**MMT DIGITAL**

# How to select a Web Content Management System (WCMS)

## Contact

**Midlands Office**
Will Hawkins
E will.h@mmtdigital.co.uk
T 01572 822278

**London Office**
Becky Sharp
E becky.s@mmtdigital.co.uk
T 020 7242 5698

## Overview

Most large organisations need a Web Content Management System (WCMS). Regardless of the technical abilities of a company's in-house teams, gone are the days where having a static HTML website is good enough. Content has to be updated regularly, quickly and it has to be cost effective.

Any half decent modern WCMS provides more than just a platform for changing text and images. Whilst that is still important it is also about managing blogs, news systems, RSS feeds, wikis, forums, user generated content (UGC), SEO, multi-language support, ecommerce, data capture, interfacing with real-time Web 2.0 data streams and integration with existing bespoke or 3rd party business tools (to name but a few). A good WCMS should be able to form the corner stone of your online presence, whatever you are trying to achieve.

Trying to choose the right WCMS is difficult. Over the last 10 years we have written two of our own WCMS using classic ASP and, more recently, in ASP.NET. That was back in the 'good old days' when clients were really impressed that you had built your own WCMS. These days it is one of the number one rules; "Don't use a bespoke WCMS built by an agency". A few years ago we realised this and went through the process of choosing a WCMS that we would use to deliver all our clients websites. This was a massive decision for us. It was a long term commitment and the quality of the product we chose would be reflected in most of the work that we would produce. We had to get it right and we think we have done.

## What to do and what not to do

### Should we go 'open-source' or 'ASP.NET'?

It's a question we hear getting asked a lot. Despite the fact that is doesn't actually make a lot of technical sense (there are plenty of 'open source' .NET based WCMS), it should probably set off some alarm bells.

Choosing a WCMS should not be the task of the IT department. It must be led by your key business goals and drivers. The question above is relevant if, and only if, you have a team of internal developers. In which case the programming language that the WCMS is written in (and can potentially be used to extend or create bespoke functionality) is relevant. If you don't have internal programmers then you should not be worrying about this.

Focus on your key business needs. What are you trying to achieve with the site? Are you trying to access and support a global market? Are you trying to improve operational efficiency? Do you need a highly accountable system? Is standards compliance important? Are you trying to create a community? These are more important questions than "Is it hosted on Unix or Windows?" but all too often we see questions like that at the top of the list.

## What are the key features?

Most decent systems come with loads of off-the-shelf functionality. Advice that we see commonly given is to "do a features matrix". It is an important exercise but you have to be careful. Don't just ask for yes or no answers, as whilst it might help to remove some of the weaker WCMS there is such a broad spectrum of ways that a feature could be implemented that you really need more details.

If a WCMS supports "Workflows and versioning" you need to see how it works. Does it need a WCMS developer to setup a workflow or is it straight forward? How powerful is the feature? The most basic feature that all WCMS will offer is, of course, "editing the content of the page". It is so important that you try to establish how this works. What is the editing environment like? How user friendly is it?

Don't just ask for yes or no responses. Ask people to provide screenshots and an explanation of how it works. Even better (certainly once you are down to the final few) try to get an online demo. Get a guided tour of how it all works.

The website http://www.cmsmatrix.org is not a bad place to start looking at a simple matrix.

## Features you might not think about

There is some more 'techy' stuff that you should also make sure you consider before getting carried away with all the exciting features.

### Search Engine Optimisation (SEO)

Most people are aware of the importance of SEO in web development. As with any features, avoid asking "does your CMS support SEO?". In almost all cases this would be "yes". Ask for details about how exactly it helps with your SEO. At a minimum a WCMS should offer the following:

1. *Friendly URLs*
   Some WCMS use parameters in the URL. i.e. a news article might be on the URL:
   www.mysite.com/news.aspx?id=101
   Another news article might be on the URL:
   www.mysite.com/news.aspx?id=102
   As well as being a pain when it comes to site analytics it is bad news for SEO as the search engines like user-friendly URLs. Make sure that the WCMS generates URLS more like this:
   www.mysite.com/news/title_of_my_news_article

2. *Custom URLs*

Friendly URLs can end up being quite long and you often want to direct people to a specific page in your site as part of a marketing campaign. Custom URLs allow you to change a URL like:

www.mysite.com/products/books/history_books/bestsellers

to something like this:

www.mysite.com/history_bestsellers

3. *SEO \ Accessibility support*

You need the ability to add page title, description and keyword metadata to each page. You should be able to add 'ALT' tags to images and 'TITLE' tags to links. It should allow you to build CSS based navigation and use search engine friendly page tags like H1 and H2.

