

Adventures in Drupalia

Drupal from an outsider's perspective

Paul Nijjar

KWLUG

September 10, 2012

What is Drupal?

Drupal is...

- A content management system
- A “web distro” of website components
- A thriving FLOSS community



About Khalid



- 25 years in software development and consulting
 - Mainframe, UNIX, Linux, LAMP
- Drupal since 2003, develop, contribute, consult
- Several contributions to Drupal core
 - Site maintenance, hook_watchdog()
- Over 37 modules contributed on drupal.org
 - Userpoints, Nagios, SecondLife, Favorite Nodes, Flag Content, Nodevote, ...





About Khalid



- 25 years in software development and consulting
 - Mainframe, COBOL, C, C++, Java, JavaScript, PHP, Python, Ruby, Linux, LAMP
- Drupal since 2003, develop, contribute, consult
- Several contributions to the Drupal core
 - Site maintenance, hook_watchdog()
- Over 37 modules contributed on drupal.org
 - Everpoints, Nagios, SecondLife, Favorite nodes, Flag Content, Nodevote, ...

My Qualifications (?)

Until last year:

- Did content updates on an intranet site (which I didn't design)
- Did content updates on the KWLUG site (which I didn't design twice)
- Attended local Drupal User Group meetings

And then...

<http://jobsearch.theworkingcentre.org>



Drupal is a popular project!

There are lots of modules

There are lots of eyeballs to keep the code secure

There is lots of information on the INTERNET



Drupal is a popular project!

You will be bombarded with comment and login spambots

You have to keep up with security patches

Industrial Strength



Used by large,
traffic-heavy sites!

<http://www.bmj.com>

<http://www.whitehouse.gov>

<http://ubuntu.com>

<http://ozzy.com>

Others: <http://buytaert.net/tag/drupal-sites>



Those sites all need caching to run effectively!

Your site probably will too!



Drupal can be overkill for small sites (your blog)!

Administration headaches

Not really plug-and-play: it takes work to make a good-looking site

The learning curve is high



There are modules for (almost) everything!

Shopping carts (Ubercart, Drupal Commerce)

Third party embedding/integration (maps, photos...)

Forums and groups

Organizing content into hierarchies

Different menus



Modules are easy to install and use!

<http://drupal.org> is a central repository for almost all modules

drush makes downloading, upgrading and installing modules even easier

Uninstallation is usually painless and mostly clean



Module buffets make your sites fat!

Each module slows down your a site a little and costs memory

Some modules slow down your site a lot (hello `taxonomy_menu`)



Modules add points of failure!

There are no guarantees of quality control

They potentially open up security vulnerabilities

Modules with no upgrade path can get you stuck



The community takes care of some modules that matter!

Widely-used modules often get new maintainers or upgrade development

Lots of eyeballs means lots of patches



drupal.org gives you one stop shopping!

Project pages, issue queues, user contributed patches, workarounds

Access to groups

Access to project creation resources

Enormous amounts of documentation (but check dates)

The community is FRIENDLY and HELPFUL



The learning curve is steep!

Terminology: nodes, taxonomy, entities, hooks, blocks, sites...

Where do you put things?

What modules should you trust?



There is a natural progression from user to developer!

Custom code → custom modules → sandboxed modules → published modules

Module use → running into bugs → applying patches → submitting patches → maintainership

Unified drupal.org accounts make the transition easier



All kind of people become Drupal developers!

People without computer science backgrounds

Lots of geographic and ethnic and gender diversity

Some contributors have never programmed before!



So easy!

Drupal is written in PHP!



Drupal is written in PHP!

So easy to do horrible things!

Some contributors have never programmed before!



Drupal is written in PHP!

Drupal has standards to avoid security holes and bad practices

The standards are easy to apply via the API (`check_plain()`, `t()`)

Integrated patches are usually held to these standards



The community is friendly!

Contributors are kind to each other about fixing mistakes and improving code

Politeness is the rule, not the exception

Take that, facebook and Google!



Nymwars are not an issue!

Important contributors are known by silly handles
("merlinofchaos", "webchick") and nobody cares

People usually associate real names with accounts, but it is not
necessary

People are evaluated based on contributions, not real names



Drupal is free as in freedom!

Proprietary addons are rare/nonexistent (compare: Wordpress)

For now, even “downstream distros” like OpenAtrium and Drupal Gardens are completely free software

Free backend infrastructure helps the community as a whole



People make money using Drupal!

People maintain code because clients need the functionality

The learning curve is steep, so skilled Drupal developers can find work

Customers want unique, branded, robust themes



Certain ecosystems are weak!

The selection of quality polished themes is smaller than Wordpress

It is hard to justify working on core modules or generic backend systems (such as **features**)

When people have “scratched their itch” they can lose motivation to continue maintaining code



There is a Drupally Way!

There are usually ways to override/patch/customize functionality and still do upgrades sanely:

- hooks
- templates
- custom modules



The Drupally Way is not always obvious!

Some best practices are documented

Others you will learn about the hard way



Major versions of Drupal are “legacy-free”!

New major versions may change APIs, interfaces, approaches

New versions of core can pursue best practices (CCK), trendy features (responsive design, web services) and drop functionality that is rarely used (blog posts)

Keeps core Drupal lean and functional



Major versions of Drupal are “legacy-free”!

