UserPageVlad

Table of contents

Preamble

About me

I'm an Electrical Engineering undergrad at the University of California, San Diego. My programming experience includes C/C++, Java, and various shell languages. A linux convert, I have experience in the administration of both Windows and Linux, although the later focuses almost entirerly on RHEL4. I have yet to learn any PHP.

What I'm Good For

My strengths come from my dedication, teamwork, positive attitude, and desire to make things better. I have an inherent interest in the quality control of products and am an excellent writer (especially for an engineer!). I also have a lot of experience writing technical documentation.

My TW Experiences

I've been using TW off and on for well over a year now (probably pushing 2-3 years, actually). I was first drawn to it because of its impressive documentation and extensive features list. Development also seemed to be quite rapid. I consider TW one of the best CMS's out there. It has many strengths but several key weaknesses that I'd like to see improved.

What I'd like to do

In the future, I'd like to write and cleanup documentation about TW. Having legible, easy to follow documentation makes life better for EVERYONE. Eventually, it would be nice to have the oppurtunity to give more direct feedback to the developers and perhaps write some of my own (minor) patches. Obviously I need to learn some PHP. ;)

TW Advantages:

Lots of features, only 1 website.

One of the problems I've been having is that my intranet site needs several different features to be effective. Unfortunately, the alternatives I've tried (Drupal, Plone, etc) are extremely management intensive, overly complex, and dependent upon external 3rd party plugins to become "whole" (Plone for example doesn't offer a Wiki, but you can download a plugin and then *try* to get it to work in your site instance). Tikiwiki has just about all the features I need in ONE CMS.

This advantage can't be praised enough; uniformity is a POWERFUL tool as it minimizes the amount of work a user needs to do to achieve his or her goal. Having to manage (and incorporate!) a))MediaWiki((site, blogging software, and additional software for other features is a pain to set up and a nightmare to manage. Being able to use a single syntax in your blog, a wiki page, and a forum post is one of those things that users take for granted and TW does extremely well.

Authentication Mechanisms

There's lots of ways to authenticate your users. And while I haven't tried all of the options TW offers, TW is very flexible when it comes to authenticating users. LDAP and PAM support can really make it easy to integrate into an existing network, while Tiki's own builtin user authentication and web server authentication support allow for most web masters (including myself) to quickly get a site up and running. An important market (the corporate one) would probably refuse to use TW if it weren't for these mechanisms.

Backups, Export Wiki Pages

TW has made it fairly easy to backup the database of your site and dump/export your Wiki pages to a tar file. While there's still some room for improvement (I think being able to import and export a variety of objects directly from the database, such as blog posts, articles, etc would be a powerful and useful utility for admins), TW is at a point now where just by backing up your database you can safeguard a great deal of your site.

Templates

Although I have yet to master them, templates are a must for large sites who replicate similar content often. Tiki has kept templating as a backbone for as long as I can remember, and content creators will find it relatively painless to create templates to suit their needs.

Modules/Menus

I love the modular menu system, even if though its nomenclature totally confused me when I first started using TW. The module admin page is a great sandbox to develop your menus. Lots of builtin modules make it easy for site admins to get what they want quickly (whoever wrote these, I love you!). Tight integration with security makes them an extremely effective means for reaching specific audiences; for example, you can show ads to all anonymous (unauthenticated users), make text only ads for registered (authenticated) users, and have no ads for another group of users!

CATEGORIES!

Taxonomy, especially on the internet, is an important thing. It's all about finding, organizing, and dissenting information. Categories should try to meet all of these goals. It's clear, however, that the rather monolithic method of categorizing via hierarchies leads to rudandancy. When these trees can't overlap, it leads to aliasing. This is where TW stands right now and needs the most improvement, in my opinion, of course. Read below about what I consider disadvantages of TW for my solution to this problem.

TW Disadvantages:

CATEGORIES!

Tagging is all the rage these days, and for good reason. Using tags is a much, much more intuitive way to organize, search, and categorize data. Google makes excellent use of tags as labels in GMail.

Tags are kind of one dimensional however, in that you can't really see how any two tags relate except by seeing objects that do (and don't) share the same tags. My solution to this in another project was a hybrid tree/tag structure. This allows some basic inheritance in case you want "context specific" tagging.

In addition, more objects should able to be categorized; infact, any object should be able to be categorized. Wiki pages, Articles, Forum posts, (single) Question-Answer pairs, etc, should all be taggable. Instead of listing a FAQ in the current way (where you first create a FAQ, then add questions to it), you'd create questions, tag them, and then use a special Group Tag to effectively categorize many questions (dynamically!) into a single FAQ. More on these ideas later.

Rapid development without feature stabilization

Things tend to "most of the time" work. Features seem to be pushed out the door and fixed if broken at a later date. The addition of new features needs to be frozen while the stability and usability of current features is maximized, much like most open source programming is done. Things have gotten a lot better in the 1.9 series, though.

