

Using Wikis in Software Development

Panagiotis Louridas

In only a few years, wikis have become one of the most popular tool shells imaginable based on open source software. You can find them just about everywhere that demands effective collaboration and knowledge sharing at a low budget. Wikipedia has certainly enhanced their popularity, but they also have a place in intranet-based applications such as defect tracking, requirements management, test-case management, and project portals. Panagiotis Louridas describes wiki essentials and nicely distinguishes a variety of types. You'll find something useful, I'm sure.

—Christof Ebert

By 1995, Ward Cunningham, a prominent member of the design patterns community, was experimenting with a hypertext environment to promote the exchange of the community's ideas. The environment was based on the then-popular Apple HyperCard application. That application worked well for individual users, but Cunningham was looking for a Web-based implementation. On 25 March 1995, he posted the Web's first-ever wiki, calling it "the simplest online database that could possibly work."



Wiki means "quick" in Hawaiian. To describe the technology powering his Web site, Cunningham chose the name WikiWikiWeb, primarily because he found it more interesting than saying "quick Web." Also, Hawaiian words are doubled for emphasis, and Cunningham wanted to stress that his program was very quick. Moreover, as we will see, wikis use doublings as formatting markers. Finally, he wanted an unusual word for an unusual technology. The fact is, both the technology and the word have stuck with us ever since, creating new forms of collaboration and knowledge sharing on the Web and empowering many people to contribute to online content.

What is a wiki?

A wiki is actually two things:

- a program that makes it exceptionally easy for anybody to edit Web pages and
- a philosophy regarding how users should go about that editing.

Generally, Web pages are written in HTML. While it's not difficult to learn HTML, non-programmers tend to shy away from it. Even programmers find it easier to edit text than structured HTML, where linking between pages requires attention to things such as addresses and anchors. When many people edit the same Web site, you must also pay attention to how these people collaborate.

In a wiki, users write simple text following a small number of conventions. The system creates the necessary links automatically, so Web page editors have much less to worry about. Wikis, however, move beyond offering practical help to adopting a particular philosophy on who can edit material and under what circumstances. Specifically, just about any system user can edit a page written by any other user.

This might seem like a recipe for chaos, but it turns out not to be—primarily because edits are visible to all. So if somebody abuses his or her editing rights, another person will edit the

inappropriate edit. We can witness how such a system works in practice by visiting Wikipedia (<http://en.wikipedia.org>), whose English version counts 965,042 articles as I write this—probably more than a million by the time you read it. The vast majority of articles are of very high quality.

There are many ways to implement a wiki, so we now have a multitude of wiki engines at our disposal—more than 200 at this time. All follow the same fundamental principles, but they differ in both the underlying technology and the assumptions about how they'll be used. Here, plurality is strength, and somebody wishing to create a wiki is really spoiled for choice.

Editing a wiki

Figure 1 shows the current WikiWikiWeb front page (<http://c2.com/cgi/wiki>). Apart from its content, the page includes a set of links for editing it. (Figure 1 doesn't show the links; they're at the bottom of the page. Also, depending on the wiki engine, the editing links might look like tabs, and the terminology might differ from what I use here.)

Clicking on the edit link opens the wiki page's source text. To edit the text, we write in a human language that follows a few wiki conventions. For example, to end a paragraph, we use a blank line (double enter). We use doubled single-quotes for emphasis (usually italics) and tripled single-quotes for strong emphasis (usually bold). Putting an asterisk in a line's first column creates a bulleted item in a list; more asterisks create deeper indenting. Some wikis offer editing tools like spell checkers and previewers that make life even easier.

Figure 2 shows the edit text for the page in figure 1.

After we're done editing, we click on the save link (or another mechanism with similar wording). This makes our changes permanent, and all users of the wiki will see our edited material from now on. This being a wiki, they can also edit it themselves. This leads to a live document, something between a mailing list and a Web site.

