

Table of Contents

Table of Contents	2
The Product Paradox	3
Deciding Whether to Build or Buy	8
Techniques to Get the Job Done	10
How to select decision makers	10
How to select a product	11
A high-level overview	12
Canvassing the Market	12
Making the first cut	13
The sales presentation	14
The RFP	15
Selecting finalists	15
Technical drilldowns	16
Envisioning the project	17
References	18
Vendor presentations	19
Making the final call	20
Taking Stock of the Deliverables	21
Product files	21
The selection criteria and RFP	21
The score card	22
The design diagrams	22
The decision report	23
Risk assessment update	23
Project plan update	25
Taking Stock of Your Staffing Needs	25
Sorting Through the CMS Selection Criteria	26
Business criteria	27
Overall criteria	28
Project process and staff	28
Development environment	29
Localization	31
Training	32
Technical support	32

Deployment	34
Collection criteria	35
Authoring criteria	35
Acquisition criteria	37
Conversion criteria	38
Aggregation criteria	39
Management criteria	40
Storage	40
Administration	41
Workflow	42
Management integrations	44
Version control	45
Reporting	46
Disaster recovery	47
Security	47
Performance	48
Publishing criteria	48
Templating criteria	49
Content deployment	50
Publication platforms	51
Personalization	52
Publication integrations	54
Web UI	54
Web architecture	55
Summary	56

In years past, there were no CMS products on the market. In the future, you'll no more think of creating your own CMS than you'd think of creating your own ERP system. Today, it's likely that you can get most of what you need from one of the commercially available CMS products. Still, it's likely that you need to do a fair amount of custom programming to get the results that you desire.

In this white paper I provide an overview of the build vs. buy decision, giving you some basis on which to decide whether you're better off building or buying your own system. Assuming that most organizations prefer to start from a commercial product, I spend the bulk of the white paper discussing the process for selecting the product most suitable to your needs.

The Product Paradox

Between the time that it became possible to deliver information electronically and the time that it became necessary to do so, an impressive array of content mills were invented to fill the gap for

organizations that had no choice. In the late 1980s and early 1990s, I invented five or six authoring and publication systems that were tailored to a particular group of people with a particular content set. About the time that I started taking a more general approach to the problem of content management, so did others, and the content management industry was born.

In the mid 1990s, I did most of my work for Microsoft. At that point, the company's Web site (at www.microsoft.com) was already getting out of control. One of the main contributors, the Microsoft Office group, had thousands of pages that needed to be tracked and posted each month. I was given the assignment to create some sort of management system for this process. "What's to be on the pages?" I asked Howard, the project manager. "We don't know," came his response. "Okay," I said. "What kinds of pages are you creating?" "We're changing that all around, so we can't really tell you," replied Howard. "Hmm," said I. "What sort of metadata do you need to track for these pages?" Howard smiled. "Any kind," he told me. "We're going to want it to be open, so make it that way." This wasn't going to be an easy project. "Wow, okay, umm, I'll see what I can do," I said. "When can I get the final page designs?" Howard smiled even wider. "You can't," he chuckled. "We'll be redesigning right up until launch. Make it so that whatever design we come up with, the tool can handle it." I picked my jaw up off the floor and we started designing. Before too many hours, Howard and I had a plan that *ought* to work.

Working feverishly with Daniel, a very young but quite brilliant recent immigrant from the former Soviet Union, I conceived and birthed WebPub. WebPub was open. It made no assumptions about the kind of pages that the company would produce. We could set it up for any navigation structure, page types, and page design. We invented the following:

- ✂✂ **An administrative interface** for setting up the components and elements of the system (although we didn't call them that).
- ✂✂ **A contributor interface** that presented forms, where you could enter the content of a component (although we didn't use Web-based forms).
- ✂✂ **A templating system** that enabled you to type special syntax into an HTML file to add content and navigation to whatever design the page had (although the syntax was proprietary and limited).

WebPub was a success. It became one of the first Web-site content management systems used at Microsoft. The system helped the Office group cut the redesign cycle on their large site from three months to three weeks and was used for more than two years (a real accomplishment at Microsoft).

I spin this tale not to get you to buy WebPub, but to discuss the lessons that it taught me.

