

1 Create a checklist for your documentation needs

As with any tool, evaluate what it is about the existing system that leaves the team unsatisfied. What are the obvious areas for improvement of your documentation system? Generate a list of criteria that would create the perfect tool for editing, storing, accessing, and tracking the project's documentation.

3 Invest in your wiki

Nothing is perfect off the shelf and your wiki will be no different. Most wiki packages are designed to be customized and extended, so take advantage of that by investing in a few days of developer time to modify it to suit your needs. Re-evaluate wiki usage at regular intervals to gather feedback on the most common gripes and dedicate resources to adjusting the wiki accordingly. The few hours or days spent tweaking it will come back tenfold in time saved by users.

2 Choose the right wiki

There are dozens of different wikis, from free open source models to commercial enterprise packages. Use your list of criteria to research the available options and evaluate each to see how it fits your team. Plan for the scope of the wiki, will your studio need to host tens of thousands of pages or hundreds of thousands? How many users will access it simultaneously? Will the server hosting it be local or remote? All of these factors will affect the speed at which users can work, and faster is always better. www.wikimatrix.org is an excellent comparison tool.

4 Create a structure

While Wikipedia might look anarchic, it has a well defined structure that is moderated by thousands of volunteers. Your wiki will need its own structure, based on your needs identified in Tip 1. Make sure the first page users see presents them with the most pertinent info, based on their account. Wikis often support a variety of structures, including hierarchies, which are often easiest to follow for docs relating to a single project. Users build habits quickly; presenting them with a well defined structure right off the bat can create best practices organically. www.wikipatterns.com outlines many common structural issues.

5 Train users in stages

Start with a 5 minute tutorial on how to create an account, find a page, and edit a page. This will allow users to become familiar with using the wiki to find the information they want and help them to begin contributing. Introduce the Talk/Discussion pages (if your team has decided to use them) and then more advanced features as needed, like history and revisions. The barrier to entry on a wiki is remarkably low, and with just an account and basic editing features users can add to the project. Staged training makes it simple for users to grow into the various wiki features while contributing along the way.

6 Give each project its own namespace or area

When storing multiple projects' (or departments') documentation on the same wiki, use the namespace functionality to clearly delineate the area of each. Doing so will make searching easier and sorting among multiple projects far more effective. While most wikis support namespaces, some also have project areas which group relevant pages even more explicitly.

7 Make it publishable

Depending on your project/publisher/studio model, it's likely that you will have to publish documentation for external review. With a little planning a more traditional design document can be automatically generated from the wiki. Identify the pages which contain the information crucial to your audience and flag them with a template warning ("Page for external review!"). Team members will know to keep those pages up to date and clear of rough notes. Use page transclusion to generate a single page containing all core design documentation and export it out to a PDF or similar format.

8 Link to external resources only when necessary

Many new wiki users will create a page and place only a link to .doc or .txt files stored in the source control system. Ensure that users are aware that text and image documentation is best stored on the wiki so that history is easily tracked. Linking to spreadsheets with complex formulae is unavoidable, so document the purpose of the file and any notes about it on a wiki page prior to linking to it.

10 RSS is your friend

Most wikis support RSS and other notifications for page updates that broadcast when changes are made. These alerts make it a snap for users to be informed when activity important to them happens. An e-mail alert assigned to the pages for publishing externally keeps the lead designer and producer abreast of any changes to information regularly broadcast outside of the team. Team members assigned to a particular level know when the level design is updated, programmers can track build note updates easily, etc.

10 Tips for a Successful Wiki

9 Use tags and search

A good structure makes it easy to find a page, but a large number of pages can still take time to sort through. Good search functionality saves enormous amounts of time, and tagging pages make them more flexible within your structure. A designer can tag a page with relevant terms, such as "design", "system", and "combat". Doing so allows a user to filter content by tag, helping them find the pages they are looking for more quickly.