Of course, no Web page is worth

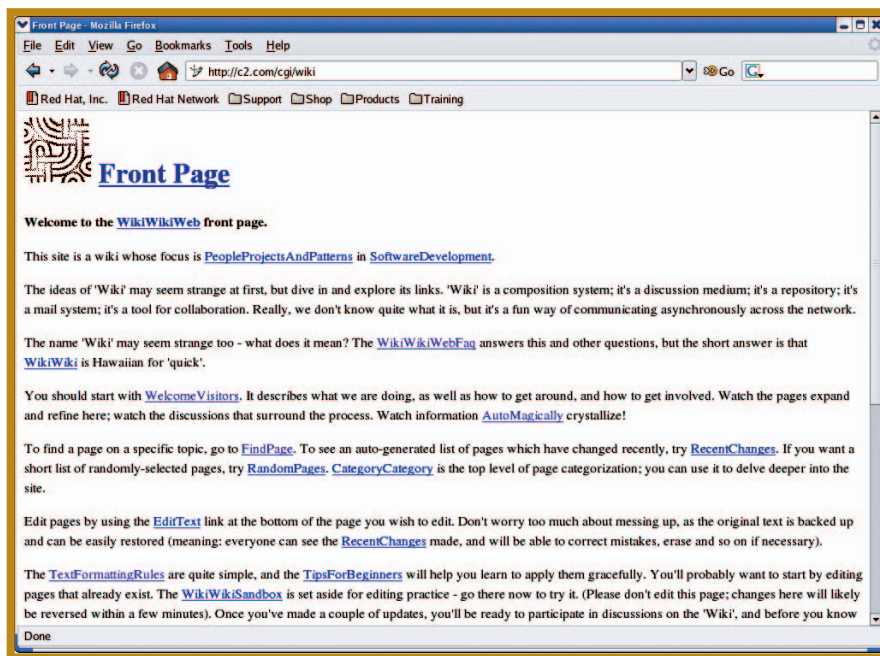


Figure 1. WikiWikiWeb front page.

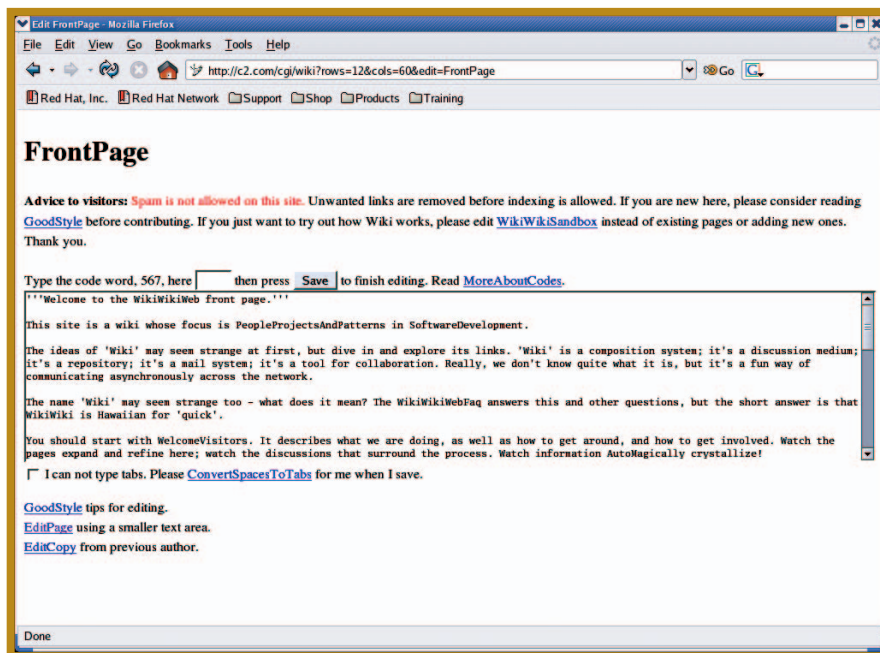


Figure 2. WikiWikiWeb edit page.

much by itself. It must be linked to other pages. When we edit our text in the wiki, a joined capitalized word (LikeThisOne) is a link to a page of that name. Again, the convention changes depending on the wiki engine; for example, it can be the name of the page surrounded by double square brackets ([[Like This One]]). When we save a page that uses a convention, it auto-

matically creates a link to the named page. If the page doesn't already exist, we can click on the link and the wiki engine will take us to a blank page to fill in.

