

CMS WISDOM REPORT VOLUME I

54 inspired tales of web content management trials, tribulations and tips
from the men and women who lived to tell their stories

Why CMS Wisdom Matters

Here at ISITE Design, we've been hip deep in web content management software projects for the past decade. As a national digital agency, we learned a long time ago that delivering great digital experiences depends on getting web content management right.

Several hundred CMS projects later (not to mention a few grey hairs) it's safe to say we've seen the good, bad and ugly when it comes to CMS.

We've seen software vendors and products come and go.

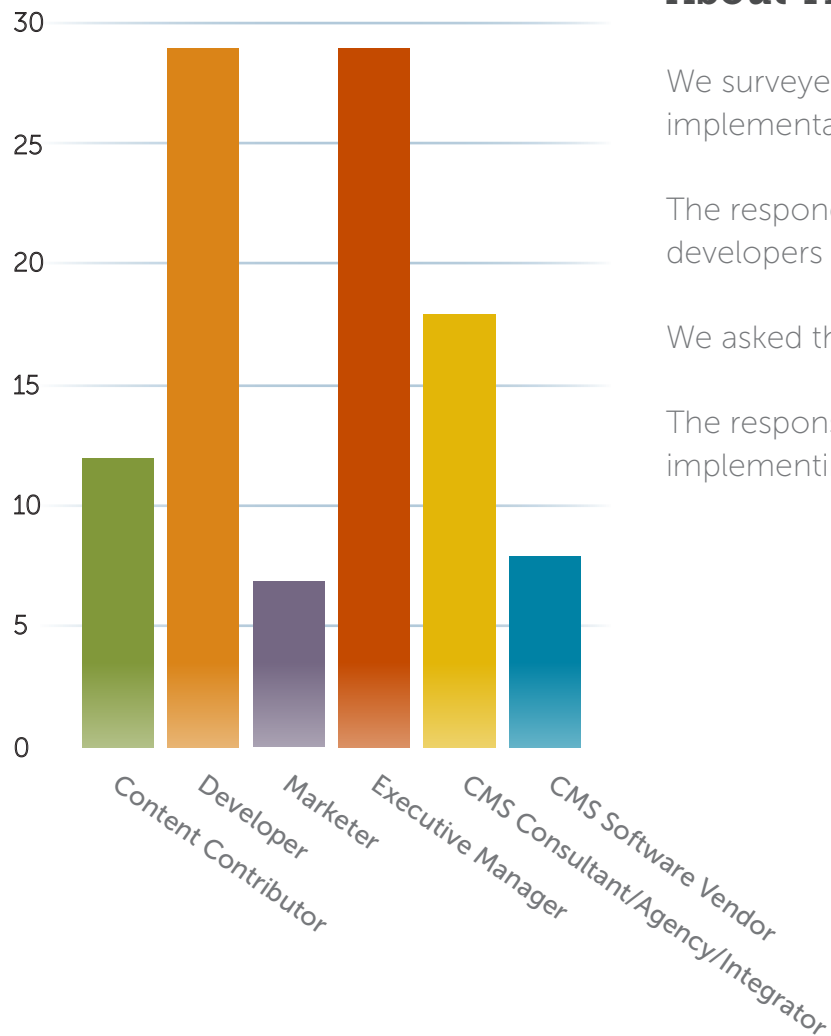
We've watched other providers start as saplings and grow into industry giants.

We've witnessed the evolution and vast improvements in web content management systems, from the early days of simple content editing tools to today's complex, marketing-centric platforms that drive user engagement and form the linchpin of corporate online strategies.

Everyone's got an opinion about their CMS -- marketing teams, content writers, web developers, IT teams. And everyone has lessons learned. So late this year, we put out the call asking for your CMS wisdom. Call it a humanitarian mission to help out others following in your footsteps.

Apparently we hit a nerve. Your comments expressed the true range of CMS emotions, from anger and frustration to elation and humor. We've compiled the responses into our first annual CMS Wisdom Report. We hope you enjoy the stories as much as we do.

About The Report



We surveyed more than 100 people who have been through real-life CMS implementations and are involved with managing the content post launch.

The respondents cut across all roles in a CMS project from the content users to developers to executive stakeholders

We asked them all one simple question: What's the biggest CMS lesson you've learned?

The responses should be helpful to anyone starting the process of selecting and implementing a new content management system.

54 Inspired Stories of CMS Wisdom

**“A content management system
is merely a tool.”**

Who's Driving The Bus?

