

THE **5 KEYS** TO SUCCESSFUL **SOFTWARE** DOCUMENTATION



If you were the coach

Of a championship basketball team, would you sit your most valuable player on the bench?

Of course not. But many businesses leave their documentation on the sidelines in the game of “employee training and customer support.” They send everyone else into the game – instructor led training, community forums, email support, online chat and telephone support – while their documentation sits idly on the sidelines, ignored and useless.

But your documentation doesn't have to be benched.

You can create and deliver great software documentation. And once you start creating and delivering great documentation you will see significant improvements in customer satisfaction and business efficiency.

The information that follows will give you a simple framework that will get your documentation off the bench, get your documentation into the game and turn your documentation into the most valuable player on your customer communications team.

So let's get started creating successful documentation!

THE 5 KEYS TO SUCCESSFUL SOFTWARE DOCUMENTATION

The 5 keys will change the way you think about documentation, the way you create documentation and the way you deliver documentation. They are listed below:

- 1 **Start with the end in mind**
- 2 **Author your documentation in small chunks**
- 3 **Show, don't tell**
- 4 **Attach your documentation with Velcro, not cement**
- 5 **Use process not projects**

The 5 keys may sound a little strange at first but keep reading and you will quickly see how simple and effective these concepts are.

1 START WITH THE END IN MIND

When you sit down to create software documentation, how do you begin? Do you start to write about each screen of your application? Do you start to document each feature of the application?

These common approaches may seem logical but ultimately deliver disappointing results. Let's work backwards. Documentation should serve as the go-to reference point for questions that may otherwise end up on a help desk ticket. The right approach to creating and delivering information will save both you and your customers time and frustration. So let's start with the end in mind.

Start with the end in mind

Remember, people won't read your documentation, they will reference it. Document specific tasks in your application, not screens or features.

● Documentation isn't read, it's referenced

When is the last time you cuddled up by the fire with a great user manual? Your users aren't going to do that either. In most cases people won't read your documentation until they are already stuck and confused.

When they get to that point, your documentation should serve as a **reference with clear answers** to their problem.

Thinking of your documentation as a reference tool should change the way you author it. Don't assume that a reader of one help article has read the others in a particular order — or even read them at all. You need to design your documentation to be easily referenced so that your customers can use it to get “unstuck.”

● Document tasks; not screens

In most cases users are going to reference your documentation when they get stuck. They will get stuck when they are trying to complete a specific task with your application.

Therefore the most useful type of documentation you can create is task-based documentation. Task-based documentation focuses on what users want to do with your software as opposed what screens or features your software contains.

Traditionally you might see documentation for a software application broken down as follows:

- **The Account Screen**
- **The File Export Dialog**
- **The Preferences Screen**
- **The Formatting Window**

But if you focus on task-based documentation and focus on how your users will most likely use your documentation you would be better off creating documentation with sections such as these:

- **How to create an account**
- **How to export a file**
- **How to change your email preferences**
- **How to format a paragraph as a block-quote**

By thinking about how and when your users will use your documentation before you start writing, you will do a much better job of structuring and organizing your content.

2 WRITE CONTENT IN SMALL MODULES

Long software manuals are painful to write, read and update. Many users who see a 5000 word article or a software manual that resembles an encyclopedia won't even begin to read your documentation. So how can you deliver software documentation in a complete and digestible format without intimidating your users?

The key is to author your documentation in small modules. As you get started, use a specific question as the title of each help article to deliver information in a concise and helpful way. The more specific the question, the better.

Write Content in Small Modules

Title each help article with a question, make the question as specific as possible, Write many small articles instead of a few long articles

! Example 1

"How do I use the account screen?"

This is a bad title. The question is too broad in scope to be addressed in a small chunk of information. As an author, you don't have a clear idea of how much or how little information you should include about the account screen.

? Example 2

"How do I create an account?"

This is much better. The title is very specific. As an author you know exactly how much information you need to include.

