

The 5 Keys to Successful Software Documentation



Foreword

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 sitting on the sidelines in the game of “customer support.” They send everyone else into the game – 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!

Table of Contents

Introduction to the 5 Keys	3
1. Start With the End in Mind	4
2. Author in Small Chunks	6
3. Show, Don't Tell	8
4. Attach Your Documentation With Velcro, Not Cement	10
5. Use Process, Not Projects	13
Making it Work	15
Conclusion	16
About Blue Mango Learning Systems	17

Introduction to the 5 Keys

Is your documentation good enough to help your product get a 5-star review from a major technology publication? Would your customers give your documentation an A+ rating? Do you and your customers use your documentation on daily basis to decrease customer support requests and increase customer satisfaction?

If not then you are probably missing one of the 5 keys to successful documentation. The outcomes mentioned above are real results from real customers. They range from multinationals to school districts to single person start-ups. The size of the team doesn't matter. *What matters is their willingness to change their attitude and approach to 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.

The 5 keys to Successful Software Documentation

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**

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 both common approaches that may seem logical but which will give you disappointing results. Let's work backwards. We want our documentation to be used as often as possible. The more it is used the more time it saves both us and our customers. So let's start with the end in mind.

Documentation isn't read, it's referenced

When is the last time you cuddled up by the fire with a great user manual? Your customers aren't going to do that either. Nobody is going to read your documentation from beginning to end. In most cases people won't even look at your documentation until they are already stuck and confused. When they get stuck they are going to *reference* your documentation.

Thinking of your documentation as a reference tool should change the way you author it. You shouldn't assume that a reader of one article will have read other articles. You shouldn't assume that readers will read your help articles in a particular order. You need to design your documentation to be easily referenced so that your customers can use it to get "unstuck."

Start with the end in mind



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

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 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.

Case Study: Sticky Notes

We used to develop training for 3D/4D ultrasound systems for the three largest medical device manufacturers in the world. They hired us to create training materials for them because nobody was reading their software manuals and users were struggling with the new technology.

We first considered creating several different types of interactive training. But, as we observed the doctors and sonographers, we noticed that they all had multiple sticky notes pasted to their machines. The sticky notes had bullet lists of instructions for key tasks they had to perform.

By looking at how the ultrasound users were *creating their own documentation* we were able to deliver a documentation solution that better met their needs. We took their sticky notes and improved them, creating task-based documentation with pictures.

2. Author in Small Chunks

Long software manuals are painful to write, to read and to 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 your software documentation in a format that will be complete and understandable while at the same time not being overly intimidating to your users?

The key is to author your documentation in small chunks. A simple approach that will help you author in small chunks is to use questions as the title for each help article you produce. The more specific the question the better.

Look at these two examples:

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.

How will authoring in small chunks benefit you? In two ways:

1. 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 just responding to a question.
2. 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

Author in small chunks



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

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.

Case Study: “Just in Time” Documentation

Basekit is a web-based SASS company that simplifies the way the world makes websites. Basekit was one of the winners of Seedcamp 2008 (the European version of Y Combinator).

As a new start up they weren't sure exactly what they needed to document. Gordon Plant, Head of User Experience explains their approach:

“I really got inspired by what Blue Mango said about ‘on demand’ documentation because, other than a few basic things, we realized that we really didn’t know what we wanted to document.”

“What we started doing was every time we got a user question we would see whether we could write a document to answer that question. We’ve now expanded that process. For example, we now have online chats. Very often, in the time it takes a support agent to answer a question online, I will create a new document. We have really taken the documentation on demand thing to heart. We try to get the documents into the system right at the moment they are needed.”

Basekit began by just documenting the questions customers had. This allowed them to create their documentation in small chunks on an “as needed” basis. Authoring in small chunks made the task much less daunting and created documentation that provided instant benefit to their users and their business.

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 literally 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.

You need to use pictures correctly though. Adding a single screenshot to each section of your documentation isn't enough. If you have a bulleted list of instructions then *each bullet should have a picture associated with it.*

Pictures aren't there to enhance the text. The text should be there to enhance the picture.

