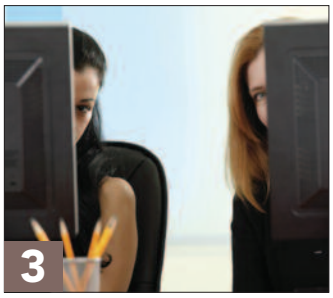
Follow-up to
Best Practices for
Developing a Web Site

Developing a Content Management System Strategy

an internet.com Project Management eBook

contents

Developing a Content Management System Strategy



Paul Chin (www.paulchinonline.com) is a freelance writer and journalist. He has previously worked in the aerospace and competitive intelligence industries as a software developer and CMS specialist. He currently writes on a wide range of IT topics, including systems development and security, digital communications and media, content management and web design.

Developing a Content Management System Strategy

By Paul Chin

Timely and accurate information forms the basis of everything we do. It provides us with the proper knowledge to make the right decisions at the right time. Anything short of this and you might as well try your luck in Vegas.

But the bulk of an organization's knowledge and content is scattered throughout many departments, workgroups, and employees. Information is stored in departmental file servers, personal computers, filing cabinets, and desks where only small groups of people have access to it.

Jupiterimages

So how do you make the collective knowledge of an organization's employees readily available to those who rely on this information for their day-to-day work? How do you organize and manage all this information while maintaining a high degree of accuracy and a low degree of content duplication? You do it with a content management system (CMS).

The goal of a CMS is to allow users to tap into an organization's pool of knowledge in a centralized environment with a distributed model of content management. Although technology is used to build a CMS, the system must be based on more than the sum of its bits and bytes. A truly successful CMS must be built on three equally important components that need to be addressed in the following order:

1. **Culture:** Understand the collective mindset of your organization and designing your CMS with this culture in mind.

2. **Process:** Identify your organization's content and determining how it will be managed.

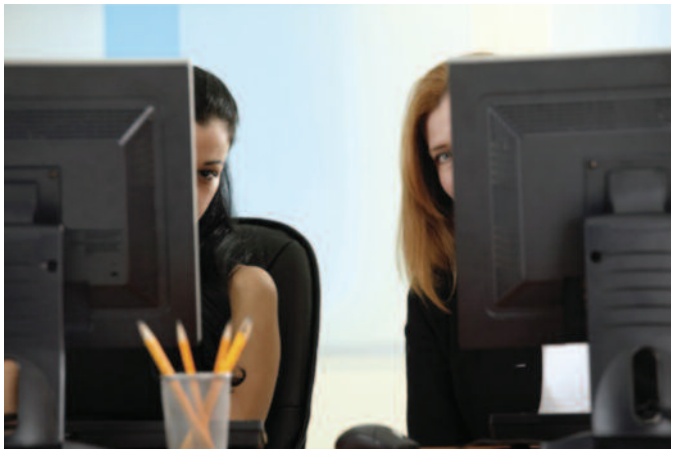
3. **Technology:** Choose the right tool and technology for the right job.

"Information is stored in departmental file servers, personal computers, filing cabinets, and desks where only small groups of people have access to it."

Culture's Impact on a CMS

Will a CMS Work in Your Organization?

One of the fundamental rules of system development is to build an application to accommodate the needs of an organization's user community, but we can't discount how this community and its culture affects and shapes the outcome of the system. If you ignore these effects, your system might end up looking very different from what you originally planned—if it survives at all.



Jupiterimages

Technology isn't the panacea software vendors would have you believe. Makers of commercial CMS suites tout their products' abilities to bridge geographically dispersed employees and create a virtual environment free of physical boundaries and time constraints. The technology allows you to do this, but will your employees?

One of the primary causes of CMS failure is trying to inject a technology-based solution into a corporate environment that's unwilling or unable to collaborate and communicate at the most basic human level. Employees that work with blinders on and feel territorial about their "personal space" aren't likely to work well with others to begin with. No amount of technology will solve what is fundamentally a human problem.

Technology is meant to augment our innate ability to carry out certain tasks; it's not meant as an outright replacement. Organizations that harbor negative and counterproductive attitudes aren't conducive to teamwork at a basic interpersonal level.

This problem can't be covered up with technology. In fact, adding an extra layer of complexity can compound the problem, furthering the disconnect between various departments, workgroups, and employees.