Major upgrades are often lots of work (essentially redesigns)

Important modules sometimes lag far behind official Drupal releases

Working modules are dropped because porting to newer versions is too hard



Drupal recognizes that major upgrades are hard

There are always two supported versions (currently D6 and D7)

Drupal publishes detailed guides of major changes between versions

Module maintainers make “pledges” to have versions of their code ready for the next version

Late adopters benefit from accumulated upgrade knowledge and sometimes migration helpers



Minor upgrades are usually easy

Especially with **drush**



Maintaining testing and deployment sites is difficult

Drupal keeps both configuration and code in one database

Production needs to keep data intact but change modules/configuration

The **Features** module helps but this is still an open problem

Obligatory Educational Content:

Secrets of the Drupally Way

Don't hack core! Don't hack contrib!

Use hooks and custom modules

If you HAVE to modify contributed code then track those changes carefully

Having different folders for modules (**contrib**, **patched**, **custom**, **features**) can be helpful

Split configuration and data

Configuration: in files controlled with version control:

- Put templates into files, not the database
- Use **features** to extract **views** and config into “custom modules”
- Use version control

Data/content in database:

- Master copy should live in production site
- Pull production copy to staging site when doing upgrades or configuration changes

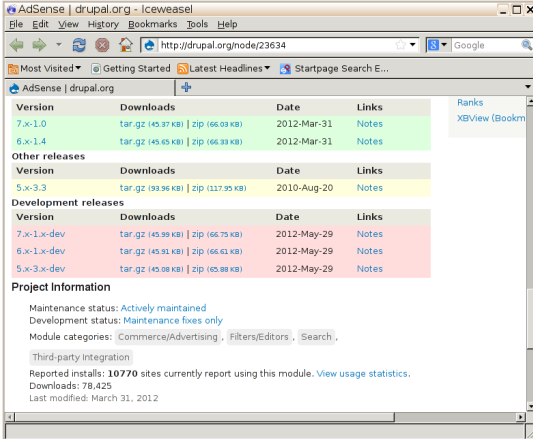
Choose modules carefully

Use popular, well-supported modules when possible.

<https://drupal.org/project/usage>

Find out which direction the community is headed

Look for maintainer interest, upgrade paths, alternative modules that accomplish the same thing:



The screenshot shows a web browser window with the address bar displaying `http://drupal.org/node/23634`. The page title is "AdSense | drupal.org". The browser's address bar includes a search box with the text "Google". The page content is organized into several sections:

- Version** table with columns: Version, Downloads, Date, Links.
- Other releases** table with columns: Version, Downloads, Date, Links.
- Development releases** table with columns: Version, Downloads, Date, Links.
- Project Information** section containing maintenance status, development status, module categories, and third-party integration information.

The "Version" table lists the following data:

Version	Downloads	Date	Links
7.x-1.0	tar.gz (45.37 KB) zip (66.03 KB)	2012-Mar-31	Notes
6.x-1.4	tar.gz (45.65 KB) zip (66.33 KB)	2012-Mar-31	Notes

The "Other releases" table lists the following data:

Version	Downloads	Date	Links
5.x-3.3	tar.gz (93.96 KB) zip (117.95 KB)	2010-Aug-20	Notes

The "Development releases" table lists the following data:

Version	Downloads	Date	Links
7.x-1.x-dev	tar.gz (45.59 KB) zip (66.75 KB)	2012-May-29	Notes
6.x-1.x-dev	tar.gz (45.91 KB) zip (66.61 KB)	2012-May-29	Notes
5.x-3.x-dev	tar.gz (45.08 KB) zip (65.88 KB)	2012-May-29	Notes

The "Project Information" section contains the following text:

- Maintenance status: [Actively maintained](#)
- Development status: [Maintenance fixes only](#)
- Module categories: [Commerce/Advertising](#), [Filters/Editors](#), [Search](#)
- Third-party integration
- Reported installs: **10770** sites currently report using this module. [View usage statistics](#).
- Downloads: 78,425
- Last modified: March 31, 2012

Keep the number of modules you use as low as possible.

```
drush pm-list | grep Enabled | wc -l
```

Sometimes you can replicate the functionality of a module in custom code.

Sometimes you can do without.

Use the community

Go to Drupal User Group meetings: third Thursday of the month, 58 Queen Street South :

<http://groups.drupal.org/waterloo-region>

Become familiar with the drupal.org website

Become familiar with resources published by Drupal contributors and companies (Lullabot, Acquia...)

Consider attending Drupal events (Toronto DrupalCamp)

Use drush

Don't not use **drush**

Easy to learn (even for people uncomfortable with the command line)

Makes administrating certain Drupal tasks much easier

Worry about security

Keep up with security updates

Restrict use of PHP code for end users

Beware of spambots and spamhumans flooding your site

- Mollum
- Skill-testing questions

Worry about Performance

The type of hosting matters: shared hosting vs VPS (Virtual Private Server) vs standalone server

- Shared hosting with mostly anonymous users: **boost**
- VPS/standalone hosting: **memcache**

Khalid has lots of good articles on <http://2bits.com>

Turn off caching for development; turn on for production

More best practices:

<http://drupal.org/node/17557>

Apologies and Thanks

To Khalid Baheyeldin for not suing

To Andrew Cant for obligatory educational content

To the Waterloo Region Drupal User Group

The End

Comments? Questions?

