WIKIS FOR ENTERPRISES

KNOWLEDGE MANAGEMENT USING THE WORDPRESS FRAMEWORK

BY FLORIAN KROMER

Wikis for Enterprises

Knowledge management using the WordPress framework

Florian Kromer

This book is for sale at http://leanpub.com/wikisforenterprises

This version was published on 2014-11-02

For my Family.

Contents

Foreword

This e-book is intellectual property of the author. If you have not purchased this .pdf file from the e-book website on the lean publishing platform leanpub.com[1] please buy a regular copy of this e-book.

The author has invested a lot of enthusiasm, energy and time to collect, select and structure the information presented.

Thank you very much.

[1] https://leanpub.com/wordpressbasedwikisforenterprises

About the author

I started programming web sites around 10 years ago. My father is self-employed tax consultant. A system administrator worked for my father from time to time. I was fascinated of his work and tried to join him as often as possible. I was lucky that he worked as freelancer that time. I was able to join often in the evenings so.

He recognized my interest and inspired me to program the first website for my mother. A website to promote the holiday flats in the houses of my parents. It was fun and i enjoyed it. First i used the most pragmatical way to build the websites using graphical programs.

My mother studied social pedagogy and is an alternative practitioner running a naturopathy practice. I went on to program also the websites of my parents businesses. There was always the chance to be creative and try new things out.

When i started to work as embedded software developer i began to migrate some websites to plain HTML, CSS and Javascript. I thought it would be a good idea to gain some experience for the programming of embedded web server interfaces. Despite of that it was a lot of fun…

At some point in time i was sick of all my markdown notes about work on my local hard drive. I wanted to ease the access and management of all that collected information. Overcoming the limitations regarding the access of my locally stored information would be another welcomed benefit… i would be able to access the information from everywhere and at every time i would feel like doing so. This point in time was the initial motivation for me to get in touch with the content management framework WordPress.

If you are interested in what i am doing despite of having fun with WordPress and writing this e-book in my spare time feel free to visit my website[2].

[2] http://florian-kromer.de

About this book

The following three sub chapters clarify the context of the book and its intended readers and give an overview over its content. Typographical conventions used in the book are described. In addition there are mentioned people which have contributed to the becoming of the book.

Intended audience

In general this book is not addressed to a specific audience. The book points out the advantages and possibilities of a *CMS*[3] based on *WordPress* as *CMF*[4]. This information can be used to argue the launch of a *WordPress* based *KS*[5] in your organization.

The many examples have great value to people which are going to architect/design/implement/test a *WordPress* based *KS* or are already running one. There is always something to optimize… there are never *100%*[6].

If the assumed background knowledge is not present there are third sources referenced or cross linked. With the help of the huge *WordPress* community in the world wide web it is easy to get hand-on instruction to specific topics.

For using the book as walk-trough one should have at least basic *WordPress* knowledge. Basic PHP knowledge is useful in the the context of customizing navigation and layout. Basic HTML and CSS knowledge is useful to keep on track in the short parts addressing layout and design. If the wiki shall be run in the web instead of within a local webserver basic web hosting knowledge is useful.

[3] *content management system*

[4] *content management framework*

[5] *knowledge management system*

[6] In the technical domain it is common to measure the proceeding in a project in percentages. But proceedings can not be measured against an absolute “aim” but is always relative to the “aim” of the actual point in time.

Structure and content

The structure of the book is based on a typical wiki development process. This book does not raise the claim to be complete.

Background information about principles and the functionality of *WordPress* are explained in the corresponding part of the book. The same is valid for explanations to functionality which is added to the *WordPress core* and *(parent) theme* using e.g. plugins. In general there is no background knowledge required. But there are short explanations and references instead of detailed instructions. The explanations to the general concepts are focused on the “wiki context” whenever possible. The book should be used as quick reference with “cheat sheet” character.

The following topics are NOT in the scope of this book:

- theme development
- plugin development
- widget development
- corporate identity (layout considerations, etc.)
- corporate behavior (external user groups with access rights, addressed user groups, etc.)
- corporate communication (internal/external, email subscribing, etc.)

Typographical conventions

name, etc.

important keyword, etc.

Aside or sidebar

ℹ Information

🔑 Tipp

⚠ Warning

`code within the text`

```
source
code
block
```

> Block quote

Acknowledgement

As a reward for the great design of the cover sheet… a thankful “high five” to a former fellow student Herbert Müller[7].

Thank you René (Wayandt) for inspiring me to dive into IT… brought not only fun during spare time but also at work.

I would like to thank the stackoverflow community[8] for a lot of inspiration regarding *WordPress* in the CMS context.

[7] http://herbertmullerart.tumblr.com/

[8] http://stackoverflow.com/

Samples

Text sample

Why no *WordPress* theme?

To maximize the benefit of the knowledge management framework based on *WordPress* the core functionality has to be extended. Developing a *WordPress (parent) theme* right from the beginning makes no sense. That would take a lot of time and effort which equals high costs. The value for money of professional themes is unbeatable. You cant develop a theme for a price as low as one of your own working hour.