You must understand the collective mentality and culture of your organization's users, and how they affect the outcome of a technology-based CMS. These influencing factors will determine the extent and the quality of users' involvement in the project. A successful technology-based CMS should be a digital reflection of an organization's underlying abilities to communicate and collaborate at the human level.

"Organizations that harbor negative and counterproductive attitudes aren't conducive to teamwork at a basic interpersonal level."

Controllable Factors

CMS developers must gain an understanding of their organization's culture and the users within it. How do users currently operate? What needs aren't being met with current methods? What are their work habits? Are users conducive to technology-based solutions? How can technology fill the gap? What are your users' professional backgrounds? Answering these questions can help minimize the negative influences of culture on your system.

When you have a firm understanding of your organization's culture, you have a certain amount of control over how this culture can affect the functionality and design of your CMS:

- **Functionality:** Corporate culture can dictate what features your CMS will offer its users. If your culture relies heavily on teamwork, you might consider including online collaboration tools. But if your culture is based more on independent and self-reliant users, you might consider offering online tools that allow them to query databases and search for, or create, information on their own without the need to go through a third party.

- **Design:** A corporate culture ingrained in technology will gain more from a no-nonsense CMS based on functionality and less on flash, while a culture less technically inclined might prefer one that's simpler and more user-friendly.

Uncontrollable Factors

Corporate culture will always be an unknown element in system development. You can conduct interviews with focus groups and prospective users to gain an understanding of their needs and how they work, but there will always be uncontrollable factors that can kill a CMS.

The main purpose of a CMS is to share and manage information, but every organization will encounter key information holders who refuse—actively or passively—to participate. They may hoard information for various reasons: job security, to gain professional advantage over colleagues, to use information to boost one's own status, fear of non-recognition, fear of accountability, holding information for ransom, harboring loner attitudes, or just plain being difficult for the sake of being difficult.

There are some things you can do to foster a spirit of cooperation and encourage users to share information:

- Approach known information hoarders in a diplomatic and non-confrontational manner, and ask them what you can do to make it easier for them to participate. It's important you don't confront unwilling employees with a "do it or else" ultimatum. This will alienate them further and cause resentment towards the project and all those involved with it.

- Implement a peer-to-peer support center for your knowledge community so that individual content holders won't feel as though they're alone in their efforts. This also provides them with someone to turn to if they need help with a problem.

- Put a "face" to the knowledge community by profiling key contributors. This will go a long way towards humanizing the system, and will allow employees to get to know the people behind the information.

- Create a mentoring program so newcomers don't feel estranged from the established knowledge community. Let them know that they can rely on knowledge veterans for help. These newcomers can potentially become mentors themselves in the future.

- Lead by example. Content holders will be more likely to share if they see others sharing as well. This will create a rippling effect: The more you share, the more others will share.

- Acknowledge contributions so that active content holders have a sense of recognition, and that they're being appreciated for the efforts.

- Implement a contributors "Hall of Fame" or "Contributor of the Month" to highlight workers who go above-and-beyond. ■

The Content Management Process

Although content management is the overall function of a CMS, it also serves a different company-specific purpose within every organization. It's the latter that will connect with users.

You must define a mission statement—your system's raison d'être—that's applicable to users' real, day-to-day lives. Users are more likely to adopt and use a system if they can directly relate to its purpose, more so than if a CMS had a broad and general purpose such as "to improve corporate collaboration" or "to store and disseminate information."

Jupiterimages

A CMS's mission statement forms the basis of development and will dictate how your CMS will be used to support specific business processes. But it's important to note that CMS development, although Web-based, is very different than Web site design. This is why it's not a good idea to give the responsibility of building a CMS to a pure designer with little development experience. You need to establish business process, functionality, content types and taxonomy, and user input before considering layout, typography, and color.

Identify Your Content

A CMS is a container for information. Before you can design the size and shape of this container, you need to fully understand what type of information is going to be stored, how to get it inside, and how to serve it up.

Internal Knowledge Assets

Internal knowledge assets (IKA) are an organization's intellectual property. IKA are made up of information that isn't readily available to the general public and is created through the efforts of an organization's user community. IKA can include business strategies, market trend analyses, information on competitors, financial information, internal memos, and details on specific projects and contracts.

