



DevCraft

ProTips for WordPress teams

Eric Marden
<http://xentek.net>

Who I Am

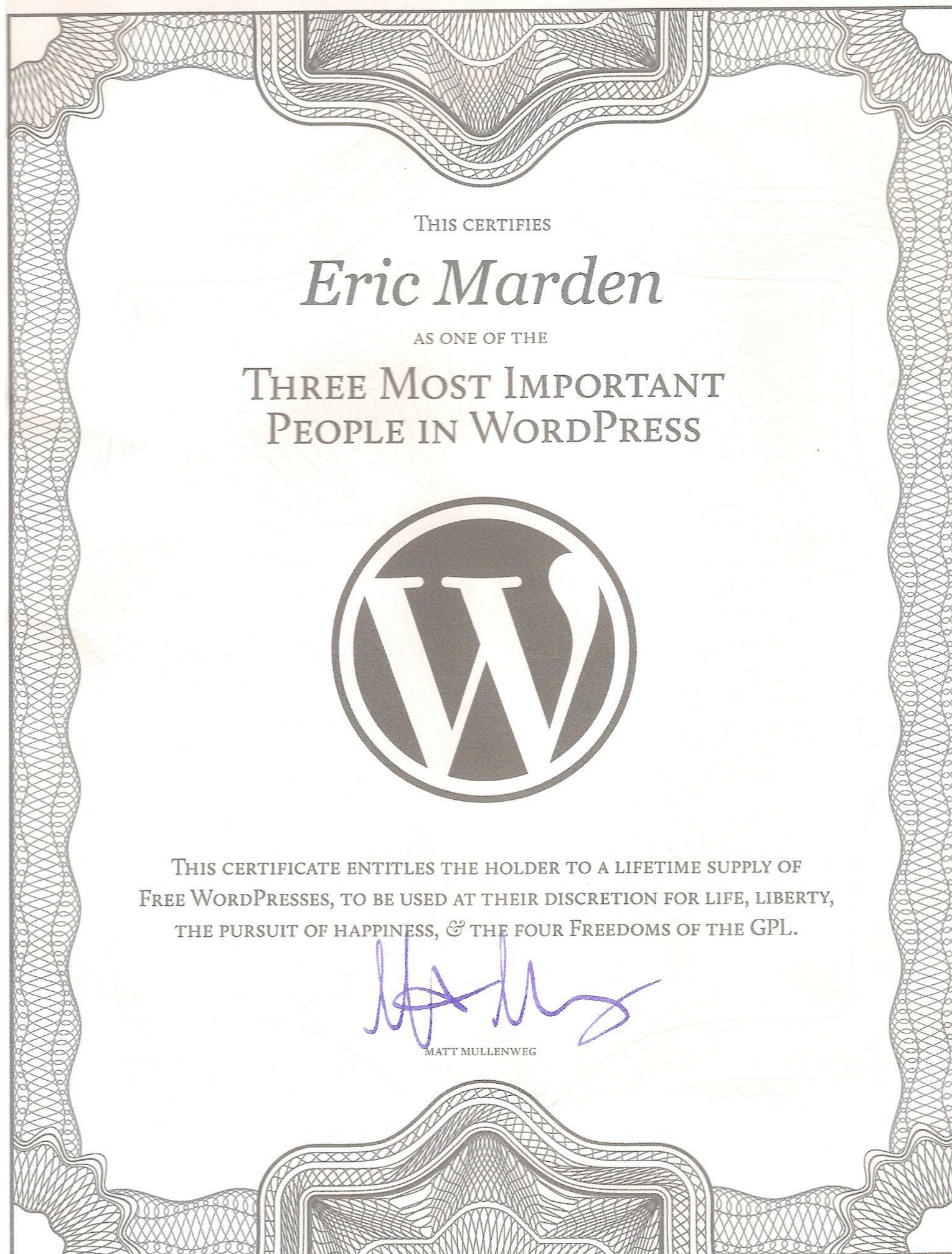
Chief Technology Officer at
Blueprint Design Studio