Unless you pony up to public transportation, you probably chose your car. You, the driver. That wasn't a decision you left up to your mechanic or the salesperson selling you the car. Yet so many content managers and marketers let their IT department or design team choose their CMS.

When this happens, they're offloading the freedom to determine whether much of their work over the next couple of years--or however long until they reevaluate the contract--will be easy or difficult, whether their tools will empower their ideas or shackle their vision.

It's this idea of vision that's often lost when IT or the design team prescribe a CMS. One lesson I've encountered is that if your end product--say, a website--will include content, the best time to involve your content strategist is before you choose a CMS. Determine what's most important and the impact of those priorities on content, timelines, and technology. After establishing a message architecture to prioritize those goals and conducting a content audit to see what we have to use and repurpose. Get a better sense of who will be doing the work--and their level of expertise and comfort with the technology.

All those communication and people factors should drive the choice of CMS. Otherwise, you're just letting your mechanic pick your car and you might as well switch to a bus pass.

Margot Bloomstein
Appropriate, Inc.

Governance is good

A good governance plan is one of the most important success factors for CMS success.

- 1.** Identify an inclusive team.
- 2.** Start with “framing” decisions.
- 3.** Determine your deployment model.
- 4.** Define a clear vision.
- 5.** Identify roles and responsibilities.
- 6.** Develop guiding principles.
- 7.** Decide your organizational comfort level with social computing.
- 8.** Define policies and standards.
- 9.** Document the plan.
- 10.** Socialize and promote.

Susan Hanley

“Choose a CMS that will allow you to grow and migrate away from that CMS at a later date because eventually you will.”

“Looking back we wish we had known how long and expensive the process would be.”

With Great Content Comes Great Responsibility

Outlining attainable roles and responsibilities at the start of a CMS implementation will make or break your project. The key word in that concept is attainable - assigning work to someone without the capacity to complete it might look nice on paper, but will ultimately impact your project and make your fellow project team members very, very unhappy.

Plan, plan ahead, then plan some more. It's impossible to plan too much for a CMS implementation. We had three years leading up to our launch day, and yet, at 10 pm the night before launch a developer, a platform engineer and the vendor were scrounging for content on the old site to put on the new "About Us" page.

Content matters, but only as much as you have the resources to maintain it. There's no point in developing great content and training CMS editors if no one is going to maintain it. If people didn't have time to produce the content in the first place, chances are they're not going to return to it later. It took us two full years after the site launched to find an owner for our top-level institutional pages. The content has never been edited. We would likely have been better off with significantly fewer pages.

“If I were to do it over again, I would probably find a CMS with a much better WYSIWYG editor and file operations, as our users are not interested in any kind of markup - no matter how easy it is.”

Think Like A Product Manager

Aside from making sure to have a strong vision of what you are trying to accomplish with the CMS, the biggest lesson I've learned is the importance of product management for a large CMS implementation. Note that I said product management, not project or program management (also important). In other words, someone needs to manage the website (and CMS implementation) as if it's a product whose overall quality must be maintained.

So, plan for product management of your CMS implementation from the start, so that you reduce unnecessary backend complexity, increase the quality for the site visitor, and reduce internal problems by having a clear process for involvement by the stakeholders.

David Hobbs
Author of Web Site Migration Handbook

“The higher-ups love the concept of a CMS because it promises to do all things for all people, yet nobody ever plans for the long-term support and maintenance of the tool.”

More People. Less Technology.

With our homegrown CMS, we found that lowering the barrier to entry with a simple, user-friendly interface was still not the answer, because we're asking people with zero web skills to suddenly know they shouldn't post that 3,000 pixel wide image from their digital camera and resize it in the HTML.

In our case, launching in the expensive CMS was a long, drawn out hand-holding process. Launching in the homegrown CMS was a 1 hour training session and setting them off on their own (and not caring that they posted the 3,000 pixel photo). The reason it's "not working" today is not the CMS– it's not having the support staff to help clients with building effective and user-friendly websites.

“Open source is far from free. If you choose an open source solution, be prepared to spend a lot of time and/or money getting it implemented.”

“Talk to people who are actively using any CMS you’re considering, both from administrative and user perspectives. Don’t rely on a sales pitch from the vendor.”

Beware Of Ghost Towns

To properly gauge the level of community and support provided by a CMS vendor, go to their forums and check how many questions get left unanswered.

The main developer forum for the CMS we use regularly has, as of the time of writing, 13 posts on its front page with zero replies. That's 13 questions their support crew haven't even been polite enough to comment on. It can make a man feel awful lonely.