Unlike externally sourced information, which can be somewhat vague and general, IKA are highly focused and stripped of a lot of the marketing and public relations spin that's often added for public consumption.

Externally Sourced Information

Externally sourced information (ESI) includes any information that can be legally obtained through a public

"You must define a mission statement—your system's raison d'être—that's applicable to users' real, day-to-day lives."

medium or purchased from a third-party information vendor. ESI can include information collected from the Internet, hard copy documents from newspapers and magazines, and information from content delivery services.

ESI's greatest advantage is volume. Since it doesn't need to be created internally, content managers can put together more information in a shorter period of time. This, of course, can be both a blessing and a curse. If content managers aren't careful about their selection and have a loose vetting process, ESI can easily lead to content clutter. There's also an inherent risk in using ESI if content managers ignore the copyright laws associated with the information they collect from external sources.

Content Engineering

Content engineering is the process of filtering raw data into more usable and applicable information. For example, a lengthy 50-page annual financial report can be pared down to a three-page summary.

Engineered content, therefore, is extremely focused and specific to a discipline or topic. But this isn't to say that non-engineered content doesn't have its place within an organization. Non-engineered content covers a wider range of topics and is most useful when users aren't looking for anything in particular.

By definition IKA are engineered because they were created by members of the organization. ESI, however, can be stored on a CMS in its entirety (if you have the copyright to do so) or engineered so that only applicable portions are maintained.

Determine Content Organization

Content organization is highly dependant on the system's purpose, the technology used, security for content managers and user community, portability and flexibility, and navigation and usability.

There are two levels of content organization that must be taken into account:

	Advantages	Disadvantages
Engineered	More applicable and targeted to your organization's industry and business processes.	Information may only address a small group of users within the organization.
	Takes less time for users to read since the information has been pared down to the essentials.	Requires much more work before information makes its way onto the CMS.
	Highly focused on a specific topic, stripped of marketing or PR spin.	Information might not be as timely.
Non-engineered	Caters to content browsing. Useful for when users aren't looking for anything in particular.	Users must sift through a lot of non-applicable information to find specific content.
	Very little human intervention is required.	Content providers might be tempted to take shortcuts by inputting content without reviewing its relevance.
	Information is posted much quicker, and can be more up-to-date, since no processing is done.	Non-engineered content can easily lead to CMS clutter.

Physical Content Organization
Physical content organization defines the physical organization, or architecture, of a CMS. It exists either in a database or in a series of tiered folders and files. Efficient physical content organization contributes to overall system performance and eases site maintenance for the technical staff.

Logical Content Organization
Logical organization defines the contextual relationship of one piece of content with another and usually (but not always) defines the system's navigation type. Efficient logical content organization contributes to overall user experience and allows users to navigate from one piece of content to another in an intuitive manner.

Establish CMS Management and Governance

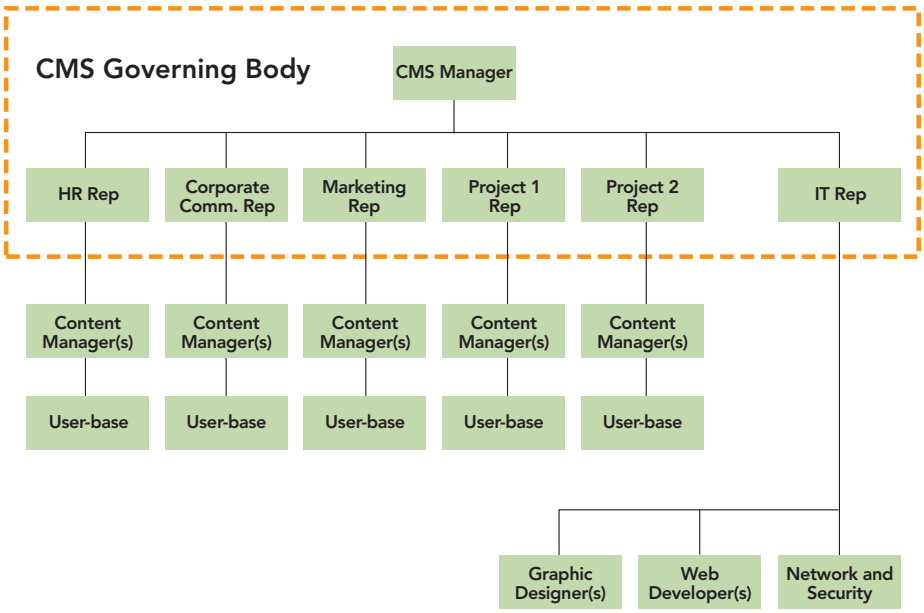
A CMS, unlike many other technology-based systems, involves the ongoing participation of many professionals from many disciplines, departments, and workgroups. With so many people dipping their hands in the pot, it's easy for things to get out of control if you're not careful.

