

Software help writing just got easy



www.drexplain.com

WRITING COST-EFFECTIVE DOCUMENTATION FOR SOFTWARE SYSTEMS

WHITE PAPERS

... All software
needs
documentation ...

All software, from simple command line utilities to large scale corporate systems, needs documentation. It may be done either in the form of several text strings shown in a console window or in the form of thousands of web pages stored on the company portal in a distributed database. No matter how documentation is done, it must be done.

Nevertheless there are many situations when you have an application but there is no help file with it, and you have no time to write complete documentation yourself. At the same time you have no budget to hire a professional technical writer who can do this tedious work for you. Let's consider the most common of such situations.

Software that needs the cost-effective documentation

The in-house software systems

You likely have applications that you developed specially for your company and use inside your organization only. Frequently, such systems have almost no documentation because nobody thought about it during the process of development. There was an urgent need to solve a critical business task, not to create complete product with full-bodied documentation. However, even in such systems the documentation is necessary because new employees or team members will have to learn the system before they can start to use it too. The documentation is an ultimate source for gathering knowledge about the program and its use: key functions, main modules, best practices, use cases, software issues, etc. Hence, even in-house software must be documented and the cost-effective approach will help you.

Pilot projects and mock-up demos

Test projects are supposed to be a subject for frequent modifications and improvements. Nobody wants to document a program that will likely be reworked completely. Although the probability of big changes in pilot systems is high the application must have at least a basic documentation. There are many people who may want to have the system documentation readily at hand: customers, co-developers, project managers, testers and other engaged people. It's another good chance to apply the cost-effective documenting tactics.

Non-profit, educational and research projects

This group of software systems is similar both to in-house and pilot projects. Research systems are also used in a narrow environment, are focused on their internal algorithms and may be frequently modified by developers. Along with these factors, research projects are on a very limited budget formed by grants, prizes and donations. Every cent is spent on experiments, equipment and development. There are no free resources for software documentation writing. This is another case where the cost-effective techniques may be applied.

Undocumented custom software from third party vendors

For some of your business tasks you have to use third party utilities that may have no documentation at all. The developers may not support the programs anymore or you simply cannot contact their support team. Nevertheless, you cannot stop using these programs because they solve your everyday problems. Keeping all the knowledge regarding these programs only in your mind is a great risk for your business. Why not write simple and cost-effective documentation for these orphan utilities?

Software by small independent vendors (ISV)

Small software vendors and independent developers in the startup phase often profess bootstrapping philosophy by keeping their expenses as low as possible. They literally do everything themselves: market research, programming, art and web design, product promotion and marketing, user support, and of course documentation writing for their software. To survive and to grow, you have to apply the cost-effective approach in every aspect of your business, and in software documentation writing in particular. There is no question of whether to write or not to write - you won't sell your products without decent documentation. The cost-effective technique is the only option under these conditions.

How to write cost-effective documentation for your software system

First, cost-effective documentation is not poor documentation. Cost-effective documentation is decent documentation, which costs less and can be done faster. Let's examine the factors that make the software documentation cost-effective.

Write it yourself. Don't hire professional technical writer

The documentation written by a professional technical writer may cost several hundred or even several thousand dollars. Technical writers can work wonders with your documentation but if your project has limited budget then hiring a technical writer is not for you. Take it easy but don't get upset. If you have basic writing skills, then you can write it yourself. Moreover, you know your program better than anybody else and you can proceed to writing immediately without learning the program features. Self-made software documentation saves you money and helps you meet product release deadlines.

Keep design plain and use templates and patterns

Another way to significantly cut cost and reduce the writing time is by using predefined design templates from your help pages. You may grab them from the Web or from another application, but beware of copyright and license violations. If you use a specialized help writing tool then it most likely includes a set of design templates with different color themes, fonts, design elements, etc. If it has no templates then throw it away and use another tool – you have no time for font and color matching. You have to work fast.

Include basic graphics only, mainly screenshots

One of the most time consuming tasks is creating graphics and illustrations for your documentation. It requires a certain experience in image processing to make your pictures look clear and apposite. The cost-effective approach requires your documentation to have only essential images. Forget about the marketing artworks and photos, useless diagrams, paradigm schemas, complex hierarchy charts, and everything that doesn't directly deal with the software use. The only pictures that users need are screenshots with explanatory labels and captions. To reduce the screenshot-making efforts, use help authoring software that has built-in screenshot capturing and processing tool. Automated capturing, processing, and labeling of screenshots will save hours of your time.

Use simple language with instructions

Software documentation is not fiction. Use simple language to tell users what they must do to complete their tasks. Write nothing undue. Don't waste your time for inventing marketing slogans and sophisticated metaphors. It's better to spend the time on more important tasks.