Show, Don't Tell



- Use a picture for every step of a task you are documenting

Case Study: Show, Don't Tell

Hover, a Tucows company, is a service that provides Domain Registration, Domain Forwarding, DNS and Email services. Their key differentiators are a simple, clean interface, exceptional customer support and clear, visual documentation. They use screenshots for almost all of their documentation. Here is just a small sampling of customer responses to their visual documentation:

- "Great job! Worked perfectly the first time."
- "This step by step made me glad I made the switch to Hover."
- "Great tutorial...very clear, and very effective. Thanks!"
- "Great tutorial ... I waited to the last minute but was able to get it all done in one sitting anyway. Without this guide, would have not been able to figure out, on the fly, the hoops through which to jump and would have ended up having to renew with the GoDaddy another year. Thanks again!"
- "Superb instructions! Simply superb. The new management 'gets it.'"
- "Great instructions. simple. to the point. intuitive. love everything thus far. :)"
- "Great, best instructions ever as said before thanks!"

Example

Look at the example to the right. The same article is presented side by side, one with pictures and one without. Which article would you rather read?

Could you recreate your documentation on a typewriter?

Look at your current documentation. Could you reasonably create the same content using an old IBM electric typewriter? If so then 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.

The image shows two side-by-side screenshots of a web page for 'hover', which provides step-by-step guides and articles. The page title is 'Step By Step Guides and Articles' and the subtitle is 'to follow instructions and videos to help you get the most out of your subscription'. The page features a search bar and a 'Back To Email Tutorials' link. The main content is titled 'iPhone' and includes a 'Download PDF' link. The text describes how to configure an iPhone to work with an email account, specifically mentioning iPhone OS 2.2.1. The left page includes a screenshot of the iPhone home screen with the 'Settings' app icon circled in red. The right page is a text-based version of the same content without images. The text on the right page includes instructions on how to launch Settings, tap the 'Settings' button, and enter the information requested in the password field.

4. Attach Your Documentation With Velcro, Not Cement

Attaching your documentation with cement instead of Velcro is the single worst mistake you can make when delivering your documentation. How do you attach your documentation with cement?

- 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 is attached with cement 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 attached with cement 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 is attached with cement.

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!

Attach with Velcro, not Cement

- Make it easy for your users to "pull" your documentation apart
- Don't deliver large PDF or Word files
- Deliver your documentation on the web
- Make your documentation searchable



Examples

Question:

How do I export a CSV file?

Bad answer #1:

