WHITE PAPER

OPEN SOURCE vs. PROPRIETARY CMS:
WHICH IS STRONGER?

Open Source vs. Proprietary CMS: Which is Stronger?

Content Management System (CMS) software runs more business websites every day. Hundreds of brands exist, costing anywhere from nothing at all to tens of thousands of dollars.

Which brings up an obvious question – why would anyone pay for a CMS, if free ones exist?

Simply put, there are two kinds of content management systems: **Open-Source and Proprietary (Closed Source)**.

a. In an open-source CMS, the underlying source code is freely available to developers for modification. Development and support are performed by a large network of volunteer developers, nonprofits, and some paid developers. Typically open-source software is free.

b. In a proprietary CMS, the underlying source code is compiled and NOT freely available for modification. Ongoing product improvement and support are performed by the company who developed the product. A license fee is charged.

Each type of CMS has advantages and disadvantages that go way beyond price.

Examples of Open-Source CMS and Proprietary CMS

The most popular free/open-source Content Management Systems used today include[1]:

1. WordPress
2. Joomla
3. Drupal
4. DotNetNuke
5. Typo3

All are free for businesses & individuals to use in building websites. Change the source code however you want. Often programmed in PHP, these CMS have large developer bases behind them. Many skilled developers volunteer time & code outside of their day jobs.

Several popular proprietary Content Management Systems are[1]:

1. Telerik's Sitefinity CMS
2. Kentico CMS
3. SiteCore
4. Ektron
5. Autonomy TeamSite

You must pay a licensing fee to use these. ASP.NET is a popular programming choice, as is Java. The development companies who create the product also provide support, and sometimes license partners to do the same. You may build on top of the CMS, but its source code is not open to change (except by licensed developers).

Both open-source and proprietary CMS do the same job—help you build & manage websites. Their underlying elements convey varying advantages – and disadvantages, depending on your business priorities.

We'll go down the list of elements, identifying the issues relevant to businesses. And which option is the stronger for each. At the end we'll determine what the stronger CMS choice is – proprietary, or open source.

Developer Type: Working for Themselves, or For You?

In terms of sheer developer numbers, open-source CMS has more than proprietary. For example, Telerik, makers of proprietary Sitefinity CMS, has a 500-developer team. By comparison, there are 10,000 developer accounts in the open-source Drupal community.

Most open-source developers are not paid to work on Drupal; they are typically self-taught and contribute outside of a day job. ALL of Sitefinity's developers are paid and professionally trained.

The difference between them is experience level and accountability. Most open-source developers **are not paid** to work on Drupal; they are typically self-taught and contribute outside of a day job. ALL of Sitefinity's developers **are paid and professionally trained**. Their employment is dependent upon producing solid, functional code for Sitefinity's customers.

Would you rather be at the mercy of open-source developers who have no obligation to help you...who could even discontinue their product without recourse?[2] Or would you rather work with a proprietary CMS developer who gives expert assistance in real time, and whose obligation is to you, the buyer?

CMS Support Options:
Decentralized Network vs. Dedicated Staff

"If your organization wants professional support, it should hire a professional company. Every solid CMS has a forum where you can ask questions, and there are many blogs that post tutorials on how to implement features or solve problems, but without good support there are no guarantees." – DigitalActivist.net [3]

A large volunteer network means many possible support options for open-source CMS – forums, documentation, contracted help. However, the network's sheer size and the "volunteer" aspect lends itself to a "someone else will do it" mentality.

Most open-source developers are independent. Which means no accountability to anyone (not even you!)

Also, most open-source developers are **independent.** Which means no accountability to anyone (not even you!). What happens when your open-source developer goes MIA in the middle of your project?

Proprietary CMS developers are typically employed full time at development companies and are paid to provide support. So they're accountable for their work and response times.

Plus, **development companies** (as opposed to independent developers) are more likely to invest their money and developers' time in reliable "business grade" software solutions and work to achieve partner status. For example, it took PlanetMagpie three years to become a Sitefinity CMS Platinum Partner. The effort included developer trainings and certifications, successful product implementations, case study submissions, etc.

continued...

Website Security: How Vulnerable is an Open-Source CMS to Attack?

Open-source CMS loses the security battle due to the serious challenges it faces. Proprietary CMS applications have a clear advantage when it comes to protecting your website and its users.

In some cases, a CMS' underlying code promotes vulnerabilities. In others, the very openness of open source leaves it vulnerable to attack.

Code Vulnerabilities

The Secunia website[4] hosts a large database of security vulnerabilities for popular software. These are its current results for the top CMS we listed above.