## Scalability

You should ask questions about the size of sites that can be created using the WCMS. Not just the number of pages and documents but also the number of users that can be supported in the database. Is there is an upper limit (which, of course, there will be)?

Can it be scaled up by running the WCMS across multiple servers (a server farm)? It is also important to ask questions about the load testing on the site. How will it perform during spikes in demand? Of course this will, in part, be down to the quality of your hosting platform but the efficiency of the WCMS when it comes to processing and serving page requests is an important factor.

## Hosting

Other than an open-source WCMS, almost all hinge around a licensing model. You need to think about whether you are going to host the WCMS on any existing servers that you might have or whether you are happy for it to be externally hosted.

If you are developing your site through an agency and are happy to host with their providers then the license fee is not as important. The agency is likely to have a server wide license for the WCMS.

Of course, the license cost still needs to be considered because you need to look at your worst case scenarios. If you fall out with the agency you may have the need to transfer the site to your own servers at which point you will incur a license fee for the WCMS. If your site generates a huge amount of traffic, you may need your own dedicated server and you will need to be aware of any licensing costs at that point.

## How much does it cost?

Of course, cost is important, but there are some serious pitfalls. You cannot make any real correlation between the price of a WCMS and how good it is. There are plenty of very powerful open-source (i.e. free) WCMS and there are plenty of very expensive "enterprise level" WCMS that really have fallen behind over the last few years and just don't offer value for money.

This slightly comes back to the 'open source' versus a more commercial solution argument but we see far too many people obsessing over the fact that "open-source PHP-based solutions mean that the WCMS is free and the hosting is cheaper". It is dangerous ground. Yes, the WCMS is free.

However, the licensing costs for a commercial solution maybe a tiny percentage of the build and maintenance costs. Open-source solutions tend to be more poorly supported and can lead to higher development costs when it comes to building customised functionality.

In 2005, the team behind the open-source WCMS "Mambo" abandoned the project and started Joomla. This was a nightmare for many companies that had adopted Mambo. With open-source systems, the product stability and quality assurance processes (as well as application security) can be weaker.

In an open-source project anyone can contribute to the code. It leaves systems wide open to hackers and formal quality programmes are rarely followed. The argument that it "saves hosting costs" is also a bit of a non-starter. If you are using a cheaper 'virtual hosting' environment then a Windows based server might cost you £10 per month more than a Linux based solution. It is hardly worth basing your decision around an annual saving of £120.

If you are hosting on a top end, dedicated web server with a dedicated database server then you might be paying as much as £20,000 a year. Most of that is for the hardware and the support. The Windows licenses will be a few hundred pounds more than the free Linux license. Again, it really should not factor highly in the decision making process but all too often we see that it does.

There are two points where cost needs to be considered. At the start of the process, when you are choosing a shortlist, most companies will have a top end budget and will have to rule out certain WCMS that are very expensive. Right at the end of the process, if you really can't choose between two WCMS, then you might have to go with the cheaper one.

## What is the release cycle?
This just means "how often does a new build of the WCMS get released". You do not necessarily want the new features and you don't necessarily want to be forced into an upgrade at each release. However, the release cycle gives you a good idea about how actively the WCMS is being developed.

## Get a demonstration
Certainly, when you have narrowed the choice down to a few you need to get an online demonstration. It is usually better to get someone to give you a guided tour of the main features. Installing and trialling the WCMS software yourself can be very time consuming but if there is an online trial version setup that you can log into and have a play, so much the better.

## Why we chose Kentico

The short answer is that we did everything that we have described above and Kentico came out on top. The actual process that we went through was as follows:

1.  *We ruled out anything that was not based in ASP.NET.*

    OK, that sounds a little bit like we broke one of our own rules. However, whilst this clearly is a very technical decision, it was actually a crucial need for our business.

