Software Documentation: What is it, Its types, Importance, and Best Practices

In today's advanced age of the internet, every organization is dependent on the internet to connect with its audience and potential customers. Organizations enable public access to their websites, interactive web forms, relevant documents, and information through the internet. In short, they utilize the Web Content management systems that paved the way for 'Knowledge Management System' and the 'Knowledge Base System.'

Let us get into details about the above systems.

Knowledge Management System vs. Knowledge Base System

Both these systems are very similar because they are essentially content repositories that allow users to store, query, and make appropriate decisions. But there is a minute difference in both these systems. The existence of a knowledge management system precedes the internet. But it was only with the growing internet usage and subsequent easy access of information to users that it became popular and served as a distributed database over the internet.

A knowledge management system allows you to 'just' store the information in the database. This information is generally in the form of policies, procedures, manuals, software documentation, code, best practices. Etc.

A knowledge-based system goes a step ahead and allows you to appropriately classify and categorize the information into groups, sections, and sub-sections, leading to a much broader and more organized database.

Software Documentation

Every software comes with documentation whose purpose is to resolve the issues faced by its users. These users comprise the developer, support team, customers, and end-users.

The documentation's primary goals include

- Helping the end-user understand the software better
- Helping the customers find the information they require
- Helping the support team find the information they need.
- Resolving the issues faced by developers during the development process.

¹⁰Forbidden

You don't have permission to access this resource.

Additionally, a 403 Forbidden error was encountered while trying to use an ErrorDocument to handle the request.

The documentation must contain easily understandable details and descriptions related to the software to meet these goals.

Types of Software Documentation

Different types of software documentation are created for different types of audiences. Let us have a look at some of these types.

1. User Documentation

The user documentation is made for the end-users using the product. It is specifically created to solve their queries related to a task and help them complete those tasks by referring to the user documentation.

There are many types of user documentation. Here are some popular types.

- **How-to Guides**: These guides have a precise format and help the user complete a pre-determined task
- **Tutorials**: A series of steps listed in this document helps the user learn a concept related to the software
- **Reference Documents**: These documents have details of every technical aspect of the product like the software design details, software specification, etc
- **Configuration Guide**: All the configuration parameters listed in the guide help the administrator install the application successfully
- **Administration Guide**: An administrator can refer to this guide after installing the application. It primarily guides the admin for troubleshooting and managing the software.

2. Developer Documentation

The developer documentation is system-related.

- **Release Notes**: It includes the details about the latest version of the software and what bugs were fixed in the update. It is usually a .txt file.
- **System Documentation**: It includes the design details, UML diagrams, system requirements, etc.
- **README**: It includes a high-level overview of the software alongside the source code. It is also in .txt format usually.
- **API Documentation**: This documentation has the specifications related to invoking API calls and classes, including API in the code that is being developed, etc.

3. Just-in-time Documentation

This documentation aims to serve the support for customer-facing documentation quickly. The users need not refer to any FAQs and technical documents for information.

Importance of Software Documentation

Why provide software documentation when you have easy-to-use software? Yes, one might think of this. But it is important to note that your software will be used by people of different technical capabilities, including laymen. They might be from different ethnicities, countries, organizations, and households.

Good software documentation is necessary to the success of your software. Software documentation helps understand the product, its interface, capability, and ability to accomplish a task. The most important utility of the documentation is that it can be used to search for a query within the entire document, jump to that section and resolve the query as and when encountered while using the product.

It is also crucial for every organization to have documentation tools across their development team or everyone part of the

Forbidden

You don't have permission to access this resource.

Additionally, a 403 Forbidden error was encountered while trying to use an ErrorDocument to handle the request.

Choosing the Software Documentation Tool

It can be time-consuming and tedious to create documentation that consists of all the technical details and, at the same time, is easily understandable by the team members, end-users, and customers.

By now, you know everything about software documentation, except the tool you can use to create this software documentation. Choosing the **right software documentation tool** is extremely important to make your documentation understandable, accessible, and flexible.

Ensure that your documentation tool provides basic features like a custom dashboard, secure hosting, file management, backup and restore, and security. Other than these features, you can choose the best software documentation tool with the following tips.

1. Opt for a multilingual knowledge base

For an organization catering to multiple countries, states, and cities, it is recommended that they use a multilingual knowledge base that allows them to write the documentation in multiple languages based on their customer base. The localization feature available in a knowledge base system can efficiently enable them to perform this.

2. Opt for a secure knowledge base with a backup and restore feature

What can be more important than the data in your knowledge base? It is vital to keep it secure at every point and time. Choose a knowledge base that has advanced security features and access permissions. It should also have a backup and restoring facility that backs up the data at scheduled times and can restore it if need be.