PROPRIETARY	OPEN SOURCE
Ektron 1 Advisory, 2 Vulnerabilities (ASP.NET)	**DotNetNuke** 5 Advisories, 10 Vulnerabilities (ASP.NET Freeware)
Kentico 1 Advisory, 1 Vulnerability (ASP.NET)	**Drupal** 6 Advisories, 12 Vulnerabilities (PHP)
Autonomy TeamSite 0 Advisories, 0 Vulnerabilities (Java)	**Joomla** 6 Advisories, 13 Vulnerabilities (PHP)
Sitefinity 0 Advisories, 0 Vulnerabilities (ASP.NET)	**WordPress** 14 Advisories, 34 Vulnerabilities (PHP)
SiteCore 3 Advisories, 5 Vulnerabilities (ASP.NET)	**Typo3** 23 Advisories, 88 Vulnerabilities (PHP)

As you can see, a large volunteer developer base doesn't necessarily mean the open-source CMS has better security. Because of its openness it invites security flaws which will leave your website exposed.

"A large volunteer developer base doesn't necessarily mean the open-source CMS has better security."

Openness to Attack

For open-source CMS, support is usually given through a volunteer network. This means lots of eyes to spot security issues. Consequently, many serious vulnerabilities in popular open-source CMS are caught and fixed in due course[5] (though minor vulnerabilities may be left for "someone else" to fix).

For a proprietary CMS, a team of paid developers works to find and remove vulnerabilities. Fewer eyes, but highly-trained ones to spot security issues & provide support.

However, open-source CMS falls down when it comes to extensions.

Website Extensions

Most companies want to add extra functions to their CMS, so it will support their web marketing plan. This is done through extensions – add-on software built either by the CMS maker or third-party developers. If you've used WordPress, "plugins" are extensions.

Third-party extensions are usually made by 1 developer, or a small group, donating their time. Often not vetted for security by the CMS maker. **These extensions are now the biggest security risk to open-source CMS.**

There's no guarantee that open-source CMS extensions are safe to use. They can contain:

> » Accidental security holes
>
> » Purposely-built backdoors into the website
>
> » Malicious code dangerous to your servers

Extensions for proprietary CMS are more likely to come from the CMS maker, or licensed developers. These are professionally developed (and implemented), so the developers have no reason to include malicious code or backdoors.

Extensions are very useful in content management systems. Before using one, review its documentation and its developer's reputation. This way you can make sure the extension is legitimate before implementation.   If you can't verify it like this, don't use it. You could risk your entire website.

Updating: Who's Responsible?

Proprietary CMS updates are only done after a QA Engineer verifies their safety. Security issues have a set process for resolution, such as a provider-maintained update schedule.[6]

Open-Source CMS usually do the same. Drupal is noted for its security update process.[7] However, these CMS don't always have a central authority to direct it & be accountable.

You must plan for & implement updates yourself. If you don't, over time your site becomes more vulnerable to hacking. A single line of code can ruin all the work you've done.[8]

The web is constantly evolving and your CMS software needs to keep up with it. With open-source CMS, feature enhancements are left up to the volunteer developer community and take place at their own pace.

With proprietary CMS, new features are what keep them in business. Proprietary CMS companies actively seek feedback from their development partners on desired features. Those features are very often added to the product roadmap and quickly become part of the product.

> *Proprietary CMS companies actively seek feedback from their development partners on desired features. Those features are very often added to the product roadmap and quickly become part of the product.*

Installation/Portability/Hosting

How easily can your CMS be installed, or moved from one server location to another? You may never need to consider this (or only once). But when you do, it's a BIG concern.

Where can you move the CMS? Any new host must be able to support the CMS' underlying technology, such as databases and application servers (e.g. ASP.NET, Java, ecommerce). Not to mention the other components which make up a functioning website. Many of these elements are not portable, and must be configured on the host servers themselves.

In the past, it was easier to move a proprietary CMS due to business-grade hosting providers supporting business-grade databases. Nowadays, most hosts will support open-source and proprietary technologies equally, but open source is notorious for lacking detailed hosting documentation (back to the blogs and news groups for support).

Administration: How Much Time Will You Invest... Over Time?

There are two aspects to CMS administration. One is the admin user experience—how "friendly" the CMS is for a non-technical user who's updating the website. The other is customization—how easy it is to modify the website, and build new features or add-ons.

User Experience

Open-source CMS developers must concern themselves with how the CMS works. Trouble is, they do so from a **developer's** viewpoint:

"People with technical backgrounds have a very good knowledge about the underlying technologies used to build the CMS. Therefore, engineers will usually find an exemplary CMS task easy to perform and self-explanatory, but this is certainly not the case if a non-technical user were to perform this very same task."[9]

continued...