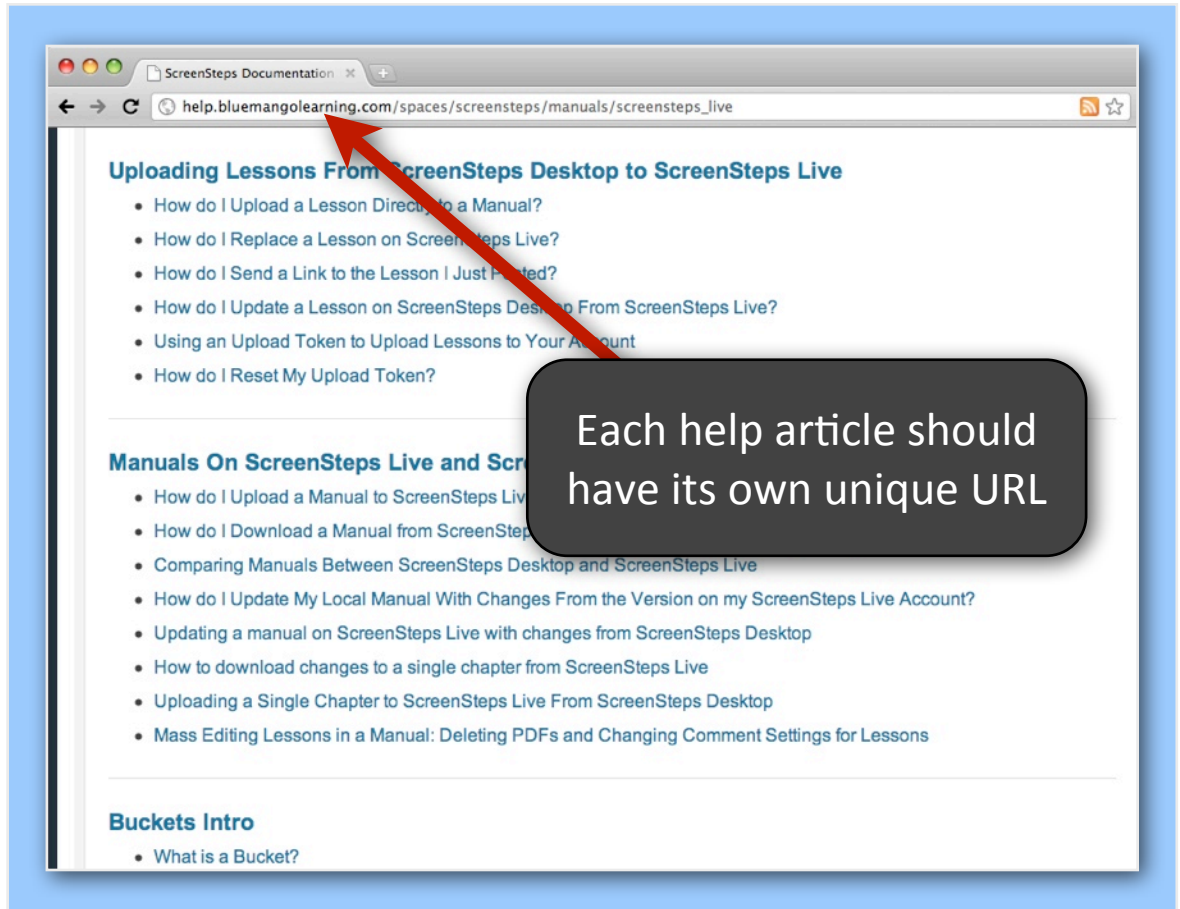
“Open the PDF help file and go to page 64. Look halfway down the page where it says ‘CSV files.’”

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:

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



The screenshot shows a web browser window with the address bar containing the URL help.bluemangolearning.com/spaces/screensteps/manuals/screensteps_live. The page content includes a section titled "Uploading Lessons From ScreenSteps Desktop to ScreenSteps Live" with a list of articles, and another section titled "Manuals On ScreenSteps Live and ScreenSteps Desktop" with a list of articles. A red arrow points from a callout box to the URL in the address bar. The callout box contains the text: "Each help article should have its own unique URL".

Uploading Lessons From ScreenSteps Desktop to ScreenSteps Live

- How do I Upload a Lesson Directly to a Manual?
- How do I Replace a Lesson on ScreenSteps Live?
- How do I Send a Link to the Lesson I Just Posted?
- How do I Update a Lesson on ScreenSteps Desktop From ScreenSteps Live?
- Using an Upload Token to Upload Lessons to Your Account
- How do I Reset My Upload Token?

Manuals On ScreenSteps Live and ScreenSteps Desktop

- How do I Upload a Manual to ScreenSteps Live?
- How do I Download a Manual from ScreenSteps Live?
- Comparing Manuals Between ScreenSteps Desktop and ScreenSteps Live
- How do I Update My Local Manual With Changes From the Version on my ScreenSteps Live Account?
- Updating a manual on ScreenSteps Live with changes from ScreenSteps Desktop
- How to download changes to a single chapter from ScreenSteps Live
- Uploading a Single Chapter to ScreenSteps Live From ScreenSteps Desktop
- Mass Editing Lessons in a Manual: Deleting PDFs and Changing Comment Settings for Lessons

Buckets Intro

- What is a Bucket?

Each help article should have its own unique URL

Choose the right tool

There are some very popular (and very expensive) help authoring tools that will create web help that *only has a single URL for the entire manual*. When you click on different help topics the URL in the browser address bar doesn't ever change. If you are using such a tool then it is time to dump it and move to something else. There is no way for your support agents to pull that content apart and deliver exactly what your users need, where they need it.