It should also have the option to 'Create a new Backup' to allow manual backups. This feature comes in handy to instantly create a backup when the document is modified extensively or significant changes are done.

3. Opt for a knowledge base that provides integration abilities

Choose a knowledge base with built-in machine translation and artificial intelligence. If not, you can always go for a knowledge base that integrates third parties. This way, you can integrate your knowledge base with modern tools and applications that help you with translation, artificial intelligence, and other capabilities.

4. Opt for a knowledge base that helps segment data

Using a knowledge base that helps you create a public-facing URL is crucial, as that is the only way your users can access your knowledge base. Ideally, your knowledge base should allow you to customize your data and segment it separately for the public, employees, support staff, and external stakeholders.

5. Opt for an interactive and user-friendly knowledge base

The knowledge base is a reflection of your organization. The software documentation can be tough to read and understand, given its technical nature. But if the knowledge base is user-friendly, pleasing to the eye, and easily navigable, it makes it easy for the users to access your documentation. Thus, choose a knowledge base that provides an interactive user experience, detailed information architecture, and flexible features.

Creating a Software Documentation: Best Practices

Undoubtedly, creating software documentation is tedious and can get exhausting at times. But when aligned with the best practices by talented resources, it can work around faster and more efficiently.

Forbidden

You don't have permission to access this resource.

Additionally, a 403 Forbidden error was encountered while trying to use an ErrorDocument to handle the request.

Besides utilizing an excellent knowledge base, you can use the following best practices to create efficient software documentation.

1. Improve and update the documents in the knowledge base

As opposed to general thinking, documentation is an iterative process. It cannot be made once-and-for-all. It has to be consistently improved based on new developments, customer feedback, or needed edits.

One way to do this is by creating and consistently updating the Customer FAQ section to add solutions to the customers' queries. It can also provide reference to the sections that provide alternate solutions and help increase the productivity and efficiency of the product.

2. Use the customer feedback loop to update information

As mentioned above, customer feedback is crucial to update the documentation. But it is not necessary that the customers put in their feedback for you. It would help if you took it upon yourself to collect their feedback. You can collect feedback by using a web form, adding a comment section, or asking them to review the content on a specific topic or page.

Also, before the product's launch, organizations invite beta testers to use their product and take their reviews. This feedback is also crucial and should be updated in the documentation for better utility.

3. Interacting with the Subject-Matter Experts (SMEs)

The subject matter experts and the software developers have an in-depth knowledge of the product. The technical writer might not be very well acquainted with the same. This brings to the forefront a crucial requirement – the SMEs and the writers to collaborate regularly to understand each other and acquire knowledge about the product and its details to create successful user documentation. Other people involved in the collaboration can be the entire documentation team, engineers, document reviewers, and support staff.

4. Use the Agile Methodology for documentation

Despite being the traditional method, the waterfall method has been replaced with the Agile and DevOps methodology. Most companies have adopted the same because adding changes to the existing product design with the waterfall technique is challenging.

Agile methodology focuses on modular development, allowing new changes to be implemented even during development. This methodology is preferred in the documentation development life cycle and the software development life cycle.

5. Create a Style Guide for consistency through the document

The first page of the document should have the same style, look and feel as the last page of the document. How can you ensure this? Simple.

Use a template with a pre-determined style. It would be best to define a style sheet that applies to different document levels and structures. Some standard software style guides like the Microsoft Style Guide or Chicago Style Guide are available on the internet that you can refer to create one of your own.

6. Create SEO rich knowledge base content

What is the purpose of creating user guides if the users can't reach them? The software, documentation, and content should be indexable by the search engines. It should have the relevant meta-tags and meta-titles for this purpose. Ensure the documentation is linked from the application and website as well.

Conclusion

403 Forbidden

You don't have permission to access this resource.

Additionally, a 403 Forbidden error was encountered while trying to use an ErrorDocument to handle the request.

Other than being specific, relevant, and concise, software documentation needs to be developed from the perspective of everyone involved. An efficient documentation tool like a knowledge base is crucial to the success of the documentation.

PHPKB is among the best and most affordable knowledge bases that provide outstanding capabilities to organizations looking to create, modify and expand their software documentation. The features and abilities of PHPKB make the software documentation attractive, understandable, and accessible to the organization's end-users and other stakeholders.

The need for good software documentation cannot be undermined. Keeping the programmers, testers, end-users, and support staff's thoughts, goals, and vision in sync is vital for the overall success of the software and the documentation.

Online URL: https://www.phpkb.com/kb/article/software-documentation-what-is-it-its-types-importance-and-best-practices-287.html

Forbidden

You don't have permission to access this resource.

Additionally, a 403 Forbidden error was encountered while trying to use an ErrorDocument to handle the request.