Budget For Success

When deciding between different CMS systems, be sure to look at the whole solution. That sounds obvious but too often, cost becomes the overriding factor and what you end up with is not the best product for your needs. This has happened to me twice. You really do get what you pay for. Just because one choice might entail the lowest initial expense doesn't mean the overall cost is better.

You might be surprised to find how much time and/or money gets spent in the end to customize and hack the cheap solution to do what you want/need it do to. Ask what kind of ongoing support is included and what that entails. Are you going to get charged for every 10 minute phone call post-launch? Find out now. Make sure the people who are holding the purse strings are there for all of it. I know they usually don't have the time or inclination to sit through things that detailed, but it's too easy to pick the cheapest solution when you don't see the full picture. Have people who will use it every day be involved in the decision making process too and let their voices be heard.

Look Before You Leap

We're in the process of implementing our first CMS. We're finding lots of functionality that doesn't work as smoothly as the demos led us to believe, hidden costs, and some oddball administrative functions. It has taken months longer than we anticipated and we're still not close to a launch date.

We realized after taking a five-day training course that what we had bought was more of an XML editing environment instead of a full-blown CMS. I hadn't brought the right staff to learn their APIs, which we were told we'd probably want to dig into to build our own admin functions necessary to manage a site of our size. None of this was brought up in the sales demos (or the reference calls with other customers).

“Don’t rely on the technology to make the rollout of a good CMS work for you. Whatever the technology, if you don’t pay attention to the business problems you are solving, the processes required and how the people are to be trained in using the system, it will fail and cost more money than it should to implement.”

**“A bad implementation of a good CMS
is a bad solution.”**

Partner With Caution

The partner(s) you choose will be your most important asset, or the source of your premature grey - so do your homework. Here are two areas that may help you prevent a trip to the chemist:

Check the references they provide and look for people/companies they worked with not on the reference list - there's a reason that reference list is created by their marketing team. Instead, think Yelp. Read, or in this case listen, to all the reviews and then balance what you know and what the past clients told you to make a more informed decision.

Who's on your team once you sign? Ask, and then spend time with them if possible. Do you mesh? Do you feel like you are suddenly fluent in Mandarin Chinese and they don't understand you/your question? How quickly do they respond to your inquiries? Their ability to work well with you and your team, as well as their ability to communicate proactively, is key.

Jessica Winston
MITX

Sweat the Small Content

Developing good content is the thing that will make or break your site. It's called a content management system, not a content generation system, and there are still a lot of people out there who still don't quite get this.

Every piece of content for your site, from the home page copy to what shows up in the sidebar on the Contact Us page is important, and the more content you develop up front the better your CMS implementation is going to be. I've seen too many organizations that focus too much on the page copy and not enough on everything else and then are disappointed when every page on their site looks the same.

“There is no such thing as a ‘one-size-fits-all’ CMS. Even in the best of situations, you will need a handful of customizations, and in most cases you will need more than you expect.”

Take It Step By Step

Overcoming paralysis is the big first step in the implementation process. I was able to do this in a couple of steps:

First, I clearly defined what our institution needed, and what it could afford. Given those parameters I created a server sandbox that approximated our server environment and installed and tried out different open source CMS options. I assigned myself the simple task of doing the installation and templating a page in our graphics. If I could do it, it stayed in the running. I was going to have to teach people to use it; if I didn't get it, nobody else would...

After understanding how the CMS worked, the next important thing was confirming the stability of the product. History, active development community and frequent upgrades are good indicators of a product's long-term viability. I opted for a soft roll out. The CMS I chose did not require abandoning the mash-up of static pages and hand-built applications. I moved our institution into it one department at a time, building a user base and enthusiasm for the product with the failsafe of being able to pull the plug on it if people hated it. They didn't. It took over a year to move everything but we did it.

Jonathan Pierce
Linfield College

Your Website Is Not A Project

There needs to be a post-launch maintenance plan in place after the site goes live. No matter how many rounds of testing you go through, there is no substitute for actually working with the product in the real world.

It's better to acknowledge that issues will come up in the initial rollout and plan for them, than to struggle with a product and have no documented plan of action.

“Your boss has to have an understanding of the challenges of CMS and the web as a communications tool for the organization. It must be treated as a priority.”

“Keep in contact with the CMS users. Placing the right content, using the right images and keeping in mind the initial design/ideas/strategies is not something that users think about all the time. I’ve seen great designs (both visually and in usability) destroyed this way.”