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

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

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.

Case Study: Attach with Velcro

Second Street Media Solutions provides white-label software solutions to media organizations ranging from the Los Angeles Times to the Washington Post. They had extensive documentation for their products but it was locked up in a 100+ page PDF file. Their documentation was virtually useless in support situation communications.

Todd Gilbert, head of customer support switched their delivery method to a web based format where each article had its own unique URL. He immediately started sending out links to the answers his customers needed; answers that were in his documentation.

The results? Customer support requests dropped significantly.

5. Use Process, Not Projects

Does your organization do documentation projects? Documentation projects involve four phases:

- Planning – you decide what to document
- Doing – you write the documentation
- Editing/review – you revise the documentation
- Delivery – you deliver the documentation and pray that you never have to update it again.

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.

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

- **Very short planning phase.** You write down some 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.

Process not projects



- Document answers to user questions
- Respond to new questions by creating and delivering new 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. You will create documentation a little bit at a time as you need it.

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.

Case Study: Establishing a Documentation Process

Michael Morrison is the Director of Technology Services for Saddleback Valley Unified School District in Southern California. His team is in charge of supporting thousands of students, teachers and staff.

Michael instituted a policy that their team would no longer answer questions via email. They would either point the user to an article in the documentation or create new documentation to answer the user's question.

The results? Michael explains in his own words:

"We are now creating more documentation on a regular basis. We find we are spending less time teaching people how to do things and more time working to improve the District. We not only feel more organized we ARE more organized."

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. The good news is that we have a methodology that is very simple and effective. 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:
 - a. If the question is already answered in your documentation then respond with a URL that points to the answer.
 - b. 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).

Conclusion

Documentation is only useless if it doesn't get used. 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

Thanks for Reading

Thanks so much for taking the time to read our eBook. We hope that you have found the information useful. If it's all right with you we would like to take a moment to let you know about the software we develop. We create a suite of ScreenSteps products that help everyone from individual consultants to Fortune 500 companies create beautiful, clear documentation and use it in their customer communications.

ScreenSteps Desktop (create)

[ScreenSteps Desktop](#) is our authoring tool that integrates screen capture and document authoring. It is specifically designed to help you create task-based documentation with screenshots and pictures. Think of it as a word processor for pictures. Just capture images and ScreenSteps assembles the document for you in the background. ScreenSteps includes tools for annotating images, adding text and easily updating content when your documentation needs to change.

ScreenSteps Desktop can publish to a wide variety of formats including blogs, wikis, content management systems as well as PDF, Word and HTML. It is available for both Mac and Windows.

ScreenSteps Workgroup (collaborate)

[ScreenSteps Workgroup](#) adds important collaboration features to ScreenSteps Desktop for people who need to work on a documentation team. All source files are stored in a central location on your internal network. Authors can check articles in and out for authoring and can publish to any of our supported formats.

ScreenSteps Live (deliver and collaborate)

[ScreenSteps Live](#) adds cloud-based publishing to ScreenSteps Desktop. You can instantly create a searchable online manual. ScreenSteps Live then gives you tools that make it simple to push your documentation into all of your customer communication channels whether it be through email, support tickets, support chat, support forums or social media.

About Blue Mango Learning Systems

Blue Mango Learning Systems creates customer communication software that:

- Helps businesses communicate more clearly with screenshots and pictures
- Helps businesses communicate more quickly by streamlining and simplifying complicated workflows
- Provides businesses with tools to leverage their documentation in all of their customer communication channels
- Shows businesses how to establish workflows and procedures that make creating and updating documentation simple and fun

Blue Mango Learning Systems is based in McLean, VA and can be found at BlueMangoLearning.com

Twitter: <http://twitter.com/screensteps>

Facebook: <http://facebook.com/ScreenSteps>

Blog: <http://bluemangolearning.com/blog>

Feedback?

Have any feedback on this eBook? Please send it to info@bluemangolearning.com.