Because WebPub was such a success, other groups inside and beyond Microsoft also wanted to use it. We hastily produced version 2.0 with documentation and tutorials and called it a product. *Ha!* After a few months, we realized that WebPub was the idea for a product, not the product itself. To turn this idea into a reality, we'd need to remake our entire company. In addition to a great idea, a product company needs the following:

- ✂✂ **Continual development:** Before version 2.0 was even released, we saw how much more flexibility we really needed. Potential clients constantly asked for more than our immature application could deliver. Daniel and Bob needed to be replaced by a complete development group.
- ✂✂ **Support:** Few people trust a software product (and for good reason), and they all have a lot of questions. Daniel and Bob needed to be replaced by an entire support staff.
- ✂✂ **Sales and marketing:** As a service organization, we were accustomed to personal relationships with a small number of clients. Selling product is a numbers game. You need to hit as many people as possible and move a lot of units.

☞ **Finance:** A product organization spends a lot of money up front and reaps profit through high sales down the road. As a service organization, we were set up to be paid for every hour that we worked. There was no pot of money waiting to be spent up front.

Note

Other CMS product companies are on the road not taken by WebPub. They had a good idea that worked in a particular context and the vision to sell it to the world. They've tried, more or less successfully, to remake their companies into product development, marketing, and sales organizations. The good ones are those that have a strong and continually developing technology base as well as an organizational structure that supports the continual revolution that's needed to keep the product moving forward.

Doing and then not doing WebPub taught me a lot about the contradictions and paradoxes that motivate the content management industry. Being involved in CMS vendor selection processes since then has helped me see from the customer's side the following contradictions that CMS companies must live with:

- ☞ **Innovation vs. legacy:** In any fast-moving industry, product companies have a constant struggle between continuing to support their old versions and releasing new versions that transcend the old rules. This conflict, I believe, is what kept the document management industry from dominating the content management industry. The drag of its existing client base kept it from seizing the new opportunities of the Web.
- ☞ **Sales vs. technical staff:** To build a strong product, you need a great technical staff. To sell that product, you need people who know how to make and close deals. The people who want to buy a CMS usually prefer to talk to the technical staff. The people who sell a CMS want you to talk to the sales staff. The sales staff has more time to talk to you, is usually much better at talking, and knows how to get you to buy. The technical staff is usually too busy, not good at talking, and may even tell you not to buy the product. The position of sales engineer was created to deal with this dilemma. If they're good, sales engineers can talk as well to programmers as they can to business people and bridge that gap. If they're bad, they're too poor at interpersonal communications to make it in sales and not technical enough to answer even the least probing questions.
- ☞ **The "next version" issue:** The functionality that you really want is always going to be in the next version. The reason is simple: The released product necessarily lags behind the market need, which, in content management, grows and evolves really quickly. In your search for the right product, you naturally take for granted the basic functionality that most products have and focus most on the stuff that's particular to your initiative and infrastructure. So that's what you ask for. To survive, the product companies must be responsive to customer needs and so, if enough people ask for it, they must plan to include functionality to meet the new need in the next version. Of course, by the time that the next version ships, the new functionality is taken for granted and there's a new set of needs that everyone wants. You should expect the next version issue. In fact, if you don't hear "next version" in response to many of your questions, you should be suspicious that the company isn't keeping up with the market. On the other hand, you should require solid evidence that there's more than vapor behind the next version.
- ☞ **Vision vs. reality:** Every good company has a vision of the future (such as the paperless office, a computer on every desktop, and so on). On the other hand, just like the next version, the future is always brighter than the present. The present is marred by the sad facts of reality. (The supporting technologies aren't yet ready, bandwidth isn't high enough, and so on.) The company needs to sell you fully on the future, so you put up with the present. You need to listen to this vision of the future and make sure that you like it and then get back to the present and make sure that you can really live with all those sad facts.

⚡ **Biggest vs. newest:** As all big software companies have learned, you can be market leader or you can be market innovator, but you can't be both. The bigger a share of the market that you own, the more legacy customers you must support and the more new customers you need to attract to hold your position. Market leaders need to play it safe and play to the masses to be market leaders. Innovators, on the other hand, are up-and-coming players: They seek to undercut the leaders with their speed of development, revolutions in technology, and their ability to remake themselves as the market changes. Neither position is bad or should necessarily reflect poorly on a particular company. But you should question any company that claims to be both, and you should recognize that, in an industry as immature as content management, long-term industry leadership is far from decided.