**“A CMS has to work the way users
already work and think.”**

Avoid CMS Sprawl

After a few years of CMS-based development, I've learned that if solution is going to be updated frequently, it makes sense to break it into the small pieces that can be easily dropped and rewritten from scratch. Business users don't understand how their "small requirements update" will affect architecture; be ready to change it on the fly.

After some ups and downs with current and past CMS solutions, our team has found that the most important key for success is to put the right people in charge of the project. If non-CMS users are running the show, you are just asking for trouble. Also, although 'customized' and even 'homegrown' may sound appealing, we've found (the hard way again!) that sometimes 'out of the box with personalization' is the key.

“Always make sure to have a content plan before doing any evaluation. If you are just looking at pretty wireframes and visuals, but not thinking about workflow, syndication, skill of people and entering content, then you are going to fail.”

Avoid Development Hacks (Where Possible)

Beware the lure of adding CMS add-ons/plugins/extensions/modules!

As a developer, it's easy to become tempted to add non-critical functionality or interactivity to your website via CMS add-ons. And even though I shouldn't, I still use more add-ons than are necessary. They nearly all work smoothly and as advertised. And it's easy to get carried away while in the rush of development.

The "gotcha" invariably will come later -- usually when updating the core CMS or the add-on. In a recent case, I nonchalantly updated an add-on for a live site, only to discover that the way I secured the site with SSL combined with changes made by the add-on developer caused a critical search function in the site's admin control panel to stop working. And rolling back was a pain. The new version of the add-on had made changes to the structure of the database too. Ugh. Live and learn (eventually).

Build the Business Case

Convince the people that are paying the bills, not the techies. Upper management should understand the strategy first, before any procurement decisions are made. They'll feel less like they're being sold something they don't need.

Remember the Care and Feeding

Clients/customers/businesses don't think about the day-to-day maintenance required for most content management systems. Are you going to have user-generated comments on your website? Profiles? Who is going to moderate that content? Who is going to respond to all of those comments? What about future security updates to your CMS implementation? Who will install those, and how much will it cost you? How will you handle spam?

A lot of times these community features can be popular, but are poorly implemented at best -- and ignored once they are.

“No matter how many times you train a content contributor, if they only use the CMS once in a while, you’ll need to retrain.”

Commit To Your Platform

I've been through three Enterprise CMS implementations. None was perfect or without issues. CMS selection is like an artist choosing a medium, and committing to that medium for an entire period. You have to pick your canvas dimensions, a material, perhaps a subject -- and (unlike the art world), you have to stick with it, despite not knowing what future inspiration may lie ahead. Inevitably you'll reach a point where you want to do something different... And it's at that point when you'll feel the full pain of your previous decisions.

Every CMS operates under a certain paradigm. Some are more flexible than others. None are infinitely flexible. The biggest lessons I've learned are:

- 1.** Whenever possible, stick with the CMS paradigm you've chosen. If your CMS assumes a certain layout, don't go against the CMS's assumptions without really good rationale.
- 2.** Avoid non-critical customization! Work within a CMS's framework. You'll thank yourself when it's time to upgrade or migrate later.
- 3.** Always be aware of the ROI. A CMS should save you time/effort/money. If it's not doing that, don't use it. Don't fight with your CMS. You'll always eventually have to upgrade or replace your CMS, so there is no perfect solution, only ones that provide positive return. Success is measured by the level of effort saved and the amount of work accomplished relative to the level of effort applied.

“Get help from someone who has actually installed the CMS for similar clients and environments in the past. This is rarely, if ever, your IT department or the CMS vendor itself. There’s nothing worse than a CMS that is improperly configured.”

Content. Content. Content.

I worked as a developer for a vendor for several years. I developed the proprietary CMS, and I often was involved with its implementation and launch for the clients.

Every project that was delayed or wasn't launched was due to issues with content.

The client often didn't have a content strategy, which then translated to a lack of content when the deadline approached. Content needs to be the central focus to a CMS strategy. Without content, and a coordinated strategy, the most powerful CMS will do nothing for you.

“Don’t let the IT team take over! The CMS will mostly be used by people who aren’t tech wizards and who aren’t interested in the technical process.”

Teaching An Old CMS New Tricks

Imagine working in an experienced company, running dozens of websites on a mature, individually tweaked CMS. Probably a typical scenario and one might describe it as a perfect environment, since your people are used to the system, have in-depth knowledge and the system itself is just a mature and reliable piece of software.