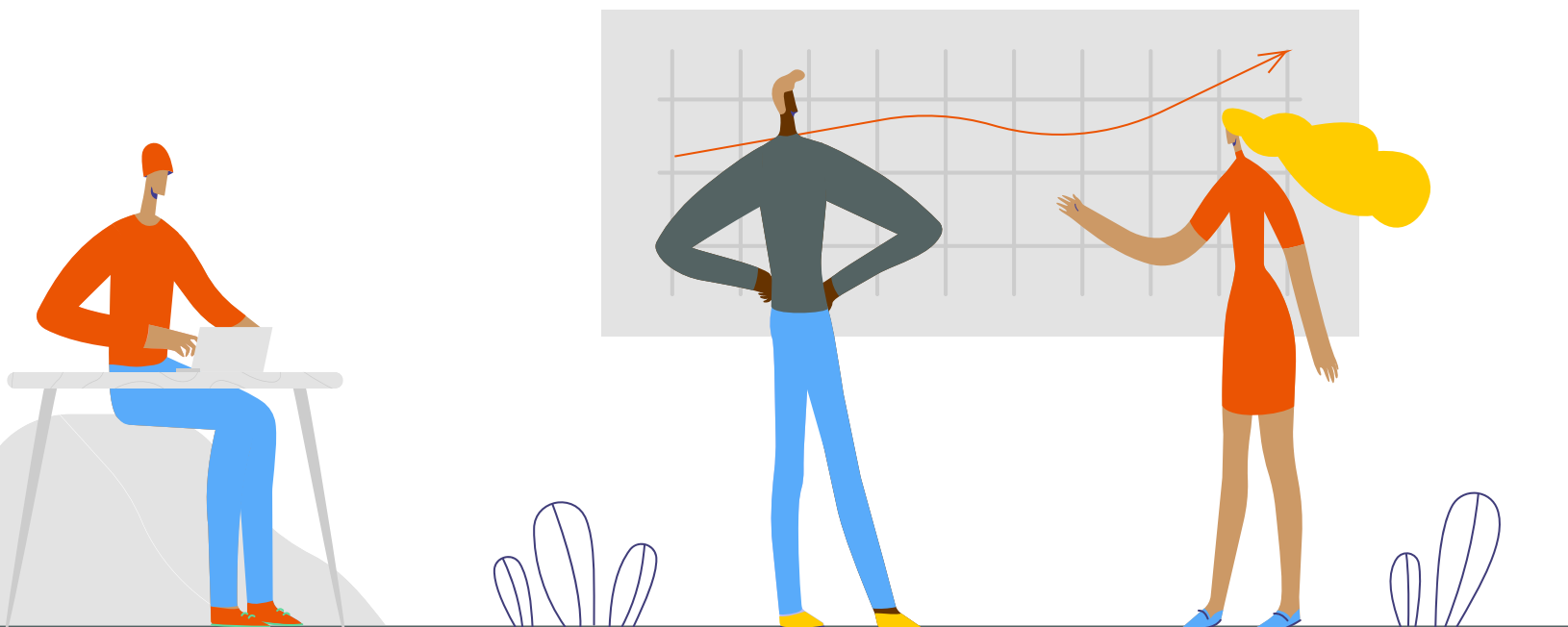


There are two ways in which authoring in small modules will benefit you:

- You will be able to write articles in much less time. If you create a great, specific title, the help article practically writes itself. You don't have to think too much about the content because you are simply responding to a question.
- You will be able to better deliver exactly the information your customers need when they need it. Your customers are often going to get stuck when trying to accomplish specific tasks. If your customer is trying to "create an account", which article would be more useful to them, "How do I use the account screen?" or "How do I create an account?"

When you take this approach the number of help articles you write will increase while the time to author them will decrease.

You will create more content in less time.



3 SHOW; DON'T TELL

Technology has advanced in many amazing ways, but for some reason most documentation looks like it was written on a glorified typewriter. It's all text. The benefits of including images and screen captures in your documentation are too significant to ignore:

● Clarity

A picture really is worth a thousand words. By adding screenshots and images you will eliminate the ambiguity that is inherent in text-only instructions.

● Brevity

With pictures you can communicate much more information with less text. That means less reading for your users and less time they have to spend trying to understand your documentation.