Bizarre configuration system

It can be difficult to turn features on and off. While this is partly due to the nature of the CMS (and not a bad thing), some features, such as calendaring, are confusing when enabling. Things have gotten much better since the early 1.6 days, though. Take a look at the WIKI Admin module to see what I mean when I say "bizarre."

Poor documentation (but getting better!)

The PDF documentation outlining how so much of TW worked is sadly outdated. Good documentation is one of the best selling points of a CMS, and unfortunately, the TW community has let this slide. Help links in the configuration section link to pages that dont exist on the TW site, adding to new user confusion and frustration. This is everyone's fault (including mine). Having help links everywhere (for admins only, in most cases) would greatly educate new users.

Terrible Developer<->Community Feedback

The dev site is great and all but it fails to make the link between the community and the developers. The forums are nice as well, but seem painfully ignored . Developers need to maintain a development blog that keeps the community apprised of what's going on; what the next release will bring, what still needs work, etc. Additional community feedback (can we say comments?) will help the developers do their work more efficiently and meet the desires of the community more effectively. Polls, feature bounties, Bug Days, etc, would all be good ways to encourage communtiy<->developer relations and raise funds! If the dev's decide not to include a commonly asked for feature for an understandable reason (too difficult, whatever), a Tiki Feature FAQ (or dev FAQ, whatever) should be created. Some of these are "in general" and not really problems that TW is facing.

Features I'd Like To See / Changes I'd Like To See

More advanced WYSIWIG Editor

If you've ever used the python based content management system Plone (based on Zope), you'll know what I mean when I say TW's WYSIWIG editor is decent at best. I've had strange formatting problems, such as doble spaces appearing when hard returning, blog text coming up filled with HTML (this should be checked for!), etc. It's not bad, it just needs to get some kinks worked out ;)

Per post categorization of blog entires.

I think that (if allowed) each user should have one "collective" blog. Each user's blog would have global options and per post options. Global options would become per post defaults, for example, when creating the blog, the user could select options such as "By default make my blog entries private (only viewable by me)" or "By default categorize my posts as Community::Feedback" or "By default, make my blog posts searchable if not marked as private".

When reading someones blog, it's desirable to have all entries shown to you by default and THEN select a categorization mechanism. This is how most blogs are designed and I've found that this is the most intuitive (ie, user friendly) methodology. Using multiple blogs each with a different category on a per user basis makes it too difficult for the reader to follow (who wants to click through 7 blogs just to find your latest post?). Of course, you don't have to limit users to one blog each, that's just an opinion 

More Quicktags / Plugins Menu

There should be significantly more Quicktags by default. Some of them could easily be ripped from

popular forum software (bbcode anyone?). Take a look at The Neowin Forums as an example of a site who gives posters a lot of control over a post's content. Also, plugins are an extremely powerful, but slightly hidden, feature for content creators. Having a "+ Plugins..." menu when editing/posting (just like Quicktags) would be really great!

Do one thing, do it well, and THEN do something else

The developers should hear what the community has to say about such features as the forums, wiki, blogs, etc, and then go back and fix them! How about icons for categories? Moving the Home/Welcome page to a "News Blog" that can be posted to according to its permissions.

I'm critical of TW because I love TW. I want to see it become the best CMS out there, as it is one of the ONLY CMSs that has all the features I want in one place. I'd just like to see it implement those features in a better manner than it currently does to make the site easier to navigate, easier to manage, and easier to improve!

Cleaned up Modules (Menus)

This is an important usability feature that will drastically simplify the navigation, search, and usefulness of a TW site. Right now on my site (when not logged in, ie, anonymous) the Wiki folder in the primary Menu looks like this:

Wiki
..Wiki Home
..Last Changes
..Dump
..Rankings
..List Pages
..Orphan Pages

In my opinion, it should simply look like this:

Wiki
..Search

The Wiki link would work as it does now (link to Wiki HomePage). Clicking on the Search link would open a page offering you a variety of options and allowing you to...

- Search wiki page by
  - Title
  - Body
  - Category
  - Author
- Click links such as
  - Last Changes
  - Rankings
  - Orphan Pages
  - Master Page List

Authenticated users should then have a more filled out menu, such as

Wiki
..Search

..My User Page
..My Pages

Where search would be the same as above, My User Page would load))UserPageUsername((, and My Pages would perform a search for wiki pages with Username as the author. Admins could then have the Structures link, a link to the Administration: Wiki page, etc.

I realize that these changes may sound cosmetic, but I'm fairly certain there's some coding that would have to be done ;) If nothing else, it's beyond my ability to implement.

Stuff That Needs Fixin' That I'm Too Lazy To File A Bug Report For

More helpful error messages

Problem

I noticed when trying to send a user message that I recieved an error message notifying me that I must enable user messaging in my user profile (+1 for clarity). It then gave rough directions on how to do this (+1 for mentioning where to go).

Solution

Add a link directly to User Preferences. Or, even better, say something like "You currently have messages disabled in your user preferences. To enable them, click here". Clicking the link would enable messaging in your preferences and send you off to the page you were trying to get to.