

# Squiz White Paper

## On Toolkits and Temples:

### A Guide to Successful Publishing of Multi-Channel Web Sites (in Particular, Universities, Publishers and Multi-Brand Corporations)

February 2007

Contact [info@squiz.co.uk](mailto:info@squiz.co.uk)

#### Executive Summary

This white paper looks at the challenges presented by publishing multiple web sites in any distributed organisational environment – such as **universities**, **publishers** and **multi-brand companies** – where sub-brands or departments *must* have their own unique web identity and/or services (and must therefore deploy their own production resources), whilst at the same time balance their own web requirements against that of their parent organisation.

A classic example is the work that Squiz has conducted with Oxford University in producing the institution's network of Library Services web sites. The production and support issues are neatly summarised by Bodleian Law Librarian and network (web) project leader, Ruth Bird:

“With libraries working independently to produce their own sites, a number of problems were arising. Most notably, the look and feel of each site was different, and this lack of consistency was leading to inefficiencies from a user perspective. Essentially, we were making it harder than it needed to be for users to access their resources.

“From a technical standpoint, support issues were also growing, and, as the people responsible for helping the library network, this was unsustainable for us. My web team is small, and their ability to cover all the bases was being compromised. There were simply not enough hours in the day and skills within the team to manage such a disparate group of sites!”

Based on our experiences with Oxford University, and other multi-channel organizations and/or brands such as Greenwich University, Future Publishing (who publish their online stable of magazines using MySource Matrix), Lema Publishing, Mercator Media, the NHS Modernisation Initiative, Eleco (who publish their company brand sites using Matrix), and US Airtours (ditto), this paper provides some best practice guidance on how to create a web publishing strategy that minimizes the types of problems described above.

In Squiz's experience, the provision of such solutions involves two distinct and equally important areas of change: on one hand, the delivery of new technology tools as an enabler; and on the other, the introduction of new organisational and cultural practises in order for new technologies and working processes to be adopted.

As such, we will recommend a grassroots 'bottom up' approach to introducing new ideas, rather than a 'top down' master plan. In our experience the latter - although great on paper - usually proves to be: impractical in political terms; too expensive; and often obsolete in terms of the timing of deliverables (ie, grand systems generally take too long to build in this type of environment).

### **Using Toolkits vs Building Temples (or 'The Joy of Telephones!')**

To communicate with customers, colleagues, etc, we use a variety of tools everyday: email, fax, phones. Let's take the phone as our example. To talk to someone I simply key in the number and I'm away. I'm not concerned with BT's local loops, exchanges, or general telephony protocols. The phone is GREAT. It's easy to use, and from a service provider's point of view it requires very little in the way of technical support. It's also hard to break (mobile phones can even survive a good dry cleaning!).

Now, to put a web page in front of someone, a high proportion of today's web authors need to....

- i. ...get trained in a web authoring application (eg, Dreamweaver)
- ii. ...get familiar with their organisation's web templating system
- iii. ...get a good understanding of their organisation's branding conventions (and an appreciation of the reasons why they exist!)
- iv. ...become a great designer
- v. ...get a user account, and permission to publish from a system administrator or a computer services department
- vi. ...become a great copy writer
- vii. ...throw together some stylesheets, some html, some images and upload them to a webserver
- viii. ...ensure that the production of the above has not thrown out any previous pages (eg, linking, etc)
- ix. ...coerce other people within the organisation to provide links to this new page to enhance its usefulness
- x. ...etc

Essentially, each time an author wishes to create a new page they need to build the structure of the site all over again - before they even get to writing page copy or placing images on the page.

The phone is a tool. It's a no-brainer - we just use it and it does what we need it to do.

Creating a new web page, on the other hand, is like building the Sistine Chapel. Each time they wish to communicate something new, authors must become the (web)master of many trades - architect, bricklayer, plasterer, and politician. (And, since such people are rare, this situation also places a large burden on an

organisation's IT support team, who often end up providing services beyond their remit – such as debugging Cascading Style Sheets.)

In practise, maintaining and rebuilding multiple web sites as 'temples' in this way becomes unwieldy and counter productive for a variety of reasons, including:

- Unless extremely well authored and/or policed, they tend to dilute the organisation's brand and identity: page authors will interpret brand guidelines in idiosyncratic ways (ie, "wouldn't it be cool to make the navigation float around the page!?")
- Whole sites tend to break because authors attempt to introduce functionality that they're not skilled enough to implement
- Central IT team resources tend to get stretched as they're required to support a variety of non-standardised practises, applications and technologies
- As sites become faulty and disorganised over time, the user experience deteriorates, resulting in less use of the site/services
- Longer term sustainability of the site is threatened as working knowledge of the system tends to reside within 'individual experts' rather than documentation and processes – and these valuable individuals have a habit of moving on

Clearly this situation is untenable. In our experience, when departmental or brand-level sites are deployed without a supporting technological and/or cultural system in place, anarchy prevails. And although this experience is both expensive and painful it's surprisingly common.