● Speed

By using screenshots, images and the right authoring tools you can create useful documentation in much less time.

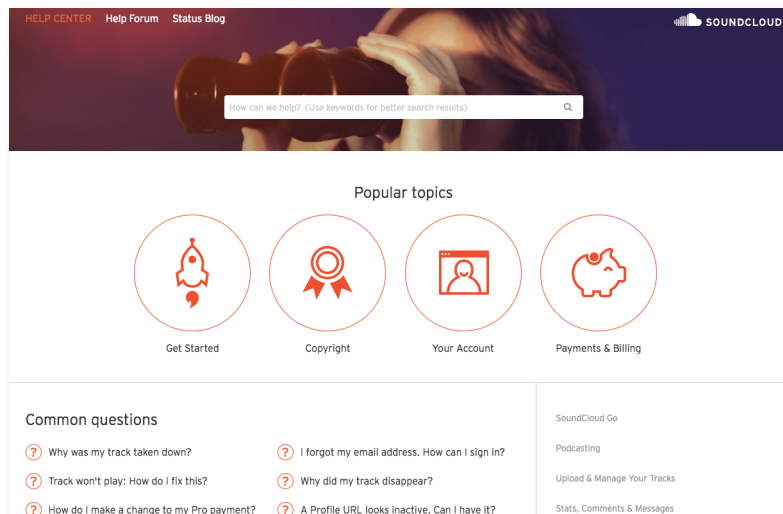
However, you need to use pictures correctly for them to be an effective addition to your manual.

Adding a single screenshot to each section of your documentation isn't enough. A user should be able to scan through a series of screenshots and **understand exactly what they must do to** accomplish their task.



EXAMPLE

Look at the examples below. The same article is presented, one with pictures and one without. Which article would you rather read?



Popular Topics

Get Started

Copyright

Your Account

Payment & Billing

- Why Was My Track Taken Down?
- Track Won't Play: How Do i fix this?
- How do i make a change to my Pro payment?
- I forgot my email address. How do i sign in?
- Why did my track disappear?

QUICK CHECK:

Look at your current documentation. Could you reasonably create the same content using an old IBM electric typewriter? If so, you need to add more pictures. As a help author in this day and age your primary tool should be a camera (or screen capture application), not a typewriter.

4 ATTACH YOUR DOCUMENTATION WITH VELCRO, NOT CEMENT

Creating software documentation that is not modular is the single worst mistake you can make when delivering your documentation.

If the individual parts of your documentation can not be easily accessed then it is like you are attaching your documentation with cement. How do you prevent your documentation from being modular?

- **Deliver your documentation as one large PDF or Word file**
- **Create long web pages that contain multiple topics on the same page**
- **Use help authoring tools that don't create unique URLs for each topic**
- **Deliver your documentation in a format that doesn't include search**

Documentation that isn't modular is **impossible to pull apart**. When you deliver your documentation using these approaches you severely limit your ability to use documentation in customer support situations.

The Twitter Test

To see if your documentation is sufficiently modular take the "Twitter test." Take a question that one of your users might ask. Now try to answer that question in 140 characters or less using your existing documentation. Can't do it? Then your documentation isn't modular enough.

Here is a more detailed example: Let's say that your documentation is on the web, but each page contains multiple topics on the same page. When a customer asks a question you point that user to the URL for that page. But then you have to explain where on the page they need to look to find the answer. You are asking your users to play hide and seek with your documentation!

EXAMPLE

Question:

How do I export a CSV file?

Bad Answer #1:

"Open the PDF help file and go to page 64.

Look halfway down the page where it says 'CSV files.'"

Bad Answer #2:

"Go to this page <http://mycompany.com/massive-help-page.html>.

The information you want is about halfway down the page under

'How do I export CSV files?'"

Good Answer #2:

"See here: <http://mycompany.com/velcro-answer.html>"

Here are the rules for help content that is connected with Velcro:

- It must be delivered via the web.
- It must be searchable.
- Each topic must have its own unique URL that could be pasted into a support ticket, email, forum post, support chat or Twitter response.