> *"People with technical backgrounds have a very good knowledge about the underlying technologies used to build the CMS. Therefore, engineers will usually find an exemplary CMS task easy to perform and self-explanatory, but this is certainly not the case if a non-technical user were to perform this very same task."*

Those who make updates to the company website need an interface that is user-friendly, and doesn't require a call to IT every time. This is why many proprietary CMS companies have in-house UX (User Experience) specialists. It's their job to make updating the CMS easy for **non-technical users**, saving on additional developer expense after the site goes live.

Customization

When you try to modify an open-source website, an even bigger disparity emerges.

The ability to modify any aspect of an open-source CMS appeals to many organizations. That is, until you consider how much work is needed to create and maintain layout changes, new content, new services…[10]

Since open-source CMS have their source code freely available, any and all aspects of the CMS can be modified or added to (so long as you have the time). Proprietary CMS, by definition, do not have the same flexibility. However, that doesn't mean you can't customize a proprietary CMS.

In fact, proprietary CMS developers often put full API reference materials on their websites.[11] In case an on-site developer wants to create or modify code that accesses the CMS (for example, to build a new feature on top of the CMS' available resources).

License Cost: Is "Free Software" Really Free?

The biggest perceived advantage of open-source content management systems is that they're free. No license costs. So where does their funding come from?   Funding can come from donations, some nonprofit foundations[12], or paid add-ons. Nowhere near a solid income stream.

On the other hand, proprietary CMS are funded by sales of the software and paid support.

Even if funding isn't a concern to your organization, there is another cost to using a CMS every organization must consider: **Time.** Design, implementation, ongoing maintenance, upgrades, and constant battles against security breaches.[13]

Consider this: The average lifecycle for a CMS is 3 to 5 years. Figure 5% of a CMS' total implementation cost comes from the licensing fees. Reducing that 5% by saving on the licensing can drive up other

implementation costs (such as maintenance time, security patching and customization).[14]

The 'savings' evaporates over the course of the CMS' lifecycle.

You Get the CMS (and Support) You Pay For

Judging from the categories listed here, **it's clear that proprietary CMS is a better choice for most business Web needs**. Proprietary provides reliable support, increased security, and it can even be less expensive in the long run to operate.

Many organizations like using open-source CMS just because they're free to download. But the "it's free!" argument stumbles when you consider factors like increased development time, less stable platforms, security vulnerabilities, and lack of paid support.

Many organizations like using open-source CMS just because it's free to download. But the "it's free!" argument stumbles when you consider factors like increased development time, less stable platforms, security vulnerabilities, and lack of paid support.

Even the price argument doesn't hold up. A Sitefinity CMS standard license goes for $1,999, about equal to 13 hours of web developer time. Do you think an open-source CMS site will only take 13 hours more to build than a proprietary CMS site? Hope so if you go that route. Don't forget to factor in maintenance costs over time, security issues, etc.

Another web development firm weighs in:

"Some enterprise companies design with open source because they have qualified developers. If you don't feel that your team is ready to take on extra challenges of working with an Open Source software, then Closed Source (and its built-in support) typically is the best route."[15]

Remember the old axiom: You get what you pay for. If not up front (licensing cost), then later on (support, maintenance, recovery from attack). Choose carefully!

P.S. – Remember to consider the end user when you make your decision. See the CMS Critic article, "What to Look for in a CMS from an End-User Perspective"[16] for advice.

Sources

1   **Closed or Open Source: Which CMS is Right for Your Business?** – Mashable.com, 5-5-2011

2   **Umbraco - v5 RIP** – Umbraco.com, 6-13-2012

3   **Select a Content Management System** – DigitalActivist.net, 6-30-2011

4   **Secunia Advisory and Vulnerability Database** – Secunia.com

5   **In Depth Analysis of Open Source CMS Security** – CMSCritic.com, 1-19-2012

6   **Hotfixes List: Kentico DevNet** – Kentico.com

7   **Keeping Drupal Secure: How the World's Largest Open Source CMS Combines Openness and Security** – Acquia Blog, 2-28-2012

8   **Lessons Learned from a Hacked Website** – Digett Blog, 12-28-2011

9   **Sitefinity CMS Compared to Open Source Solutions** – Telerik.com White Papers

10   **Competitive Landscape: Centralpoint** – Oxcyon.com, 2009

11   **Sitefinity Documentation: Fluent API** – Sitefinity.com

12   **OpenSourceMatters** – Joomla Non-Profit Foundation

13   **Is Open Source CMS Really Free?** – Kentico Blogs, 4-24-2012

14   **The Siren Song of Open Source CMS: Should You Listen?** – Sitecore.net White Papers, 2011

15   **Web Design: Open Source vs. Closed Source Software** – Terradon Communications Group, 9-13-2012

16   **What to Look For in a CMS from an End-User Perspective** – CMSCritic.com, 10-25-2012