Fundamentally, the problem lies not in the existence of site 'temples' as described above, but in the ownership and the position of them within an organisation. A web site is a critical part of every firm's make up – it's there to communicate, sell, generate leads, service users, etc – and as such it must occupy a position of strategic importance. However, this importance can be misappropriated and the overall effectiveness of the system can be compromised.

So, let's look at how a network of web sites can be managed more effectively....

## A Temple Blueprint

A good web site should possess the following basic attributes:

- Easy to maintain by non-technical authors
- Different site elements should be maintained separately – design by designers, content by authors, code by developers

- Site design should be managed by a template-style system
- Easy / quick to publish
- Easy to plug in / roll out new features
- Easy to extend into different areas – ie, to deploy new sites and/or services
- Simple management of site authors and general production via some form of workflow

In practice, a good web site should simplify the authoring process. To create a new web page, an author should do the following:

- ...browse to a page
- ...click the 'edit' (or 'add a page') button
- ...sign in
- ...edit the page / write a new page
- ...upload an image or two
- ...insert relevant links
- ...click 'save'
- ...go get a coffee!

From a systems point of the view, a good web site should provide a consolidated technology platform and approach to training, documentation and support – and ultimately a reduction in support and production costs!

From a marketing and services perspective, a good web site should deliver a consistent branding and messaging; quicker and cheaper delivery of new communications channels – such as, blogs, RSS, etc; and the the delivery of more sophisticated user services – private zones or extranets (eg, student research zones, media zones, partner enablement areas), etc – with a view to driving more leads / revenue / usage of the site.

OK, so much for the 'temple' blueprint. How do we build one?

## Building Effective Templates...

There are two tricks to improving site production within a multi-channel environment. The first lies in selecting the right technology. The second lies in selecting the right 'building site.'

---

## Selecting the Right Technology

Your end users and your site authors don't really care *how* you build your site. Both of them view it as a tool – to either work within or to access. So, all you need to worry about is ensuring your tool is usable and accessible.

- From an end user's perspective, this means following best practice design guidelines (please, no floating navigation!) and building your site around standard web formats such as [XML](#) (Extensible Mark Up Language), [CSS](#) (Cascading Style Sheets) and [XHTML](#) (Extensible Hypertext Mark Up Language) to ensure that they can be accessed in tact. (In addition, the use of standard technologies will ensure the lifespan of your site and/or services, since by making this choice you are also opting into the widest possible pool of skilled people who can support and develop them in the future.)
- From a web author's perspective, this means providing them with a simple application within which to work – ie, a familiar interface that uses principles such as file trees; a 'Word'-like WYSIWYG authoring console (What You See Is What You Get) that enables them to easily format text and insert images and links; and beyond that to not have to worry about much else! (Importantly, from a support and site production point of view, this tool ought to be centrally managed so that different users can be given access rights for specific parts of a site, and their authoring rights can be either restricted or expanded according to their application proficiency.)

So, where do you find such technology? The answer is simple: in a Content Management System.

A CMS is a software application that helps organizations control their information. A good CMS will help you create, manage, and publish your content to a web site, intranet or other collaborative online application. Behind the scenes, a CMS uses a database to store, manage and retrieve content in a consistent fashion, which enables you to radically enhance the way you publish your sites/services. And a good CMS will always separate the presentation of a site (the design) from its content so that you can deploy specialist resources to where they're most needed – ie, designers can focus on designing and content authors can focus on writing.

On a small scale, a CMS can dramatically improve the way you control the presentation and design of your site. A larger implementation or 'enterprise' CMS can transform the way your entire organization operates by improving the creation, management and retrieval of information. The most common form of content management is Web Content Management which allows people to easily control and update their web sites. When a web site gets very large or it is published in multiple areas, a CMS allows content to be aggregated and re-used efficiently, and for a team of non-technical authors to contribute easily to the content creation process.

So, where can you find a good CMS?

Well, the good news is that behind every important piece of web technology there is a multitude of interested parties, the majority of which can be tapped in to free of charge – otherwise known as ‘open source’ software development.

