



Content Management System Comparison

White Paper written by Darren L. Nye,
VP Interactive & I.T.

www.jkdesign.com

JK Design Bespoke CMS:

<http://www.jkdesign.com/services/web-design-nj/content-management-systems>

January 10, 2012

OPEN SOURCE VS COMMERCIAL VS CUSTOM CMS (BESPOKE CMS)

A Summary of the Debate

Utilizing an open source Content Management System (such as Drupal, Joomla or WordPress) versus a commercial off-the-shelf CMS (such as Open Text, Microsoft SharePoint, Ektron's CMS400.net, or EllisLab's Expression Engine) versus a custom CMS solution tailored to your needs, is a never ending debate in Web development circles. It seems that most people are strongly in favor of only one of these options. However it largely depends on your requirements and your budget.

The Open Source CMS

Drupal, Joomla and WordPress are some of the most popular open source CMS solutions available. One of the misconceptions about open source is that it's a "free lunch". However it really is not. Although there are no license fees with an open source CMS, you still need to pay a vendor to create an interface design (aka: "theme"), install the CMS, configure it for use, implement the "theme", and also pay for ongoing support. Alternatively when you pay a vendor for a commercial or custom CMS product, your fee typically pays for an extensive QA process, documentation, and quick and succinct support – because the vendor built the commercial or custom product, and they stand by it. When you choose an open source solution, you will be paying whoever implements it to provide those services for you, often taking much longer for support for an open source system than commercial or custom/bespoke systems. Whomever you are paying to implement an open source CMS, will not be able to "fix" any bugs or security issues that you discover with the actual core CMS – nor will they likely be willing to change functionality of the actual core CMS – either is known as "hacking the core" – a big mistake. If you hack the core, then you cannot use new versions of the core CMS or modules. Any bugs found with the core CMS (or modules), or any perceived lack of functionality you need, would need to wait till the organization responsible for the open source CMS, finds the time to fix your bug or security issue (some may not even be "fixed" as they may not be deemed a serious "bug" or issue), or adds the functionality.

Client Administrative Interface usability, quality of code, speed of execution and enhanced Web server requirements of open Source CMS systems is an ongoing issue. Drupal and Joomla have been developed over the years to attempt to provide every conceivable solution for every conceivable web developer's problems and as a result many feel they are bloated, contain "spaghetti code" that is hard to understand, and are ultimately non-specific platforms that normally require enhanced Web servers and complex customizations for specific client requirements. WordPress is fast becoming another bloated CMS.

These open source CMS packages do have "modules" (aka: "plug-ins" or "widgets") that can be installed to add new functionality that a client may need. However the open source module systems and documentation are typically filled with system specific jargon that means a developer has to be familiar with that open source CMS package. Developers basically have to be specialists for the CMS package in use, not simply good PHP developers. In addition, the functionality of these modules is often overlapping or competing with other modules – and they in turn add even more bloat to the original CMS, in an already unfriendly and clunky interface. The interfaces for these modules are often much different than the core CMS, and other installed modules. There is rarely a common administrative interface between CMS modules. You may not believe or appreciate these comments, until you actually try installing, administering, extending and then upgrading an open source CMS such as Drupal.

When it becomes apparent that a client needs something customized, because an exact Drupal or Joomla module doesn't fit, or an exact module doesn't quite work or has security issues, then building a custom module is required. Creating custom modules for these platforms takes considerable development time and resources (from training all developers on the platform to maintaining and supporting), often with an inelegant solution as the end result. One of the leading Drupal developers, Ber Kessels, has interesting related opinions about the appropriate use of Drupal: (http://www.leftontheinternet.com/message/Drupal_not_good_for_government_websites).

Perhaps a bigger issue with these systems is once your custom modules are created, they will need to be continuously revised by your developers, when new versions of Drupal, Joomla or WordPress are issued. These

CMS systems issue new versions of their own core CMS system quite often – normally to close security holes, fix bugs and add new features. And installing the new versions of Drupal, Joomla and WordPress is required, or your Web site will potentially be easy to hack if security holes are not fixed. Installing a new version of these CMS systems, Drupal in particular, is complex and convoluted, and requires disabling all existing modules prior to an upgrade.