Ownership and governance of an organization-wide CMS must be structured in a manner as to allow all stakeholders to have equal say in the direction of the system while avoiding the pitfalls of a bloated bureaucratic process.

Multi-tier governance models
The goal of a multi-tiered governance model is to give every CMS stakeholder a voice; and to make a diverse set of resources—technology, content, business process, and personnel—operate as a seamless unit.

The biggest CMS governance mistake is to appoint a single department as the sole governing body of the system. This can open up the possibility of partisanship since every CMS section owner has their own priorities and objectives. They might end up, consciously or subconsciously, undermining the efforts of other section owners.

Instead, you need to form a governing body comprising all top-level CMS stakeholders—representatives of each core section and key technology personnel. The goal of this governing body is to ensure every section is represented, that the system's mission statement is upheld, and to prevent any one group from hijacking the site for their own purposes.



Example of a multi-tier CMS governing body.

> "The biggest CMS governance mistake is to appoint a single department as the sole governing body of the system."

A top-level CMS manager should also be appointed to coordinate the efforts and activities of all governing members. It's crucial that this CMS manager be independent and non-partisan, and open to all ideas and suggestions brought forth by members. The CMS manager keeps both the system and its management team flowing freely, acting as an administrator, mediator, and key decision-maker.

The CMS manager needs to coordinate and oversee three distinct components:

1. **Infrastructure:** Infrastructure includes all technology components of the system such as software, hardware, design and development, and security and fail-safe mechanisms. Infrastructure is the responsibility IT.

2. **Content:** Content management falls under each individual section owner and encompasses all matters of intellectual property and business process.

3. **Administration:** Administration involves the overall management of the CMS. It includes issues of policy, decision making, standardization, and future direction of the system. All top-level stakeholders should participate in CMS administration. ■

You need to implement a multi-tier governance model to:

- Determine and regulate development and technology standards
- Determine and maintain content presentation standards
- Promote fairness and equity among all CMS stakeholders
- Act as peacekeeper, resolving conflicting goals between different section owners;
- Ensure that no one special interest group hijacks the system for their own partisan purposes
- Determine the future direction of the system
- Maintain the overall integrity of the system

Identify CMS Governing Body Personnel

	Name	Responsible for:
CMS Manager		
TECHNOLOGY		
Designer(s)		
Developer(s)		
Infrastructure (network & security)		
CONTENT		
Rep for <section 1>		
Content manager(s)		
Rep for <section 2>		
Content manager(s)		
Rep for <section 3>		
Content manager(s)		
Rep for <section 4>		
Content manager(s)		
Rep for <section 5>		
Content manager(s)		
Rep for <section 6>		
Content manager(s)		
Rep for <section 7>		
Content manager(s)		
Rep for <section 8>		
Content manager(s)		

CMS Technology

If your organization doesn't have dedicated IT staff then this is going to be your biggest question mark. Deciding on a technology, or set of technologies (often called a "solution stack"), has a big impact on the longevity of your CMS.

Sometimes you're forced to compromise what you'd like to do with what you're able to do based on available personnel, technical resources and expertise, financial resources, and allotted timeframe.

Jupiterimages

The most likely initial question you'll ask is whether to develop the system yourself or hire someone to do it for you. This subject is covered in the eBook "Best Practices for Managing a Web Site" (http://www.devx.com/ebook/Article/37584), and so it won't be repeated here.

Before you settle on your technology, take stock of what IT infrastructure you already have in place and how your potential technology candidates will affect it. If your organization is based entirely on a Microsoft backbone, do you really want to introduce a LAMP environment?