As a movement of developer talent, open source software generates solutions faster than any traditional software firm could ever manage. The quality of its code is unsurpassed, and its focus on market dynamics is unique. It scores highly on both of these things because of the way it’s organized. Firstly, a problem (or a market opportunity) is spotted, then a piece of software is developed to address it. Secondly, this software is released to the outside world in a naked form (accessible source code) and subjected to peer review, then it is tweaked and improved. Thirdly, it’s used rapidly in production environments by people who are empowered to offer feedback into the development effort. In this way open source solutions can evolve at the speed of light to address common requirements. And take note – because of its unique licensing model, it’s FREE to acquire!

As such, you can install, test and deploy an open source CMS without spending a zillion pounds. Forums such as [Sourceforge](#) are a fantastic repository of open source projects and application. And, if you don’t have the resources to go it alone, then new breeds of commercial companies are on hand to help through the provision of development, implementation and support services around open source products (ie, [Squiz](#), with our market-leading open source CMS, MySource Matrix!).

### Selecting the Right ‘Building Site’

However, just having the right type of technology is no passport to success. As mentioned, it’s also important to find the right environment in which to deploy it. The technology is, after all, only an enabler for change.

Essentially this point is all about finding the right ‘building site’ for your new web site. Key elements to consider are:

- Will the target system users be able to operate the new tools?
- Will they need to give anything up?
- How much of a ‘win’ will the new system be for them? What do they gain?
- Are there any technological constraints? eg, concrete departmental or divisional policies?

Indisputably, the question of ‘giving stuff up’ is the most important of these considerations. In fact, we’ve found that it’s the single most important factor in delivering a successful web site solution in a multi-channel environment. As such, for those organizations that wish to follow this route, *we would advise against the creation of a ‘top down’ master plan.* There are many reasons for this, not least:

- ...it's an extremely complex and expensive way of proving the viability of a new web site strategy (success can never be guaranteed 100%!)
- ...oftentimes, because of their scope, 'master plans' become redundant before they are even implemented - technologies change quickly, and what may have looked good on paper 12 months ago can become obsolete!
- ...it's a good way of alienating the people you're trying to help - in any web project, stakeholder ownership is critical...it's a temple, after all and webmasters, content authors and designers do not want to lose control and/or power
- ...it's a good way of losing valuable knowledge from the project's most important people (webmasters, etc). ie, a plan that requires the scrapping of departmental / brand sites may also scrap the valuable know-how that has been built up around them - eg, knowledge of user/customer habits.

Alternatively, just as we've done with Oxford University, the NHS and commercial firms like Future Publishing, we'd recommend creating a 'grass roots' campaign in which to embed new practices and to further the cause of effective web site production.

A good process would look like this:

- Identify a smaller, side project at a departmental or individual brand level where a new, open source CMS-based approach can be implemented. This might be for the redesign of a college site, or the launch of a new product. Your ability to follow this route will be facilitated by choosing an open source CMS solution, since it's a relatively inexpensive and - if the product is developed by a commercial firm, like MySource Matrix - a low-risk path to take.
- Involve the key departmental/brand actors as the key players in the project - have them run it themselves, with a helping hand (but no more!) from your central IT team. In this way, the project will enjoy more immediate buy-in and greater momentum. As mentioned, it will also ensure that valuable non-technical know-how will remain within the project.
- Ensure that these players are:
  - Extremely motivated
  - Extremely knowledgeable about the new system benefits
  - Extremely well trained
  - Extremely good at what they do!
  - .....so that they will be in a great position to evangelize the project after its launch!

- Give the project a sensible amount of time to settle once the new site/service has been launched, and then identify a set of best practices that others might benefit from. Ensure that these are not technical in nature – eg, they should speak about how much more efficient/cheaper/reliable web publishing has become as a result of the new project. Create some documentation around these practices in order to validate them, give them prestige, and allow you to circulate them.
- Encourage other departments/brands/divisions to admire your work and make it easy for them to follow a similar route through ‘best practice’ seminars and using speaker slots on important cross-party meeting agendas. Further, as momentum builds and other parts of your organization begin to follow suit, you can also create simple education sessions to help move things on.

NB: talk to your services and support vendor about how you might implement the types of activities mentioned above. They ought to be able to facilitate these types of events and documentation by providing speakers, written material, training courses, or just plain old advice. This is exactly what Squiz has done for Oxford University, Eleco, Future Publishing and others in the past.