⚡ **Me, too, and not me:** Especially in these early days, the content management industry is extremely competitive. Companies are constantly monitoring each other's messages and coming up with responses. If one has a good idea, for example, the others quickly latch onto it and say, "Me, too!" Conversely, if one product is seen to have a flaw, the others are quick to claim, "Not me! I do it right!" From the company's perspective, this attitude is necessary for survival. It's a hostile world out there. Customers judge you based on a single random fact that they pick up in an article or by hearsay. You must get past the first uninformed hurdles to get to the real value that your product can offer. If you don't have quick and decisive answers to the "me, too" and "not me" comments, you're sunk. From the customer's perspective, pat answers that always manage to be just what you may want to hear can be a real turnoff.

⚡ **Alliances:** No CMS product can do the job by itself. There's too much involved for one product to cover. On the other hand, customers naturally expect that the CMS can handle any issue remotely related to their information. Some companies respond by continually widening their offerings to try to cover the entire process. Even these companies, however, can't cover everything. And even if they did cover everything, they'd be accused of being proprietary and not allowing you to integrate "best of breed" third-party products. It's a bind - companies lose if they're seen to cover too little or too much. The answer most product companies have found is alliances. They tell you, "Rather than do all this ourselves, we partner with X, who's a known leader in the field of Y." But what does it mean to partner? In the computer industry, it can mean anything from agreeing to put your partner's logo on your Web site to having an ownership stake in your partner's company. The alliance that the customer wants to see is one where the two companies have tested and documented integration methods and have agreed in writing to ensure that their future releases stay compatible. Such solid agreements, however, aren't easy to find. They bind the two companies to a lot of work and to each other for the highly competitive and unknown future.

⚡ **Defining the terms of the discussion:** A successful product company is proactive. They craft a message that they think will resonate with an audience, and then they design a campaign to deliver that message in as many ways as possible to that audience. (Some even use their own systems to do it.) From the company's perspective, they want to lay out the issues and frame a solution by using their product. Then they can train sales staff to deliver the set message, and they can ensure that they can progress from problem to solution, to purchase in an effective way. Finally, by using a set message, they can quickly qualify sales opportunities. If you're too far from the message, you aren't a viable prospect. A set message can be propagated widely and used to quickly find the portion of the target audience that's ready for the message. Customers, on the other hand, want to get comfortable with the messaging and then move into how this product meets their specific needs, not a general need. In a field as immature as content management, your needs are likely to vary from the general (or at least you're likely to think so). A set message works well for the product company but may be seen as shallow or too marketing-oriented by customers.

⚡ **Simple vs. complete:** A product can either be simple to use or complete in its coverage. This is a basic fact of application development. The more stuff that the application can do, the more difficult it is to make it all simple to understand and use. Customers, however, want

both. The answer that people want to hear is, "We can do that, and it's really easy." In reality, the best applications make the easy things easy and the difficult things difficult. Beginners mostly want to do the easy things and need the simple user interface. Advanced users eschew simple interfaces and want all the power of the application in front of them in one hugely complex screen, where they can see everything and do everything. The best products layer functionality so that the beginner's tasks are in front of the more advanced functionality. So, rather than expecting that all features be easy, you should ensure that the difficulty of a feature is proportional to the ability of the intended user.

☞ **Why you shouldn't just wait:** From the product company's perspective, of course, you should never wait. You should buy now before your problems get the better of you. Also, while you wait for the right price or functionality, new needs may arise and you may end up waiting forever, your needs always ahead of the released product. From the customer's perspective, waiting may or may not be good. If there's money that needs to be spent, waiting is bad. If there's indecision or lack of budget in the organization, waiting is good. Decide for yourself how long you ought to wait for the right product to come along.

☞ **Stock vs. custom functionality:** From a product company's perspective, the best approach is to build an application infrastructure that provides a robust set of basic services that can be mixed and matched to meet any of a range of needs. They can then build a user interface on top of the basic services that can be easily modified to meet a range of needs (with some extra programming, of course). From the customer's perspective, customization is a nasty word. Customers want to flip the switch and have everything that they want already there and working. If that can't happen, they at least want to know that there are people available (internally or on contract) who can do the customization and that it's not going to cost too much.