Whichever technology you decide on, you need to think about the longevity of your CMS. You must plan not only for your immediate needs, but also your future needs.

Plan for the Future

CMS team members need to adapt their system to accommodate changes within their organization. It's not a question of "if" but "when." Your CMS must be flexible and extensible enough to adapt to these changes without having to rebuild large sections of the system—or worse, the entire backbone. You can increase your chances of surviving changes to your CMS if you:

Build an Extensible Infrastructure

Shortsightedness in technology is a system killer. You might be tempted to save time, effort, and money by buying and building only what is required to meet your immediate needs. But what happens if departments or workgroups who were not part of the initial CMS deployment decide they want in? What happens if employees on the road or satellite branches require off-site access? When this happens there will be a mad dash to boost the resources of a CMS that seems to be

"Whichever technology you decide on, you need to think about the longevity of your CMS. You must plan not only for your immediate needs, but also your future needs."

shrinking while requirements are expanding. It's crucial to plan your CMS for the eventuality that it will grow beyond your current infrastructure.

Implement a Modular Structure
A CMS, like the organization that owns it, is an evolving entity. An organization can change quickly, and as such, its CMS must quickly adapt to those changes. A CMS should be modular and its sections fairly self-contained. This will allow an entire branch of the system to be added or severed and migrated to a new place with minimal effort and redevelopment.

A modular CMS helps promote system longevity and eases management, enabling developers to adapt the system to future or unexpected changes. A CMS shouldn't be so tightly woven that removal or addition of a single component will cause the system to collapse like a house of cards.

Use Industry Standards or Widely Accepted Software
Your choice of CMS technology—both the tool itself and its underlying backbone—can greatly affect the longevity of your system. With the speed at which technology changes you might one day find yourself trying to support a system that's based on obsolete technology or using orphaned software—software developed by a company no longer in business or software that's been dropped by the company that built it.

While it's impossible to predict the direction of technology or software companies, you can increase the life of your CMS by using industry standard technologies or widely accepted software with a large user base.

In addition to system longevity, you also increase your chances of finding knowledgeable developers. The more people who use a particular software product or technology the easier it will be to find people to support your CMS. Otherwise, you'll be forced to pay exorbitant amounts of money to hire niche programmers because they're your only option.

Free Open Source vs. Commercial Software?

Even though open source software is free, flexible, and highly customizable, it's going to be a difficult sell to senior management—especially in larger organizations where establishment and toeing the line is the norm. And despite all the advances in open source software, many still perceive it as being "not serious" and used exclusively by small groups of freewheeling techies who have no business sense. Free or not, senior management is much more likely to pour money into an established tool from one of the "Big Guys."

Proprietary, off-the-shelf software is the established and accepted norm in the big business world. But huge software markups can exclude small businesses or not-for-profit organizations with very limited IT budgets. Once you do invest in a commercial software tool—tying your organization to the vendor and tool—you'll be subjected to the rolling costs associated with owning proprietary, commercial software.

Five reasons to consider a DIY open source solution

1) Decide Your Own Software Fate
Open source software, licensed under a free software license such as the GNU General Public License (GPL), represents ultimate freedom. It represents flexibility and control over customization and distribution, minimizing vendor dependency and lock-in, avoiding restrictive and costly licensing agreements, and reducing total cost of ownership.

2) Reduce Total Cost of Ownership
Free, open source software doesn't cost you anything upfront. User documentation is provided online or via downloadable PDFs, and there's plenty of support via community-based discussion forums. You can download and experiment with the fully functional open source software before committing to it, and it won't cost you anything except time.

Commercial, proprietary software forces you to incur the rolling costs of ownership. In some cases, depending on your licensing agreement, you'll still be paying for the proprietary software long after your system reaches the end of its life cycle.

3) Total Software Customization
Open source software solutions allow you to fine-tune your software to your organization's business-specific needs. This gives you a much more tailored solution, unlike off-the-shelf software that often only contains the most common and widely used features. Off-the-shelf software can be a millstone around developers' necks.

4) Support from Real-World Developers
Although there's no formal technical support department for open source software, there are plenty of free, community-based support options such as online discussion forums. In many cases, the contributors to these support communities—developers themselves with real, hands-on experience—are more knowledgeable than the front line tech support agents working for proprietary software vendors who are unable to answer anything beyond typical FAQ questions (And if you want real technical support, you'll have to pay a premium either in the form of costly support licenses or pay-per-call fees.)

