

YaleUniversityITS

This case study is a work in progress. Please let me know if you have any suggestions.

Note: Structure inspired by Typo3's [UMR Case Study](#).

Introduction

This document will outline the reasons why [Yale Student Computing](#) chose TikiWiki as a knowledge management system, what requirements TikiWiki had to meet, what advantages TikiWiki offers over our previous content/knowledge management system (or lack thereof), and what future possibilities TikiWiki brings to the table.

Student Computing Services Profile

Part of Yale's Information Technology Services department, Academic Media & Technology's Student Computing group enables students to make effective use of computing technology during their time at Yale. Student Computing Services, with a staff of over 70 undergraduate and graduate Computing Assistants (CAs), serve as general computing and networking consultants as well as liaisons between the student body and Yale ITS and Technology and Planning.

Team Profile

Several students were hired over the summer to overhaul or supplement the Computing Assistant internal website, known as CAWeb. The team consisted of one person with some experience with various open-source CMS systems, and a handful of individuals working on content-creation and organization. Ultimately, we hoped to involve all 70+ student computing staff members in the content creation process.

The Project

CAWeb served as the central portal for all things relating to Student Computing and Computing Assistants (CAs). It was a gigantic collection of static HTML webpages, constantly expanding over the years, with no clear structure or organization. Coupled with the fact that Student Computing has a complete staff turn-over once every few years as the student employees graduate, CAWeb became filled with content that nobody could access or manage, with hundreds of broken links, orphaned and out-dated pages. CAWeb needed to be overhauled, or at least supplemented by a separate website so that it no longer has to serve as the student computing content repository.

We decided that that an easy to maintain content management system was needed to document common computer problems, post University regulations and policies, computing guides, and basically any kind of documentation relating to Student Computing Services. We were looking for more than general content management; we wanted a knowledge management system for our specific type of content.

The Knowledge Management System: TikiWiki

For our project, TikiWiki was the perfect choice. The following is a list of features/properties that are especially useful and attractive in our case.

Wiki

The Wiki is an extremely powerful, efficient, simple, and easy-to-organize collaborative writing tool, perfect for group documentation efforts like ours. The internal wiki referencing system is flexible, allowing us to change our structure, add and modify content with ease (as opposed to more rigid knowledgebases or static websites). File attachments allow us to upload and associate files with relevant documentation pages. Perhaps the most important of all is the wiki history feature. All changes are recorded, and all wiki pages can be rolled back. These safeguards make collective writing a much simpler and reliable process.

CMS/Portal Functionality

TikiWiki also includes features that we have come to expect from other popular open-source CMS/Portal systems. The Article/Review publishing feature can be used to post news and announcements. The FAQ serves as a quick-and-dirty place to find answers to frequent problems and questions. The File Gallery can be used to store frequently used files like printable PDFs and virus removers. The Links Directory stores essential links to external pages. There are dozens of other useful TikiWiki features that we may or may not make use of as the site is finalized.

Granular Permission Control

TikiWiki provides a very detailed permission system. We envisioned at least 4 levels of users. Administrators have full control over every aspect of the site. Moderators have full control over content of the website. Privileged users have permission to create content in some areas of the site. Basic users are only allowed to view certain types of content. (Future: category based permissions?)

Rich Core Functionality

Arguably, no other open-source CMS system is as feature-rich straight out of the box. While many other CMS systems can be extended using plugins and addons to provide similar functionality, these non-core modules are not maintained by the official development team. Unofficial modules are often unprofessionally maintained, abandoned midway, or just buggy. This is not the case with TikiWiki.

Extendibility

Adding modules to TikiWiki is simple. We added two modules which accessed the database of our support ticket system, [Request Tracker](#), listing unclaimed tickets in the user's queue or tickets owned by the user. We will also implement a module that tells the user if he has not filed his payform yet.

Project Challenges

Interfacing with Central Authentication Service (CAS)

[Central Authentication Service](#) is a Web Initial Sign-on (WebISO) system designed by Yale ITS. CAS facilitates

single sign-on across multiple web applications and provides these web services with the ability to authenticate users without having access to their passwords. From an end-user point of view, all protected pages show a standardized CAS challenge page where the user types in their NetID (a unique username of sorts assigned to everyone affiliated with Yale) and password.

Much to our delight, we were able to make TikiWiki interface with CAS without any customization. Yale ITS provides `mod_cas`, an Apache module that protect webpages through CAS. Since `mod_cas` is an Apache module, it behaves like standard HTTP authentication, storing the NetID in `$_SERVER['REMOTE_USER']`. TikiWiki supports HTTP authentication; when a user is logged in through HTTP authentication, and the username matches one of the usernames in the TikiWiki database, TikiWiki automatically logs the user in. That way, when a user logs in through CAS, TikiWiki matches the NetID (username) of the user with a pre-created account in its database, and logs the user in.

Mid-summer, we decided to publish a public frontend for students and faculty, allowing them to view TikiWiki without logging in. This rules out `mod_cas`, as HTTP Authentication does not allow for anonymous access of protected pages. Therefore, we had to modify the login code in TikiWiki to authenticate with CAS while allowing anonymous users. Amazingly, only one file (`tiki-login.php`) needed to be modified. As you can see, TikiWiki is powerful and yet easy to modify!

Convincing Others that Wiki is Easy

The first reaction that my colleagues had towards TikiWiki was that creating pages is too complicated. Most were expecting a more traditional knowledgebase with rigid categories where once can simply click add and type in the text for a new page. The concept of first creating a new reference from an existing page and then filling in the content for the newly referenced page was quite foreign. However, after people understood the ease of wiki syntax, coupled with TikiWiki shortcuts like double-clicking to edit a wiki page, content templates, auto-page-creations, auto-renaming, etc., the Wiki concept became very well received.

Some Buggy Features

We also spent considerable time working out kinks in the system, trying to reproduce bugs and filing bug reports for the TikiWiki developers. Due to our need for some cutting-edge features, we had to run the CVS branch of TikiWiki, as opposed to the release/stable branch. No doubt some of the bugs arose from constantly merging in code from CVS and our own custom modifications. Nevertheless, the TikiWiki developers are very friendly and responsive, making bug tracking and reporting a relatively painless process.

Immediate Benefits

Cost

Aside from TikiWiki being a free (as in free beer) product, TikiWiki also saved us perhaps tens and hundreds of man-hours to create a similar system from scratch or modify some other CMS system to suit our needs. With TikiWiki, we were able to set up a large and complex knowledgebase within a month or two.

Rapid Content Creation

We have been able to create dozens of wiki pages within several weeks. The ability for many people to join in the content creation process is invaluable.

Meeting Project Deadlines

Officially, our project had a 3-month deadline. TikiWiki was up and ready to use in much less than a month, and the rest of the time could be spent creating content.

Attracting Users

The attractive features and interface in TikiWiki made it more likely that users would make use of this resource and become more involved.

Multiple Frontends for Different Audiences

We were able to publish multiple frontends, one for the general public (including students and faculty), and one for CAs and ITS staff members. Using TikiWiki's sophisticated permission system, more levels of access or frontends can be added in the future.

The Future

Currently, the knowledgebase is only for Student Computing staff use. The public frontend is a very limited Intro/FAQ wiki page. In the future, we plan on opening the site up to the general public (students and faculty) so they can also make use of the extensive knowledgebase. Hopefully, TikiWiki will be able to assign permissions based on categories. Content will be placed in different categories depending on the target audience.