The advantage of having many third party modules to choose from, when using an open source CMS, is also one of their disadvantages. Drupal and Joomla do not evaluate these modules and confirm they meet any sort of security standards. For a client concerned with security, this is a distinct disadvantage of these open source systems. In addition, obtaining support for an existing module, and also receiving an upgrade to an existing module – when a new version of Drupal or Joomla is issued – can sometimes become an impossible feat.

Licensing for an open source CMS can be an issue for certain agencies or clients. For example, Drupal and Joomla use the GPL version 2 license, which means that any custom modules created for a client, must also be licensed as GPL. This means if an agency invests time to write a module for a client, and they distribute it to the client with the GPL version 2 required, the client may then distribute it to the open source community, when in fact the agency may not have wanted this. The same is true in reverse. If a client is paying an agency to write a module, the agency can then distribute it to the open source community, when in fact the client may not have wanted this. This may or may not pose and issue for an agency or client. (<http://drupal.org/licensing/faq/#q7>).

All this being said, many somewhat technical people have developed very popular Web sites running on Drupal, Joomla or WordPress. However due to the shortcomings summarized above, Drupal and Joomla in particular, are two CMS systems that JK Design would probably not recommend to our clients. However for clients that need a simple easy to use CMS, or a blog-only site, WordPress is a system that JK Design would recommend and one we use for some of our very small clients, who are not expected to need custom functionality in the future, that existing plug-ins couldn't provide. We also use WordPress for our own JK Design blog. (<http://www.jkdesign.com/blog>).

JK Design has evaluated many other open source CMS systems including Typo3, TypoLight, ImpressCMS, TikiWiki, CMSMadeSimple, and others. However none of them had more advantages than disadvantages, when compared to a Commercial or Custom CMS.

Commercial CMS Systems

Commercial CMS systems often overcome some of the disadvantages of open source CMS systems. One of the primary strengths of a commercial vendor, is there is always someone you can call when you need help.

Another distinct advantage of a commercial CMS system is that the vendor has control over the entire environment and most modules that are created for it. This often creates a more user friendly and secure end product. In addition, commercial CMS systems normally have new releases of their core CMS to add functionality, and a viable upgrade path for their clients.

Since commercial CMS systems are not open source – the community at large does not have access to the Source Code. As such, possible hacker exploits are much less likely. Granted, simply because a commercial vendor isn't publicizing their source code, doesn't mean they shouldn't extensively test their code and ensure security; many of them do, because they have the funds available to invest in this process. And when there is a security exploit, a commercial vendor will likely resolve and issue/apply fixes quickly. There are usually fewer core CMS security updates from commercial vendors than open source vendors – and as such, updating modules to accommodate a new core CMS version for security updates, happens less frequently.

Some commercial CMS systems are “cleanly coded” and well-documented, with an extensible architecture. This means that a third party developer can read and follow the source code, understand the Application Programming Interface (API) and create custom enhancements to the CMS, without extensive issues.

There are also some commercial CMS systems that are good tools usually for specific purposes. Microsoft SharePoint is a great example. It was designed to quickly create an internal Intranet, especially for clients who are

“Microsoft Shops” – using all Microsoft tools. However SharePoint is not appropriate for implementing public Internet sites, as the tools are geared toward Intranets, and customizing SharePoint for Internet applications can be a very expensive and inflexible process. In addition, expensive license fees for SharePoint, is another disadvantage.

Other commercial CMS systems are hugely complex and priced extremely high. An example of a high-end vendor would be Open Text, who formerly marketed “RedDot CMS” and who also purchased Vignette. They now market a new Web Content Management system as part of their Web Solutions group. Licensing alone for a typical Open Text project begins at approximately \$125,000.

Lower cost alternative commercial CMS systems exist, which contain some of the qualities of higher end CMS systems. One such system is Expression Engine 2. This CMS was built on an open source PHP Web Application Framework called CodeIgniter. The use of CodeIgniter as the Application Framework, leads to a more stable product and one with additional community support and better third party modules.