## Using the Tools, Living with the Temple

Returning to our tools/temple analogy, the two are not mutually exclusive. The development of a highly flexible, functional CMS toolset does not preclude the creation of a beautiful web site. Nor does it mean that your key players get disenfranchised in the process. In fact, when done successfully, it’s precisely the opposite that occurs. Here are a few examples of the work we’ve done recently (click to view the sites!):

- [Oxford Union](#)
- [Oxford School of Archeology](#)
- [Greenwich University](#)
- [Future Publishing \(X-Box 360 Magazine\)](#)
- [Future Publishing \(T3 Magazine\)](#)

In each case, the organization has moved from the relative anarchy of multiple different sites, designs and technologies, to a more harmonious, loosely coupled arrangement that’s underpinned by a single technology platform. In the case of Future Publishing, each of the magazine (brand) sites are implemented under one instance of MySource Matrix. With Oxford University, each department has maintained ownership of its site, but they have standardized on MySource Matrix as a ‘best practice’ CMS. In both cases, support costs have been slashed, user experience (and site effectiveness) has improved immeasurably, and both webmasters and content authors can produce their sites more effectively through the use of a simpler, more refined toolset (MySource Matrix).

Ruth Bird at Oxford describes the net effect: “Our new framework ensures our technology base is standardized and easy to support, and that important end user issues like branding and design are kept totally coherent. Each library site publisher now knows exactly what to do, how to do it, and who to come to should they need help.”

In addition, some important cultural changes have occurred within these organizations. They include:

- Web sites are no longer viewed as ‘temples’ to be defended. Instead, they work as publishing and communications vehicles whose content can be re-purposed for other sites and services. Overall the web site network is now viewed as a ‘confederation’ rather than a series of ‘fiefdoms.’ Importantly, the act of content authoring is now (thankfully!) recognized as one element within a greater ‘workflow’ rather than a complete designing or redesigning of a page.
- Commonality in technology and application platforms has led to a greater degree of openness in the web production process. Ideas, tips and tricks are now shared between departments/brands – and support costs are lowered as a result. In addition, simpler support and training systems mean that it’s easier for staff to switch jobs internally, and for their web skills to retain their value. This is important, because the flip side of enabling staff movement is the fact that operational knowledge of the CMS resides within the organization (in processes and documents) rather than in the hands and heads of a few people.
- The ability to repurpose content to multiple sites via the CMS has led to the production of better quality content. Since content feeds can be taken transversally and published by any site within the network (eg, in multiple online magazines at Future Publishing) then a free-market economy of content is created within the group. Good content rises, and is re-used...and authors are rewarded for quality work.
- The central (or ‘corporate’) web and/or communications teams have come to be viewed and used as experts who can add value to the site publication process – for example helping to introduce new functionality such as video, RSS, blogs, etc...as opposed to a dictatorial force to be ignored or avoided!
- IT support teams are getting more opportunities to drink coffee!

Of course, one can’t trivialise the amount of work required to make the above improvements. It’s taken Oxford University around a year and a half for a grass roots movement to take hold, gain momentum and get multiple colleges on board. However, they are living proof that given the right environment, an open source CMS-based approach to web site production can pay dividends for organisations that publish multiple department- or brand-level web sites.

---

## Religious Conversions

So, if all of this sounds like a steep hill to climb, what's the business case for change?

Aside from discussing the performance, budgetary and productivity gains mentioned above, you might like to start with an assessment of the fundamentals. For example:

- An audit. How much are you currently spending on page creation, training, and support?
- How satisfied are you with your current site's functionality and services?
- How satisfied are your users? (Poll them! Use a cheap tool like [SurveyMonkey!](#))
- Is Roger really a web designer? Or would he have more to contribute if he just focused on the content? (And, could he then make other important contributions elsewhere?)
- If you could save more money, could you use it to introduce new functionality/services?
- Is it your objective to become the best web coders or the best marketers/researchers/administrators/etc

No doubt, the answers to these questions are obvious and easy to find.

But let's make your choice even easier. Step back for a moment. What if all the energy that's currently devoted to maintaining all those little temples could be re-channelled into building better ones? And what if you could build a series of better tools in the process? And what if this work actually improved the quality of your sites and the experiences of your end users and staff? With the help of open source software and a good CMS this is all possible. What's required is not so much a religious conversion as an outbreak of common sense.

## About Squiz

Squiz is one of the world's leading open source software development companies. We give you control in a content-driven world. Our [open source CMS](#), MySource Matrix, helps leading organisations such as the UK's NHS and the Australian Federal Government to manage their content more efficiently and cost-effectively. It also helps top brands such as Future Publishing and Warner Music to sell more content and products to more customers via the web. Squiz is a privately owned company, founded in Sydney Australia in 1998. Squiz has an international network of offices in Sydney, Melbourne, Canberra, Hobart and London that provide a local service to hundreds of clients across a broad spectrum of industries. Our MySource Matrix [open source CMS](#) is internationally recognized as best in class (Gartner).

For further information about Squiz and/or MySource Matrix, contact Roger Warner on [info@squiz.co.uk](mailto:info@squiz.co.uk).