Cut the secondary content

To save your time, reduce the number of secondary pages in your documentation. This may be the company overview, some legal

information, concepts and paradigms, fundamentals, and other topics that aren't urgent for users. That information won't immediately help people to solve their problems and therefore may be easily omitted.

Choose the right software for documentation writing

One of the most critical aspects of cost-effective documentation writing is choosing a right software tool. There are a lot of documentation programs out there. They are also called "help authoring tools", i.e. HAT. To make your software documentation writing process cost-effective, your tool must comply with the following key criteria:

1. *Reasonable price.*

Although there is a lot of free software for documentation writing, this is not a good choice even for the cost-effective approach. Freeware often means a limited number of features, unpredictable support level, and a misty future for the product. Developing a good help authoring tool is a rather expensive process and requires significant investments. That is why good documentation tools costs more than the average end-user software costs. At the same time, purchasing an expensive software package for \$500 or for \$900 may be an unjustified expense. You can always find a documentation tool that meets your needs in the price range of \$100-\$200. When you're ordering a software tool for cost-effective documentation writing you must choose only those features that you really need.

2. *Ability to quickly document all elements of your software.*

To document your application with minimal effort, the chosen documentation tool should offer you some kind of automation. If you write documentation for an application with a source code then you must use a tool that can automatically parse and document your source code and create API references, class hierarchy, workflow charts, etc. If your application has a sophisticated user interface with lots of windows, forms, grids, tables, and other controls then choose a tool that automates the capturing of screenshots. A documentation tool that can analyze your GUI elements and create completed help pages for them, along with screenshot pictures, descriptions and references will literally triple your productivity. You can focus solely on writing, not wasting your valuable time on picture taking and editing, creating callouts, and sorting out dozens of visual effects. The program will do this for you by using predefined patterns and algorithms.

3. *Ability to easily emphasize key features and functions of your software.*

When choosing a documentation tool, you must check if it has specific features that would be helpful for documenting your application. For instance, if you wish to work a lot with charts and diagrams, then the built-in charting component in the documentation tool would be a great benefit for you. Additionally, if you are planning to work a lot with screenshots, then you should look for the tool which has the capacity to easily add callouts and explanatory labels to your screens. These types of specialized tools are very useful for cost-effective documentation writing.

4. *Various output formats from a single source file*

A good documentation tool must be able to compile your documentation project into various help file formats from a single source. No matter whether you need to create a single redistributable help file or a set of HTML pages to setup on-line help system on your corporate sever, you must be able to do this from a single source file without any extra conversion or additional tools. Using a single-source documentation tool will really save you time and money.

5. *Design templates with minimum of bells and whistles yet clear and elegant*

To make your documentation writing work more efficient, you must use predefined design templates offered by your documentation tool. You shouldn't waste your time by creating a brand new design for your help system. Most of users won't appreciate it much anyway. Just choose one of the given design templates and slightly modify the color scheme and corporate identities. You'll receive a clear and stylish looking document without extra effort. The documentation tools that offer such features will save another couple of hours of work for you.

Summary

As you see there are many situations when you need to write software documentation in a short period of time and without budget. The situations come up fairly often and the cost-effective approach is best. Let's summarize the tips above in order to emphasize the key principles of cost-effective documentation writing:

- | Don't hire famous expensive writers.
Write your software documentation yourself.
- | Don't waste your limited resources on needless bells and whistles.
Keep the design plain and the language simple.
- | Don't use superfluous and overpriced tools.
Choose a reasonably priced documentation tool that meets your precise needs.
- | Reduce the dull and time consuming tasks.
Automate everything you can.

The only function of cost-effective software documentation is to help users solve their problems while working with the program. So, put aside the minor tasks and focus solely on this function.

Solution

Software help writing just got easy



www.drexplain.com

If you need to write cost-effective documentation for your Windows software application then the Dr.Explain tool is your solution.

This program promptly creates interactive screenshots with references, exact and distinct menus and navigation links.

Dr.Explain gives fast help authoring due to automatic window documenting and screenshot taking. Now there is no need for additional screen capture tools and all the data are kept in a single portable source file that makes multiple output formats from a single source possible.

Dr.Explain contains predefined visual templates that enable users to customize the appearance quickly, and allow the program to be easily plugged into a web site or applied to a corporate style.

The latest version of the program also offers sub menus (standard Win32) auto-capturing, command line generation of HTML, CHM, RTF, project validation and compacting tools, mouse-over pop-up windows (Java Script) to show the descriptions associated with controls and navigations through hyperlinks in preview mode. Dr. Explain is also capable of producing print versions of pages for on-line help.

Dr. Explain has all the components necessary to make a professional user-friendly help file in one package and, most importantly, sells at an affordable price of only \$125.

This price offers a unique proposition on the market of software documentation tools. Dr.Explain will automate your software documentation writing process and will save you and your company many hours of labor. The product will easily pay for itself on the first project.

The free trial version of Dr.Explain is available at

www.drexplain.com