Custom/Bespoke CMS Systems and Application Frameworks

Using an open source or commercial CMS may be quick and easy for some technically inclined people, but when you need to build something unique, using an open source CMS can become a time consuming and expensive mission, for the reasons summarized above. If a custom or “bespoke” CMS is built, it can be created unique for that client and the end solution will likely be leaner, sleeker, faster and often easier to use. But creating a custom CMS can also be a costly and time consuming project.

JK Design doesn’t advocate developing completely custom CMS systems from scratch. JK Design has created our own custom CMS framework built on a very popular PHP Framework, www.codeigniter.com. And the core CMS includes pre-coded modules that perform the most common content management and Web site functional needs.

JK Design’s Custom (aka: “Bespoke”) CMS

JK Design’s focus is to provide a user-friendly Content Administration Interface – and a good CMS core with an extensible CMS Framework, that includes the most common CMS and Web site functions appropriate for most Web site projects.

JK Design’s Bespoke Content Management System is a custom CMS Framework that is built on top of the open source CodeIgniter V2 PHP Framework. It was designed to run on the LAMP platform (Linux, Apache, MySQL, PHP) but is portable to other platforms. It also utilizes a Model-View-Controller (MVC) pattern and thus front-end interface design is separated from back-end business logic and content/data; this makes future redesigns and adding new functionality simpler. As a Framework, the CMS can be easily enhanced by any talented PHP developer. Further CMS technical documentation can be found on our Website at <http://www.jkdesign.com/jkdesign-cms-technical-summary.pdf>.

JK Design’s Bespoke CMS is Cross-Platform

Unlike many CMS systems, the JK Design Bespoke CMS is cross-platform compatible and can also run on Microsoft Windows 2003+ and IIS V7, if properly configured. In addition, our custom CMS can utilize other databases such as Oracle or Microsoft SQL Server, though we consider these more complex database systems overkill for many projects; they also require a Database Administrator.

JK Design’s Bespoke CMS Source Code Availability and Vendor Neutral Web Hosting

Our CMS is a Bespoke CMS built in-house at JK Design and is thus considered a proprietary system; and although it’s built on top of the open source CodeIgniter PHP Framework, our PHP source code is not open source. However, our clients are provided the source code, documentation, and a license to maintain or enhance it themselves (or with a different vendor), should they choose to do so. Our CMS is also not a “Software as a

Service” (SaaS) system; care should be taken when using SaaS vendors – if the SaaS vendor goes out of business – your site and its contents instantly disappear. We install our CMS on a Web server controlled by the client. We can also setup content extraction and source code backup processes that are in addition to what a Web host can offer as typical backups. Thus you cannot be “held hostage” and forced to use JK Design or SaaS services, as you may be with other vendors.

JK Design’s Bespoke CMS – the best of Commercial, Open Source and Custom

JK Design’s PHP CMS Framework is the best of both a commercial off-the-shelf product, an open source framework, and a custom CMS. The advantages of JK Design’s CMS over other open source options such as Drupal or Joomla, include the fact that we stand by our product. We are always here to help. In addition, JK Design will not start your CMS from scratch. We save you money by starting with our custom CMS Framework and library of existing modules. We then customize our CMS to meet your individual needs. Using this approach reduces user complexity of the Content Administration Interface, yet still provides you a very extensible system.

Many of our client Web sites require some sort of customization which may include things such as a specific registration process, custom workflow processes, customized data collection, integration with existing database systems, user generated content submission, custom Google Maps integration, personalization options, etc. Often these specific customizations required, are discovered after we launched the initial client Web site. And making these customizations with the JK Design CMS Framework, simply requires a talented PHP developer.

Some of our smaller clients have no custom requirements other than a custom interface design. For smaller clients, we can launch their site or microsite relatively quickly, using our JK Design Bespoke CMS. Our Bespoke CMS includes modules for the most common functional requirements. And as a client’s business grows, they may request to have other modules added, or they may request specific customizations which we can code for them as extensions to our Framework.

For a demonstration of our Bespoke CMS Framework and additional technical documentation, or an evaluation of your specific custom needs, please contact us for a free evaluation.