Well, actually that usually turns out to be a nightmare.

A mature system implies (to the management) that you don't need to add any new features, because everything is already there. Or at least it's so plain easy to implement, because it's a mature system. In reality you need a well-designed software architecture to result in a highly customizable/extendible CMS. Unfortunately, that would require some latest-technology magic - what a pity that you won't find "latest technology" in a year old "mature" system.

So the biggest lesson: So the biggest lesson: Never ever start thinking something like, "now our system is mature, let's go on with something else and never touch this running system." Technology changes. CM systems change. Requirements change. The only mature thing in your ecosystem is your content. I wonder how many companies ever thought about migrations (or just the export of their contents)?

Know Your Users

Something I'm constantly learning and relearning is that no matter how thorough I believe my CMS manual is, it can always be better. It's taken me a long time to realize just how people process instructional information. With this in mind, we've actually taken steps backward during the training process, to simplify the CMS even more than it had been.

Another frustration is constantly being surrounded by testers (in our own organization) who are so computer proficient that they provide no accurate baseline for how our manuals and programs will be received by our users.

Getting More Bang For Your Buck

My biggest lesson is certainly never prejudge how the users of a CMS will use it!

I have been amazed in different ways: How users can be creative and can leverage and take the best of a solution to go beyond what it is supposed to deliver. (For instance, I am remembering how people could extend simple blogging systems to make much more than blogs, or, in my eZ Publish experience, how people could do online applications that are way more than the usual eZ Publish use case.)

Another way I am constantly surprised is when I simply look at how people use the CMS for very basic things. Often, they use it in a different way than we might think or teach, sometimes for the best, sometimes for the worse... hence the relevancy of often doing user research and panels to adjust both the product and also the way you teach it!

Roland Benedettie
Z Systems

“Focus on a specific business context to demonstrate real business value. Most people don’t understand ECM/WCM as it’s a broad-reaching term. You need to demonstrate business value; success is relative to each organization.”

Content Is (Still) King

When we're selecting a new CMS, we debate about workflow, security, publishing, collaboration, search and on and on. Everyone has an opinion.

But once the platform's in place, it comes down to content. If the content's well-written, concise and helpful (who knew?!) it wins, regardless of the CMS.

After the dust has settled on the CMS debate, finding people who can contribute meaningful content becomes extremely difficult.

Beware Of Automated Migrations

Content migration is a real pain. Every CMS-implementation has its own peculiarities (e.g. the way content and properties are encoded, the structure of the content types, links between pieces of content).

In a migration, you are looking at two (CMS) systems and trying to build a bridge between them, but one that you need only once (or twice). So you have to decide whether you invest in building an automated solution vs. using 'cheap' manual labor.

In many cases we've found that doing it manually is often the fastest, cheapest and better solution. Better, because usually some manual tweaking is necessary anyway.

“Prior to purchasing a CMS, get the system in the hands of the people that will be using it. Our first system was selected by a consultant and an executive, without any further campus buy-in. People had to struggle for years with an expensive product that didn’t do what they wanted it to do.”

“Write a scenario-based requirements document for a CMS before any vendors are on the horizon.”

Don't Underestimate The Effort

CMS is a lot of work and more complex than most people imagine.

We've had problems getting the infrastructure in place, getting the ITS teams on board, and getting them to understand how a CMS works and why it's different from other applications. We've also had problems getting the web production organization to understand how much change a CMS means to them, and the degree to which their daily processes would have to change.

I wish we had conducted a content clean-up prior to the migration so we would have migrated less, and I wish our CMS consultants and developers had injected more "content management" into the process. Their focus has strictly been on the development aspects.

It's (Not) Like Riding A Bike

It's not enough to train people and document everything, you need to incentivize them to use the CMS frequently. We found that both the carrot and the stick works.

The carrot: Reward frequent users with previews of new features and the opportunity to provide input. If you want to take it to the next level you can offer points or badges (think Foursquare) for frequent use.

The stick: Deactivate old/stale content. If content hasn't been checked or updated for 'x' months, an administrator gets an email, and within 10 days the block is deactivated.

Use it or lose it – that's what it's all about. If your content managers don't use the system, everyone suffers (especially your website). Find creative ways to make them feel invested in checking in with the system, even in minor ways, on a frequent basis. Over time they will take ownership and eventually show YOU how to make the system even better.