Wikis are very well documented. In fact, they're usually documented in their own wikis. Learning the full conventions for formatting and editing is simply a matter of going through the wiki

Additional Resources on Wikis

The Wiki Way: Quick Collaboration on the Web, by Bo Leuf and Ward Cunningham (Addison-Wesley Longman, 2001), is the primary reference on wikis, cowritten by the creator of the original wiki.

Wiki: Web Collaboration, by Anja Ebersbach, Markus Glaser, and Richard Heigl (Springer, 2005), describes the wiki philosophy, presents the main features, demonstrates MediaWiki and TWiki, and shows how to use wikis as project management tools.

Ward Cunningham's Artima interview (www.artima.com/intv/wiki.html) is an engaging firsthand account of the ideas behind wikis.

The Wiki Design Principles are available in WikiWikiWeb at <http://c2.com/cgi/wiki?WikiDesignPrinciples>. WikiWikiWeb also includes an annotated list of more than 230 wiki engines (at the time of this writing): <http://c2.com/cgi/wiki?WikiEngines>.

A site comparing different wiki engines, from which I gleaned the material in table 1, is available at www.wikimatrix.org.

The 2005 International Symposium on Wikis, which took place 16–18 October 2005 in San Diego, included presentations on various uses and aspects of wikis. The proceedings and discussions are available at www.wikisym.org.

input from many participants, who might not agree. Setting meeting agendas and keeping minutes are good examples. Many projects use wikis for these tasks. In fact, you can use the same wiki for both these tasks—first to set the agenda, then to capture the minutes. This is much more practical than the usual procedure of circulating e-mails soliciting input and comments and then trying to merge everybody's views.

Moving one step further in functionality, some platforms are available that build on wikis and offer many additional content management and project management goodies. The goodies range from RSS feeds and blogs to surveys, polls, and issue trackers.

Rolling out a wiki

For those who insist on doing everything by hand, you can roll out a custom-made wiki that fits a project's requirements exactly. Given that a wiki is a Web-based application, you'll just have to write some server code to accept and record users' edit requests.

The original WikiWikiWeb engine was a CGI application written in Perl. Today wikis are written in anything from Java to Ruby, Perl, PHP, ASP, and so on. At the minimalistic end of the spectrum, it's possible to create a wiki using only Unix shell commands and tools. There's probably not much reason to do that, however, as a Bourne shell wiki is already available, written in `/bin/sh` and plain `awk`. The wiki is called `wikish` and is available at <http://wikish.do.homeunix.org>.

Most developers opt to use one of the popular wiki engines, such as MediaWiki (www.mediawiki.org) or TWiki (<http://twiki.org>). TWiki is a structured wiki, meaning that it aims at combining the free-wheeling wiki attitude with a database application's benefits. It allows easy reporting and the creation of workflows and access control. Pages can include forms, so you can use a TWiki-based wiki for issue tracking or for building other wiki applications. While MediaWiki lacks such features, it's becoming more and more popular as a project cooperation platform be-

help pages. To assist novices, wikis usually come with a sandbox—a place in the wiki where edits aren't permanent, so everybody can experiment at their ease.

Using wikis in software development

No single correct way exists for using a wiki in software development (nor in any other activity, for that matter). Wikis are especially useful in distributed projects: many teams around the world use them to organize, track, and publish their work. Their flexibility frees a project manager from fretting about getting everything exactly right from the beginning. A wiki can and should change to respond to the project's needs as they arise.