☞ **Single support vs. realistic support:** Customers want a single solution and a single place to go for support (or, as one client put it, "a single throat to strangle"). No matter what or where the problem is customers want a single number to call and an immediate response. This very reasonable requirement makes most product companies cringe. Just think what it takes to provide such support. First, the CMS is so intertwined with organizational infrastructure that to even figure out where a problem's originated, let alone fix it, is tantamount to supporting the organization's entire infrastructure. Second, it takes a lot of skill to diagnose and fix problems in a CMS. This talent isn't cheap to have sitting around waiting for problems to arise. Finally, with companies that stretch across the entire world, the support must be in multiple languages and across all time zones. Can you blame companies for cringing?

☞ **Planning vs. action:** The more planning that you do, the longer it takes you to buy and the harder are the questions that you must ask of a CMS vendor. In addition, the more planning that you do, the less likely you are to want to let the product company set the terms of the discussion. Given this, you'd think that product companies would work against planning. This is rarely the case. With the exception of some companies that try to minimize the entire content management problem, most are in complete agreement that you need to plan before doing a content management project. The reason is simple: If you don't plan, the project is likely to fail, and you're likely to blame the product company. (Some have even been sued by unhappy customers.) To resolve the dilemma between planning and purchase, most product companies try to get you to buy first and plan later. They have professional services to help you do much of what I outline in this white paper. Many do a good job. The only problem is that the entire planning process starts with the assumption that the product in question is the right one for the task. There's no opportunity (as you'd expect) to define the problem and then choose the product.

☞ **What customers want vs. what they need:** Product sales staff see a lot of projects go by. If staff members are good, they can pick up a lot of good and practical understanding about what you, the customer, need to do. On the other hand, it's not in the interest of a sales

engineer to tell you things that lead you away from her product. It's also not in her interest to slow down the process by showing you how your simple understanding needs to be revisited and deepened. The customer, of course, wants the sales staff to be consultants, directing her toward the right solution and offering practical advice where needed. It serves the sales staff to become trusted consultants and offer advice, as long as it doesn't hamper the sale. Be aware of this dilemma and thank a sales person who slows the sales process for you or offers advice that's adverse to her real goal of getting you to buy something.

It's difficult being a vendor. There are contradictions at every turn that keep honest and straightforward sales staffs up at night. There can be either an overly dependent or an overly adversarial relationship between vendors and potential clients, neither of which is good.

The long and the short of it is that customers expect too much, and companies give too little. I don't even bother to discuss companies or customers with ill intent. The best of intentions still isn't enough to resolve the conflicts inherent in the software development process, the sales process, and the content management industry. I've seen many shades of gray, but never a vendor or a customer I'd call dishonest. Mostly, I've seen vendors who want a product they believe in to succeed and customers who don't want to be taken advantage of and are afraid that they don't know all that they need to know to make the right decision.

I've seen a lot of blame thrown around - customers accusing vendors of lying, cheating, and stealing, and vendors accusing customers of trying to soak them for knowledge with no intention to buy. Both sides would do better simply to understand the constraints of the other and the contradictions of the system in which both sides are embedded.

As a customer, if you don't expect vendors to be objective or to act in your interest if interests diverge, you can still trust and believe what you hear. As a vendor, if you don't expect that customers are ready or able to make a decision or that they'll easily give trust, you can still gain that trust and make a sale. After all, most customers are willing to buy something and most products have something worth buying.. The process turns out best for both sides if the customer knows what to ask and the vendor knows how to quickly answer the questions that determine whether the product fits the need.

Deciding Whether to Build or Buy

It wasn't so long ago that there were no commercially available content management systems. In those days, the decision was easy: Do all management by hand, build your own system, or stay out of the game. With the exception of the old SGML systems doing documentation and some text-manipulation tools for programmers, you could buy nothing that advertised any sort of management functions at all. This was less a problem than you may imagine, looking at today's world. Few organizations had made any kind of move to electronic publishing, the Internet was a government experiment, and any companies putting information on CD-ROM were expected to be software development houses.