    Our server-side skills are based around .NET development. It is vital to us that we can do much more than just deliver Kentico sites with the off-the-shelf features. Most of our clients require bespoke webparts for a variety of reasons from interfacing with 3rd party database software or payment gateways to displaying Flash based 'page-flip' versions of a textbook within a secure environment.

    We need to be able to use Kentico as a platform for our bespoke solutions. Not only in terms of the bespoke functionality that we deliver to the end user but, also, creating a single administration area for all the functionality. Kentico allows us to do this very well. The 'front-end' of a Kentico site can really look and feel however you want and do whatever you like. We can build bespoke 'modules' into the admin area so that site managers can control all aspects of the site (off-the-shelf and bespoke) from a single place.

2.  *A large number of features were important to us.*

    Our criteria here are a little different to an organisation looking for a WCMS to deliver a specific website. We are using the WCMS as a platform to deliver many different websites for a wide variety of clients. We don't know what our clients will want before we work with them, so the more 'feature rich' the WCMS was the more cost effective solutions we would be able to offer our clients.

    We started with the usual matrix comparisons but, as has been discussed in this document, we made sure that we looked in as much detail as possible at how the WCMS in our shortlist actually delivered the features.

3.  *It had to be highly customisable and very extensible.*

    We don't just deliver off-the-shelf solutions to our clients. In fact, it is very rarely the case. Our clients do not expect to be constrained by the technology. Yes, it is important that there is a cost effective off-the-shelf option available to us for when budgets are tight but it is equally important that the features of the WCMS (and the WCMS itself) are highly customisable and extensible. We were looking at this from a technical and a creative point of view.

Firstly, it was important that the built-in features were very flexible in the way that they can be made to look and feel. We work with strong brands and creative agencies. If we can deliver what they need just by re-skinning built-in functionality then we can save our clients money. However, in some cases we simply have to build extra, bespoke functionality. We needed a WCMS that either gave us access to the source code, provided a way of extending and building new 'webparts' or had a powerful API. Kentico gave us all of these options. Over the last few years we have built many of our own webparts that have been integrated into our client's sites.

If we have to create a highly bespoke management tool in the administrative area (Kentico's 'CMS Desk'), we can create new 'tabs' in the interface and \ or bespoke 'modules' so that site administrators manage all of the built-in functionality alongside the bespoke functionality from one place. This was very important to us.

4. *It had to do the basics well.*

For all the fancy built-in features or bespoke webparts and modules that a website might have, most of the time clients are just editing fairly standard content.

For example, adding news articles, editing text on the home page, changing contact details on the 'about us' page etc, etc. Not very exciting stuff but it is important that this basic editing environment is simple to use. Most of the time, it is used by people who are not highly IT literate.

The CMS that we had previously developed in-house had a really fantastic on-page editing environment that was about as close to using Microsoft Word as you can get in a browser. Kentico also has a brilliant page editing environment and this helped push it to the top of our shortlist.

5. *We had to be able to use it for small clients and multinational organisations.*

We have a large range of clients from local businesses to multinational organisations. We needed to make sure that the WCMS could be scaled right back so that it could be used on smaller sites without overwhelming the site administrators.

On the other hand it was important that the WCMS had powerful, enterprise level user management. We needed a WCMS that offered a very granular security model so that different levels of users could be setup that had different levels of access to different parts of the website.

We also need to be able to setup workflows for those different users and allow site administrators to manage those workflows and the users that they are assigned to.

Finally, we needed a WCMS that had good multi-language support. Kentico was very impressive in all these categories. On top of this, Kentico 5.0 has a fantastic

versioning tool. It allows 'side by side' comparisons of any two versions of any page on the site (it works a little bit like Microsoft Word's 'track changes' features).

6. *We had to be able to see it in action.*

> We could not take a risk on a WCMS that we could not actually have a very detailed look at. We ruled out WCMS that offered no free trial or online demo straight way. Once we had our shortlist, we sat through a number of online live demos to get an idea. We also took the time to install demo versions on our server to have a more detailed look.

Our shortlist came down to Sitecore, Ektron, Umbraco, DotNetNuke and Kentico. For us, Kentico was actually an easy winner. The others on our shortlist are excellent products but, for the reasons outlined in this document, Kentico was the one that best suited our needs. Hopefully this whitepaper might help you find the one that best suits your needs. If it turns out to be Kentico and you need someone to help with the development, give us a call!