14 years experience

WordPress Contributor, Plugin
Dev, Theme Author

Polyglot Programmer

One of the three most
important people in WordPress

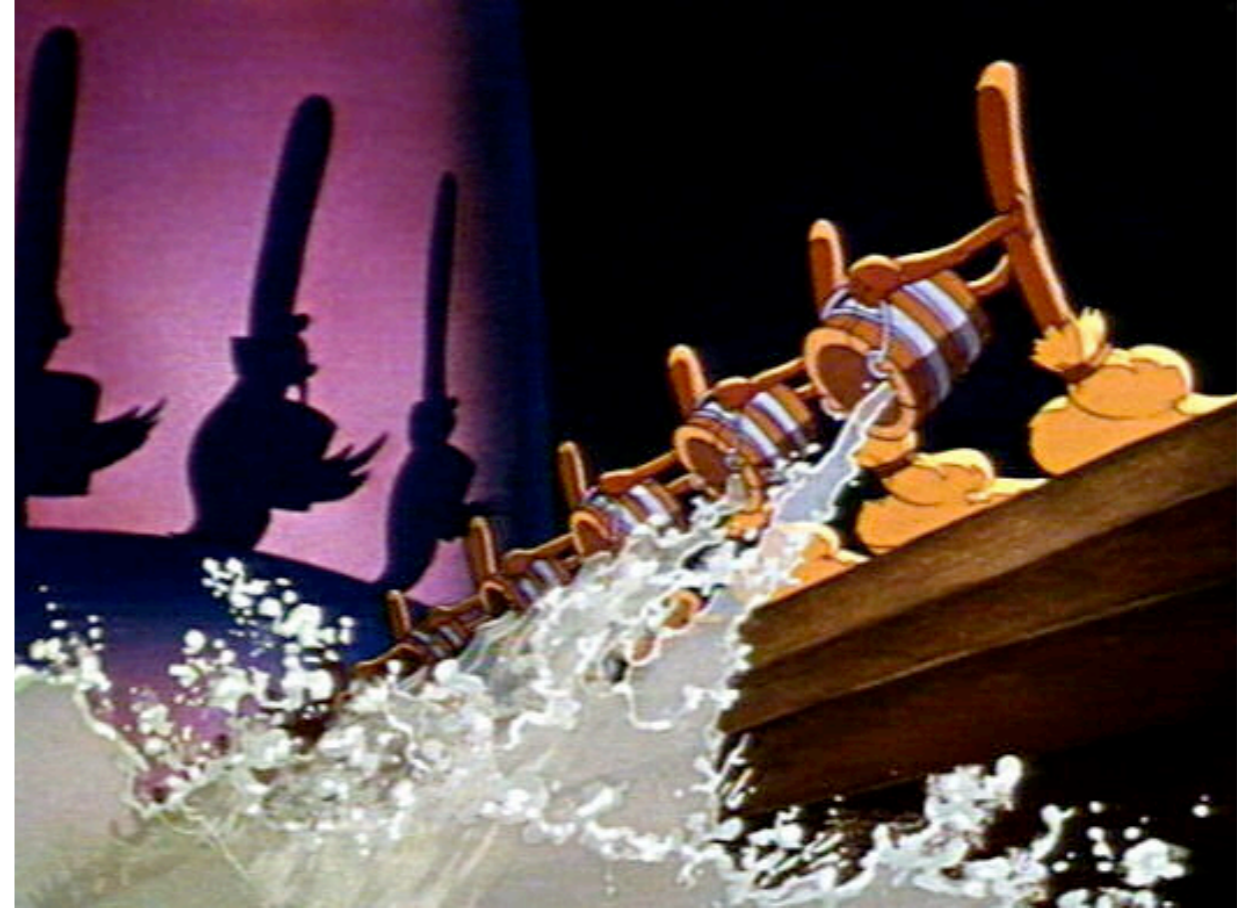


any sufficiently advanced technology is
indistinguishable from magic

Arthur C. Clarke

if your job is to **wield that magic** for your clients

then you need a set of incantations you can count on



or this happens

what you need is a process

a set of rituals designed to create reliability, predictability, and stability into your workflow

the development process is fraught with dangers

the process is your saving throw



components of a mature development process

- **task management**
- source control
- multiple server environments
- configuration management
- data/content synchronization
- continuous integration
- one step deployments
- iterative development
- automated testing

task management

the method by which you describe what you see in your head so that other people know what the hell you're talking about

task management

break down the project into discrete tasks

task management

write your tickets as use cases

task management

every task on a project should be documented in a system you trust

tools:

trac

unfuddle

bugzilla

task management

provides visibility, tracking, and project status

components of a mature development process

- task management
- **source control**
- multiple server environments
- configuration management
- data/content synchronization
- continuous integration
- one step deployments
- iterative development
- automated testing

source control

the means by which you cover your collective development ass and ensure that the team is always on the same page

source control

every project, no matter how small, belongs in a repository.

no exceptions

source control

commit early and often

each time you've created anything of minor consequence on your project,
check it in

source control

commit messages matter. they provide **the why**
the tool already provides **the what** and **the where**

source control

vendor repositories, svn:externals, and other advanced workflows

source control

there is no other tool more crucial to your development process

you owe it to **yourself**, to **your clients**, and **your teammates** to learn your source control system like the back of your hand

components of a mature development process

- task management
- source control
- **multiple server environments**
- configuration management
- data/content synchronization
- continuous integration
- one step deployments
- iterative development
- automated testing

multiple server environments

the strategy you use to keep from blowing shit up

multiple server environments

local

development

staging

production

multiple server environments

local → development → staging → production

multiple server environments

speeds development and tightens the feedback loop

components of a mature development process

- task management
- source control
- multiple server environments
- **configuration management**
- data/content synchronization
- continuous integration
- one step deployments
- iterative development
- automated testing

configuration management

the technique you use to avoid clobbering your server environments and breaking your site

configuration management

problem: each server environment will likely have different file paths, and database requirements

solution: make your wp-config *smarter*

configuration management

1. your server needs to know who it is

```
<VirtualHost *:80>
  DocumentRoot "/Users/xentek/Sites/test"
  ServerName test
  ServerAlias *.test
  SetEnv webenv xentek

  <Directory "/Users/xentek/Sites/test">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

typical apache virtual host configuration, with an environment variable set

configuration management

1. your server needs to know who it is
2. your site needs to know which server its on

configuration management

`apache_getenv('webenv')`

configuration management

1. your server needs to know who it is
2. your site needs to know which server its on
3. your configuration should adapt to this knowledge