It is better to consider architecture and design, optimize the child theme and switch to another parent theme if applicable (better value for money, means again: including feature which would be too costly to add on your own).

> ℹ My first idea was to develop a *WordPress* child theme in parallel to the book and to deploy it with this eboook. Unfortunately many of the plugins i do recommend are not free. The purchase of the plugins is not worth mentioning for an individual. Also for an individual company because in most cases there is no distinction between private or commercial use. But dependence on the amount of *WordPress* installations the plugin may be installed on ("site license" or "domain license"). The price for a plugin license model including the opportunity to deliver it within your own theme are proportionally higher [e.g. plugin "wpDataTables"]. Some plugins are available with site licenses only [e.g. plugin "WordShell"]. Especially if the license model is a combination of site and time limited as "price per domain and per month/year" (e.g. plugin "WP Content Copy Protection & No Right Click"). The implementation of the plugins functionality on your own is not worth the development costs and development time (and in my case impossible).

The road map

1. Analyze if *WordPress* is the right *CMS* for you.
2. "Design" the *WordPress knowledge management framework* architecture (*theme* + *plugins* + *widgets*)
    1. Identify the quality attributes.
    2. Prioritize the quality attributes.
    3. Analyze the impact of constraints on the possible architecture.
    4. Perform workarounds for possible constraints with impact.
3. Set up the local development and test environment.

4. Install a *WordPress* instance in the local web server.
5. Install and configure the *theme*.
6. Install and configure your personal *child theme*.
7. Install and configure the *plugins*.
8. Install and configure the *widgets*.
9. “Design” the architecture of the website/content structure architecture.
    1. Design *page templates* and create *pages*.
    2. Create *custom post types*, design associated *post templates* and define *taxonomy*. Create *custom posts* and structure them using *taxonomies*.
    3. Define *categories* and *tags* for *posts* and design associated *post templates* specific to single categories.
    4. Set up the content structure using the elements of 1. to 3.
10. Backup the WordPress environment (installation, database, theme, plugins, widgets, etc.).
11. Start use-ability testing in your organization.
12. (optional) Migrate the wiki into the web Assumption: Webhosting/virtual server/root server/server and domain is already present. The setup of that is NOT scope of this book.

Plugin configuration sample

Example plugin: "WP Admin Protection"

Against brute force hacking the admins user name and password. The plugin adds a visual captcha protection on the wp-login.php login site. In addition to that there has to be added a secret word in the URL of the wp-login.php!

Features (must have): - visual captcha protection on wp-login.php login site (against brute force) - forces a secret word in wp-login.php site URL ("secret link") - IP white listing - IP black listing - email notifications about not authorized access attempts

Features (optional): - failed login attempt monitoring - general account activity monitoring - WHOIS functionality (get info about ip addresses which access on wp site) - comment spammer ip monitoring - information about the security status (e.g. per dashboard widget)

1. Install plugin "WP Admin Protection"
2. (optional) Get reCAPTCHA code:
   - visit the Google reCAPTCHA website[9]
   - press button "Get reCHAPTA".
   - press button "Sign up now".
   - domain: `[the URL of the admin login page]`
   - press button "Create".
   - press on site link "Account"
   - (choose the appropriate URL)
   - Public Key: (copy the [private key] for the following plugin configuration)
   - Private Key: (copy the [private key] for the following plugin configuration)
3. Configure the plugin (free version):
   - Secret Link: `ticked` Enable Secret Link
   - Secret Link Word: ([keyword] which has to be entered in addition to the regular wordpress login site URL `../wp-login.php?[keyword]`, free version is limited to `4` symbols)
   - Send notifications:
   - Send for successful login action (pro version only)
   - Send for failed login action (pro version only)
   - White IP List: (allowed IP addresses, one per line)
   - Black IP List: (not allowed IP addresses, one per line)

[9] http://www.google.com/recaptcha/intro/index.html

- Captcha Code (reCAPTCHA): `ticked` Enable reCAPTCHA for the login page
- Public Key: `[paste the copied key from the reCAPTCHA website]`
- Private Key: `[paste the copied key from the reCAPTCHA website]`
- press button "Save Changes"

⚠ If you use this plugin you have to request a public and private key combination whenever you change your URL (server directory of your WP installation) in your rechapta account on the reCHAPTA website. The public key has to be updated in the WP configuration.

1. add new site for new root path of the URL (e.g. website.de WITHOUT following directories e.g. website.de/www/..) in reCHAPTA
2. export the WordPress MySql database to the local file system
3. in the local file: search `'recaptcha_public_key'` and replace the old public key behind with the new one
4. in the local file: search `'recaptcha_private_key'` and replace the old public key behind with the new one
5. delete all old tables in the MySql database
6. import the local file including all database tables into the MySql database

ℹ Alternatively you can open database table "wordpresswp_plgwpap_config" and replace the keys without backing it pu before.