app or website. A screenshot of the full system at the beginning of the article can pique a reader's imagination. A few screenshots at key checkpoints can help readers ensure that they are following along in their own implementations without issue. Pay careful attention to ensure that your screenshots match the actual state of the system as developed in the article rather than during the experimentation phase in sample code writing. When using screenshots to guide your reader through an existing user interface, make sure you are consistent and complete in describing the

actions that they should take and the feedback they should receive. In all cases, it is useful to annotate screenshots by circling or highlighting key elements for the reader to focus on.



Picture from Chapter 10's Article ("Macbeth: First Folio" by Matt Riches on Unsplash)

Pictures provide visual context for your writing. Unlike screenshots and diagrams, I generally do not create my own pictures. Fortunately, there are a number of royalty-free, attribution-free, and other open-licensed image repositories online. My favorite is Unsplash, but there are many others. My most common use case for a picture is as the article's header image for blog software. My second most frequent use is to break up sections in long, abstract articles. Avoid using too many general pictures because they do not directly contribute information to a piece and your readers have seen similar images hundreds of times in the same context.

Sonnet
XII

ANATOMY OF A SONNET

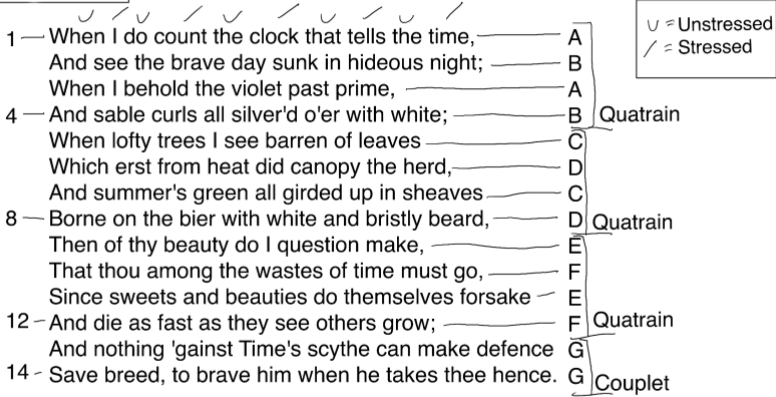


Diagram from Chapter 8's Tutorial

Charts, graphs, and other custom diagrams are the most useful graphics that you can include, but also the most time-intensive to create. My workflow of choice is Autodesk Sketchbook on the iPad with the Apple Pencil, but there are dozens of software options for both handwritten and computer-generated graphics. You can go as low-tech as a pen, paper, and a smartphone camera. What is important is finding a tool that works for you and becoming proficient with it. With charts, you can help visual learners engage with your work and you can express complex relationships that are difficult to convey in writing. Diagrams also serve as excellent introductions or summaries for concepts; they provide readers with a concrete expression of your ideas.

5.3 Overcoming Writer's Block

Everyone experiences writer's block differently. "Writer's block" is the failure to make effort or progress on a writing task. For me, it generally occurs when I am working on an article but do not have a clear next step. Regardless of the stage of the project, it is hard to sit down to work on something unless I know roughly what I am supposed to be doing. This issue can stem from one of several causes.

Sometimes, there is a gap in my knowledge of the subject, a sort of "unknown unknown." To solve this issue, I read more sources, write more test code, explore more real-world systems, or otherwise step away from the writing process and back into research. I do not believe in waiting for inspiration, though thinking about writing does take time. Keep researching and having both intentional, structured thinking and ad-hoc shower thoughts until you come to a breakthrough.

One part that is really important to me is that I will read a complicated thing and something will hit me on an emotional level. Something will hit me where I'm like, "that's beautiful" or "that's an amazing thing that they've figured out" or "what a crazy trick they did" or "look how convoluted and strange this was." I will have a reaction that is direct and visceral and if I don't, I won't write about it. The trick is that something intuitively resounded in me that made me have that reaction and I want to create or explain that reaction for my readers.... If you're trying to explain a complicated thing, period, it's complicated. But if you have a strong, clear reaction to the thing, walking people through your steps to get to that strong, clear reaction is an easier thing to do because you

feel it, you want to make jokes about it, you want to express it in a clear way. — *Matt Levine*

Sometimes I know enough to write about the topic, but the topic is way too big to fit into a reasonable article. This means that I made mistakes while outlining. The first step is to rigorously narrow the article's scope by considering the audience for the article. I remove both prerequisite knowledge and less valuable tangents based on explicit assumptions about my target audience. Instead of including these tangents, I can keep my focus narrow by linking to other sources at the end of the article to give the reader a more in-depth look at my subject.

Beyond the scope and my knowledge, sometimes the structure, outline, or topic of the article does not hold up during the writing process. Perhaps the lesson turns out to be obvious, the code trivial, or the article derivative. Again, this is a breakdown in the outlining and research process, so the best step is to set aside the article as it stands and revisit research and outlining to find a unique angle for the article or a better match for the existing content to a readership.

To write good tutorials you have to be able to put yourself in the shoes of people who need them. So if it's help tutorials, if it's for things that are broken or how to set something up, you have to put yourself there. You really have to try it from that perspective, even if that means blocking off some of your existing knowledge and approaching this as if you were the person who you're writing for. — *Angel Guarisma*

I would be remiss to neglect how your working environment affects your writing. When I am constantly checking social media, dealing with distractions or interruptions, overly tired, or otherwise not in a

good physical or mental place to work, I do not make progress. I treat writing an article the same as I do writing code; writing requires the same level of care in constructing an intentional, effective working environment.

I avoid writer's block by scheduling my writing time, working on small, achievable sections of my task, and relying on my research, outline, and sample code. I also have reinvested some of my earnings from writing in equipment that is a joy to use. I write almost every day and am flexible in changing my methods to adapt to my environment and circumstances. I even have dedicated writing playlists. This is what works for me; use it as a starting point in figuring out what works for you.

//TODO

1. Write your article. Do not try to write the whole thing in one sitting. Break down writing sessions according to your outline.
 1. Write the introduction to your article
 2. Write the first section of your article
 3. Write the second section of your article
 4. ... Write the nth section of your article
 5. Write the conclusion of your article
2. Find or create graphics that you think your article might need. If you get stuck, reach out to your editor for help.
3. Do not worry about word choices or the flow of your article until you have a complete draft. It may be very rough, but what is important is having something to revise.