Start with the Basics

One of the supposed benefits of a CMS is to allow staff people to manage their department's content with very little training. While that's true, there are a lot of "ifs," "ands," and "buts" behind that.

For starters, you might assume that people who use a computer every day can actually grasp basic concepts like shifting their thinking away from web "pages" (like they used to build in FrontPage and FTP to the server) towards thinking about content as records in a database. You'd be wrong.

You also might think they would understand that supporting files like images or PDFs need to be uploaded into your system instead of having everything in one unit like a Word document. You might also think that when the department manager designates someone as their "web person" that staffer would know how to write a little and take a decent photograph. Maybe that will be true, if you are lucky.

So, when you train people to use the system, you might need to start with some very basic concepts of how websites work and how to produce good content. Then they are ready for the CMS. Quality control will be important, and if you don't have someone knowledgeable editing and approving content, your quality will go down. Depending on your organization's culture, you might get a lot of push-back on these issues, so be ready for that.

Think Strategically

Several years ago we were engaged in a project that included migrating a web portal from a simple, custom web system to an open source CMS. In that process we wanted to unify the look and feel of all pages like articles, categories, forum topics, etc., which would lead to better user experience, as the old site was pretty heterogeneous and confusing.

We ended up pixel-pushing to exactly the same look as the old site, because the client wanted the site to be exactly the same as before. All in all, a very bad and time-consuming experience. We didn't make that mistake again.

Power to the WYSIWYG Editor

The text editor used by the people doing the routine maintenance and updates on web pages has to be easy to use. If the text editor does not give the editor all of the control they need, is too hard to learn, or is too easy to make mistakes, it creates more work for the CMS administrators.

Cultivate Internal Experts

Whatever system you have, you need at least one person to become an expert in it. Preferably with development experience, this person learns the technical details and acts as a resource when issues emerge (as they invariably do).

The expert can also modify the system (if it is open source) and assess third-party plug-ins/add-ons. If the system needs updates, this person can manage them and keep the site(s) secure. Without that, you are at the mercy of suppliers and consultants, and you won't have the flexibility to do things you really want to (or the knowledge to know what things are possible).

Test early. Test often.

Often the site owners do not consider an incremental testing of new ideas, they just jump right into an overblown overhaul. The result is often that as problems arise, which they will, massive time and energy is spent on troubleshooting all at once, not in digestible amounts and certainly not with any learning curve to help.

Technology Cannot Fix Organizational Problems

We are in the midst of a much-needed CMS adoption. Our project plan has kept us moving and, for the most part, the implementation is going well. The project involves moving content from multiple Web sites into a single CMS. Each of the original Web sites was operated by a distinct organizational unit, often with their own local Web developers. These silos had largely operated in isolation.

The project plan sought to avoid the potential political battles by starting with one silo, absorbing it, then moving on to the next. The layers of complexity raised by taxonomies and content organization within a modern CMS make this approach difficult.

If we had another crack at this, we would have spent more time at the start and had a single absorption of all content into the new CMS, taking into account the differences at the start.

Let The Users In Early

When implementing a CMS, developers and project managers tend to focus on getting the site up and running. What most forget about or give little time to is the actual roll-out of the CMS to the content editors and contributors. If this is overlooked, a CMS-driven site can become just as stale and difficult to update as it was before.

The editors and contributors need to be involved with and exposed to the CMS user interface much earlier than they typically are.

Get The Right Buy In

When selecting your CMS for a college or university (or other organizations for that matter), get buy-in from the groups you hope will use the CMS in the future, not just the department/school/administrative group that will be the initial user.

This gives you clientele who will want to know when they can get on board. It generates a strong positive push for growth of the client base and provides an argument for why you need to provide user training/support at a level that will support your future needs.

“There is no CMS that will do everything you want, as they all have strengths and weaknesses. You will love it for X but hate it for Y. That’s life.”

About ISITE Design

ISITE Design is a digital agency and CMS consultancy founded in 1997. The agency helps organizations plan, develop and execute digital initiatives with a focus on driving measurable outcomes. ISITE Design combines its nationally recognized web content management practice with expertise in digital strategy, user experience and analytics. The agency also publishes the popular CMS Myth blog and speaks at national conferences including Gilbane, J.Boye and Confab.

Portland, OR

115 NW First Avenue, Suite 500
Portland, OR 97209
888.269.9103

Cambridge, MA

One Broadway, 14th Floor
Cambridge, MA 02142
617.391.0166

ISITE Design

www.isitedesign.com

CMS Myth Blog

www.cmsmyth.com