While wikis are always easy to change, wiki engines usually incorporate comprehensive versioning and change control for their content. Anyone is welcome to edit pages, but the wiki keeps all changes in a history file. In this way, everybody can check what has changed, who changed it, and when. This is a powerful disincentive for online vandalism and abuse. It's also possible to revert to earlier versions of wiki pages.

At the simplest level, you can use wikis as a project documentation repository. You can upload all documentation to a wiki and link source code to it. This is straightforward if the developer team uses a version control system that publishes code to the Web (see "Version Control," *IEEE Software*, Jan./Feb. 2006, pp. 104–107). Using a wiki as a static documentation repository, however, is missing too much of what it has to offer. It's a lot better to use a wiki to produce, rather than merely store, a project's documentation. In this way, everybody from the project team can contribute, and updates are easy.

Wikis are especially useful as discussion media. It's true that development discussions can occur in mailing lists, as they currently do in umpteen projects around the globe. A wiki offers complementary advantages. It places each discussion item in its context, which can be easier to grasp than it is in a conventional mail thread. Moreover, newcomers usually find wikis more user-friendly than mail archives. Because they're basically a free-form medium, wikis work well in exploratory situations.

Part of the drudgery of project management is the tasks that require

Table 1

Wiki engine comparison

Feature	Wiki engines					
	wikish	MoinMoin	MediaWiki	TWiki	UseMod	Confluence
Programming language	sh, awk	Python	PHP	Perl	Perl	Java
Page preview	No	Yes	Yes	Yes	Yes	Yes
Page revisions	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Access control lists	No	Yes	No	Yes	No	Yes
Math formulas	No	Plug-in	Yes	Plug-in	No	No
Syntax highlighting	No	Yes	No	Plug-in	No	Yes
Tables	Yes	Yes	Yes	Yes	Yes	Yes
Email notifications	No	Yes	Yes	Yes	Yes	Yes
Conflict management	None	Detection, locking	Detection	Detection, locking	Detection	Detection
WYSIWYG	No	No	No	Plug-in	No	Yes
Search	No	Full text	Full text	Full text	Full text	Full text
PDF export	No	No	No	Plug-in	No	Yes
Data storage	Text files	Text files	MySQL	Text files	Text files	HSQL (included), MySQL, PostgreSQL, Oracle

cause it's simple to use, familiar (thanks to its use in Wikipedia), and attractive to many users in its look and feel—an important factor.

Confluence is a proprietary wiki-based platform from Atlassian (www.atlassian.com/software/confluence), which is also available for free for open source projects. Confluence structures content in different spaces, allowing multiple linked wikis. It offers space and page-level security restrictions. It can search in different content, such as pages, blogs, MS Word and MS Excel files, and PDF documents. It integrates well with Atlassian's Jira issue-tracking system and with email. A commercial license for a team of 25 costs about US\$1,200.

Table 1 compares some features of some representative wiki engines.

More important than the particular wiki implementation, however, is being sure that a wiki really fits in the culture of the project or organization. A wiki gives a voice to everybody in the project, delegates control, and promotes cooperation based on trust that people will do the right thing. It requires tolerance and openness. If these don't hold, a wiki might not

fit the situation. Then again, it might make sense to actually nurture these attributes. ☺

Panagiotis Louridas is a grid software engineer at the Greek Research and Technology Network and a researcher at the Eltron Software Engineering and Security research group of the Athens University of Economics and Business. Contact him at louridas@grnet.ueab.gr.



IEEE
distributed systems
Expert-authored articles and resources **ONLINE**

IEEE Distributed Systems Online brings you peer-reviewed articles, detailed tutorials, expert-managed topic areas, and diverse departments covering the latest news and developments in this fast-growing field.

Log on for **free access** to such topic areas as

**Grid Computing • Mobile & Pervasive
Cluster Computing • Security • Peer-to-Peer
and More!**

To receive monthly updates, email
dsonline@computer.org

<http://dsonline.computer.org>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.