Once you have met these requirements you can start using your documentation in really interesting ways. Your support agents can answer simple questions with a single link or more complex ones by providing links to several articles.

Attaching your documentation with Velcro is the key to getting your documentation active in the customer support game.

5 USE PROCESS; NOT PROJECTS

Does your organization work on documentation projects?
Documentation projects involve four phases:

- **Clarity** - You decide what to document
- **Doing** - You write the documentation
- **Editing & Review** - You revise the documentation
- **Delivery** - You deliver the documentation and pray you never have to update it.

Process not projects

Document answers to user questions

Respond to new questions by creating and delivering new documentation

This entire process typically takes from several weeks to several months to complete.

With the current speed of innovation and iteration in software development this approach no longer works. Things change too fast for your organization to wait for you to finish your documentation "project."

Establishing a documentation process is much more effective.



THE IDEAL DOCUMENTATION WORKFLOW

Here is what the workflow looks like when you have a documentation process established in your organization:

- **Very Short Planning Phase.**

You write down basic questions users have and document the answers.

- **Rapid Development**

You create documentation and publish it without going through lengthy revision/approval processes.

- **Incremental Updates**

As new questions come in that aren't answered in the documentation, you author new documents and add them to your online documentation.

This approach involves much less upfront work, produces immediate, tangible results and assures that you are always documenting relevant topics for your users.

You will create more documentation than you have ever authored in your life but it will feel like you are writing less.

Accept the fact that your documentation will never be “done.” Your goal is not to “finish” your documentation. It is to communicate with and help your customers.

When you establish a process, your documentation just becomes a natural, and effective, customer communication tool.

MAKING IT WORK

Now that you know the 5 keys to successful documentation it is time to take what you know and put it into practice. By following this methodology you will be applying the 5 keys without even having to think about them.

Create, Respond, Repeat

To start creating and using your documentation just follow these simple steps:

- 1** *Write down actual questions your users have. Make them as specific as possible.*
- 2** *Create a help article that answers each question. Include as many pictures/screenshots as possible.*
- 3** *Upload your help articles to the web.*
- 4** *Respond with documentation. When your business receives a new question do one of the following:*
- 5** *If the question is already answered in your documentation then respond with a URL that points to the answer.*
- 6** *If the question has not yet been answered in your documentation then create a new help article, publish it and respond with the URL to the new article.*

That's it. By writing down questions you make sure that you are delivering the type of content your users need (key #1) and you are creating it in small chunks (key #2). By adding screenshots and images you keep your instructions clear and unambiguous (key #3).

By publishing to the web and responding with URLs to your customers you make sure that your documentation is connected with Velcro, not cement (key #4). And finally, by implementing this workflow you are establishing an ongoing documentation process that is easily manageable (key #5).



Documentation that gets used can have a major impact on your business.

If you establish a workflow that ensures your documentation is clear and unambiguous, is easily delivered via your customer communication channels and is easily expanded and updated you will ensure that your documentation is **used on a daily basis** by your customers and co-workers.

Once you get your documentation off of the bench and into the game you will be amazed at how it can help your business.



NEXT STEPS

Hopefully this has been helpful in giving you information about the key success factors to software documentation.

Here's a little background on ScreenSteps:

ScreenSteps an online knowledge base and documentation authoring platform. It is used by Small to Medium Businesses, Fortune 500 companies, Universities, K-12 Districts, and Non-Profit Organizations to:

- **Decrease customer support requests**
- **Decrease employee training time while improving results**
- **Decrease call handle times in call centers**
- **Make Change Management and Digital Adoption projects successful**

Our customers have some impressive achievements that have resulted in hundreds of thousands of dollars in savings and increased productivity, including:

- **Decreased employee training time from 60 days down to 15**
- **Decreased average call time by 25%**
- **Increased call center agent efficiency by 46%**
- **Decrease post CRM rollout support training time from 6 weeks down to 1**

Want to learn more? Take the next step below.

[**SCHEDULE A DEMO**](#)

[**GET A FREE TRIAL**](#)