5) Product Longevity
Proprietary software puts you at the mercy of the software maker. They might decide the software is no longer marketable and drop it from their product line. Or they might decide to change the software's underlying architecture and technology, making it incompatible with previous versions.

The technology backbone of most open source software is based on popular, industry standard technologies—PHP, Java, MySQL, PostgreSQL. There's no shortage of expertise in these technologies so open source software puts the longevity of your system in your own hands. You decide what you want to change and when you want to change it. Your open source solution will "live" as long as you have people willing to manage your system and provide future development.

Five reasons to consider a proprietary off-the-shelf solution

1) You Don't Need a Lot Technical Expertise
Developers of proprietary, off-the-shelf software go to great lengths to hide the nuts-and-bolts from customers. They try to make their software as quick and easy to install and set up as possible—even by those with limited technical ability—by encapsulating many of the tool's components into its installation package. If you're not up to the challenge of a DIY open source solution or you don't have the time to overcome the learning curve, this is your best bet

2) Comfort from an Established Company
Large, established commercial software firms have huge market shares and customer bases. There's a sense of security that comes from adopting proprietary software from established companies such as IBM or Microsoft. You know that they're not likely to disappear overnight. But it's not uncommon for an open source project to reach an apex and then slowly begin to decline as the project community loses both interest and developers. Some will find this lack of stability in community-based, open source development far too risky.

3) Easier to Convince Senior Management
Senior management hates to rock the boat, and adopting open source software in a large corporate environment is still a novel idea to many. Getting them to agree to do away with traditional software makers in favor of an open source solution developed by what they perceive to be a community of "faceless" programmers will take them too far out of their comfort zone. They might even consider an open source solution as "not serious" when compared to tools from well-known and established commercial software makers.

4) Someone to Hold Accountable
When something goes wrong, we always want someone to blame. So if a CMS mysteriously gobbles up an organization's data, senior management will want someone to be held accountable—and possibly to take legal action against. It's tough to do this with open source software because a community of volunteer developers usually develops it.

5) More Formal and Personalized Support
Proprietary, commercial software makers tend to offer more personalized, one-on-one technical support. They can walk you through different troubleshooting procedures regardless of your technical skill level.

With open source, troubleshooting is a DIY process. Informal support comes in the form of community-based Web sites, knowledge bases, and discussion forums. If you have a problem or question, you have to hope that someone in the community will have a solution for you. Plus, when someone does answer your question, you need to have the technical know-how to put that solution into play.

EVALUATING CMS TOOLS

	CMS 1	CMS 2	CMS 3	CMS 4	CMS 5
GENERAL					
Server					
Web server					
Database					
Operating system					
Programming language					
License type					
Cost					
SECURITY & ADMINISTRATION					
Inline administration					
Online administration					
Granular access control					
Authentication type					
SSL compatible					
Activity logs					
Usage statistics					
FTP support					
CONTENT					
Inline content management					
Online content management					
WYSIWYG editor					
Content approval					
Content locking					
Version control					
Multilingual support					
Digital rights management					
Content scheduling					
Syndication					
Custom branding					
Themes & skins					

WEB 2.0					
Blog					
Wiki					
Social networking					
Tagging					
COLLABORATION					
Chat/IM					
Discussion forum					
Shared calendar					
User survey/poll					
Contact manager					
OTHER					

From Fingers to Keyboards

A CMS is meant to help you do your job; it doesn't do it for you. It's unrealistic to expect highly disorganized employees who can't even find something you gave them 10 minutes ago to suddenly see the light and manage content on a CMS efficiently.

A successful CMS doesn't begin and end with the tool itself. It's only one component in a much larger solution—one that begins at your own fingertips. A CMS can't help an organization improve on something it never had to begin with. So if you throw technology at a human problem that hasn't been addressed, it won't be the system that fails; it will be the people who put the system together who fail. ■

Paul Chin (www.paulchinonline.com) is a freelance writer and journalist. He has previously worked in the aerospace and competitive intelligence industries as a software developer and CMS specialist. He currently writes on a wide range of IT topics, including systems development and security, digital communications and media, content management and web design.