```

$webenv = ( function_exists('apache_getenv') && apache_getenv('webenv') ) ? apache_getenv('webenv') : 'production';

switch ($webenv) {
    case 'xentek':
        define('WP_CONTENT_DIR', '/Users/xentek/Sites/sitename/wp-content');
        define('WP_CONTENT_URL', 'http://sitename/wp-content');
        define('WP_PLUGIN_DIR', '/Users/xentek/Sites/sitename/wp-content/plugins');
        define('WP_PLUGIN_URL', 'http://sitename/wp-content/plugins');
        define('WP_SITEURL', 'http://sitename');
        define('WP_HOME', 'http://sitename');
        define('WP_DEBUG', false);
        break;
    case 'dev':
        define('WP_CONTENT_DIR', '/srv/www/sitename.mydevserver.tld/public/wp-content');
        define('WP_CONTENT_URL', 'http://sitename.mydevserver.tld/wp-content');
        define('WP_PLUGIN_DIR', '/srv/www/sitename.mydevserver.tld/public/wp-content/plugins');
        define('WP_PLUGIN_URL', 'http://sitename.mydevserver.tld/wp-content/plugins');
        define('WP_SITEURL', 'http://sitename.mydevserver.tld');
        define('WP_HOME', 'http://sitename.mydevserver.tld');
        define('WP_DEBUG', false);
        break;
    case 'production':
    default:
        define('WP_CONTENT_DIR', '/srv/www/sitename.tld/public/wp-content');
        define('WP_CONTENT_URL', 'http://sitename.tld/wp-content');
        define('WP_PLUGIN_DIR', '/srv/www/sitename.tld/public/wp-content/plugins');
        define('WP_PLUGIN_URL', 'http://sitename.tld/wp-content/plugins');
        define('WP_SITEURL', 'http://sitename.tld');
        define('WP_HOME', 'http://sitename.tld');
        define('WP_DEBUG', false);
        break;
}

if ( WP_DEBUG ):
    ini_set( 'display_errors', 1 );
endif;

```

replace define('WP_DEBUG', false); in standard wp-config.php with this instead

WordPress address (URL)

`http://xentek`

Site address (URL)

`http://xentek`

defining constants overrides what's in the database

configuration management

make it easy to move your site from environment to environment

configuration management

use absolute relative urls where possible and employ built-in variables, functions and options

components of a mature development process

- task management
- source control
- multiple server environments
- configuration management
- **data/content synchronization**
- continuous integration
- one step deployments
- iterative development
- automated testing

data synchronization

the process you use to schlep content from server to server

data synchronization

script it, or use a database tool like navicat

data synchronization

this is a messy problem and requires intimate knowledge of the state of your database

data synchronization

bless one server as the source of truth, and guard it with your life

components of a mature development process

- task management
- source control
- multiple server environments
- configuration management
- data/content synchronization
- **continuous integration**
- one step deployments
- iterative development
- automated testing

continuous integration

the robots you employ to keep your dev server up-to-date

continuous integration

ci software:

cruisecontrol.rb

ci_joe

integrity

hudson

xinc

continuous integration

use a build script to automate tasks

build script tools:

phing

apache ant

components of a mature development process

- task management
- source control
- multiple server environments
- configuration management
- data/content synchronization
- continuous integration
- **one step deployments**
- iterative development
- automated testing

one step deployments

the software you use to keep you from botching your code deployments

one step deployments

ftp is antiquated and insecure and should be avoided

your source control software should be used instead

one step deployments

manual processes will always be error-prone, automate as much possible

reuse your build script and/or use capistrano

one step deployments

make publishing code changes dead simple

components of a mature development process

- task management
- source control
- multiple server environments
- configuration management
- data/content synchronization
- continuous integration
- one step deployments
- **iterative development**
- automated testing

iterative development

a style of building websites in small increments to improve quality and ensure a clean consistent design to your code base

iterative development

be loose , be flexible, be nimble

this is what they mean by 'going agile'

components of a mature development process

- task management
- source control
- multiple server environments
- configuration management
- data/content synchronization
- continuous integration
- one step deployments
- iterative development
- **automated testing**

automated testing

the code you write, before you write code

automated testing

testing tools:

simpletest

phpunit

selenium

automated testing

lowers the cost of making changes and minimizes the risk that new code will break old code

components of a mature development process

- task management
- source control
- multiple server environments
- configuration management
- data/content synchronization
- continuous integration
- one step deployments
- iterative development
- automated testing

getting started

care about your craft

learn your tools

implement in small steps

keep improving. always.

your process is your product

thank you

Eric Marden

eric@xentek.net

<http://xentek.net>

twitter: [@xentek](https://twitter.com/xentek)

