

# Some possible requirements for documentation tools

---

I wrote this list a few years ago when my then employer was considering replacing FrameMaker + RoboHelp. We evaluated and compared many alternatives including Confluence, MindTouch, and PonyDocs.

## Definitions

**document:** In this context, a set of topics that are distributed as a distinct group, for example, an admin guide for version 2.3.0, release notes for 3.0.2, or a knowledge base. The same document could be distributed in multiple formats, for example, as a set of wiki pages and as a PDF.

**wiki:** a website that allows the creation and editing of any number of interlinked web pages via a web browser using markup language or a WYSIWYG editor

## Finding information (navigation and search)

- context-sensitive help
- table of contents (navigable tree) for each document
- breadcrumbs: page history / hierarchy
- “smart” global (including knowledge base etc.) search with the ability to choose product and version
- hyperlinks within system (cross-references)
- hyperlinks to resources outside of system
- automatically generated list of links to related pages
- elegant handling of 404 errors for bad links and missing pages
- master table of contents for all documents
- authored index

## Customer feedback (“community”) features

- customer comments (monitored by help system administrator)
- private customer feedback (typical implementation sends email to documentation mailing list including the page URL; sometimes combined with page rating)
- page rating
- page watch / subscription: email notification / RSS
- private customer comments (notes to self)

## Availability

- support for mirroring content on multiple servers (for geographic diversity and high availability)
- PDF download
- ePub
- local help installed with software
- HTML output option that requires no web resources (for environments with no internet access)

## **Wiki collaboration**

- edit existing page
- edit page segment
- integrated spell-checker
- embed graphics and multimedia in page
- interactive contents (e.g., plug in numbers to get out sizing info, for example by using HTML forms and JavaScript)
- elegant handling of edit conflicts (when two people edit the same page, what happens when the second saves?)
- track changes with markup
- add new page
- page history with user name and timestamp
- compare page versions (essential for translation)
- compare whole document
- ability to upload files to library and link to them
- virus-checker for uploads
- support for internal discussions, such as comments with restricted visibility
- page ownership (more policy than feature; could implement with page watch / subscription)
- informative email / RSS notifications (shouldn't have to go to page and compare versions)
- incentive features, e.g. points, karma, reputation, badges, top contributors
- file attachments
- ability to restrict file attachments

## **Reports**

- bad links
- orphan pages
- 404 errors
- top pages
- top search keywords
- aging content

## **Writing and editing**

- WYSIWYG editor with table support and good copy and paste from Word
- table of contents editor (faster and less error-prone than manual coding)
- Word import
- PDF import
- DocBook import
- LaTeX import
- source code editor (if required) with pretty printing and tag highlighting
- variables

- snippets
- conditional text
- profiling

## **Web doc server access control and security**

- LDAP authentication
- Salesforce authentication
- long session timeout for external users (e.g. 30 days)
- group-based privileges
- read-only access for customers only to appropriate docs
- document-level read-only permission by group (e.g. beta users)
- document-level edit permission by group (e.g. internal users, trusted customers, application partners)
- moderation
- page-level permissions

## **Version and language support**

- display multiple release versions of each document (e.g. 2.3.0 admin guide, 2.3.1 AG)
- display correct version for context-sensitive help
- display multiple language versions (e.g. 2.3.1 English admin guide, 2.3.1 Japanese AG)
- display correct language based on user preferences
- release tags
- update the same page across multiple versions of a document (topic reuse)

## **Automation**

- easy maintenance of context-sensitive help when page names change, etc.
- automatically add new pages to specified location in table of contents
- internal links updated automatically when page names change
- verify code on save (e.g. flag orphan tags and links to nonexistent pages)
- wiki page templates: at a minimum, CSS (band-style)
- PDF page templates: at a minimum, CSS (band-style or page layout)
- document-level integrated versioning, or integrate with CVS / Subversion, with branching and ability to deploy multiple versions to multiple wiki directories / namespaces

## **Additional requirements to be determined**

- knowledge base
- SEN-drafts / SEN
- Stack Exchange-style system such as OSQA Answers