Today, everyone in this business is at or near a crossroads. I can't tell you categorically to build your own CMS, and I can't tell you that you should clearly buy a system. In some circumstances, you may not have an option, because nothing like what you need exists. More and more commonly, however, something does exist. Then the question becomes whether it's cheaper and faster to start from the existing system and augment it or start from scratch. It's still rare, but getting less so, to find a system that does all that you want with little or no additional customization.

The buy-or-build decision, then, is a matter of degree. If a commercial CMS is cheap enough and does enough of what you want it to, it's no longer worthwhile to build your own. What I *can* say categorically is that you shouldn't take on a CMS building project unless you have the following:

- ⚡⚡ **A strong leader:** You need someone with the vision to conceive and then direct the development of the effort. This person needs to have a firm foundation in content management as well as in the tools and platforms that you intend to use.
- ⚡⚡ **A strong development staff:** You're going to do some hard-core database and Web development. You need a seasoned and sizable team to handle this task. You want developers who not only know their languages and databases, but also have a feeling for and comfort in editorial and publishing systems.
- ⚡⚡ **A desire for an ongoing effort:** It's not enough to create a CMS; just as must a product company, you must continue to develop it to keep up with the pace of change in your organization and in the content management industry. You can perhaps compete with product companies today in delivering the features your organization requests, but can you still compete in the future?
- ⚡⚡ **A strong support organization:** You must be willing to support the CMS as well as build it. All the support headaches that I mentioned in the preceding section, "The Product Paradox," are yours, not some other company's.

Indeed, all the contradictions that I mentioned for a product company in the content management industry apply to you if you take on the task of building a CMS. The organization looks to you as the "vendor" and traps you in all the same snares in which product companies fall. In short, you should be prepared to create a little product company inside your organization to build and maintain a CMS.

Maybe at this point you're saying, "That's pretty conclusive. I wouldn't bite off that task!" If you are, wait while I balance the scales a little in the following points:

- ⚡⚡ **You still may have a lot of development to do**, even if you buy a CMS. You can outsource this development, but you still must manage it and ensure that it fits into the architecture of the CMS that you buy. Of course, the number of positions and the job duration of the staff do change.
- ⚡⚡ **You still have a lot of support to do**, even if you buy a CMS. Many CMS companies simply can't provide the kind of support that a large enterprise needs. If you use one of these products, you're still faced with building a robust support infrastructure.
- ⚡⚡ **You may end up replacing most of the CMS** anyway. I've seen more than one project where the custom code ended up dwarfing the contribution of the CMS product. At first, it seemed that the CMS would take care of large chunks of functionality. As these projects wore on, the CMS just seemed to get in the way of what could just as easily have been programmed. (At least, it was presented that way by the programmers.)
- ⚡⚡ **You must pay a pretty penny** - a CMS is expensive. For the cost of many of the major packages, you can buy an awful lot of programmer and analyst time.

If you're thinking in general about building or buying a CMS, I hope that I've put you right on the fence (because that's where I am). On the other hand, I hope you see that, if you weigh the issues of your specific situation, the balance ought to tip in one direction or the other.

Finally, take heart in the fact that whatever decision that you make today undoubtedly needs to be reviewed in the next two years and very likely gets overturned by another decision (or at least undergoes a thorough reworking) within five years. This industry is moving too quickly for any decision to last for long.

If you're at all in doubt, it's worth your time to go through a product-selection process. You don't need to buy at the end of it if you decide to build, and you just may find that there's a product out there that gets you enough of the way toward your goal (even if only in the next version) to be worth the price.

In the remainder of this white paper I assume that you're selecting from among the commercially available CMS products. If you're set on building your own CMS, you can use the information in Part V, "Building a CMS," to start thinking about how you may do it.

Techniques to Get the Job Done

The CMS vendor selection process isn't so different from any other vendor selection process that it needs its own separate methods. Thus most of what I suggest is common sense to people who've done a lot of product selection. In the sections that follow, I infuse this common sense approach with a content management focus to give you some advice and guidelines to follow.

How to select decision makers

I've seen more politics wrapped around the product decision than in any other part of a CMS project. More than once, I've seen a CMS project stall and die because of the following:

- ⚡️ The lack of consensus that you previously could choose to ignore comes out and fouls up the entire process.
- ⚡️ Some groups have their favorites and inside players, others have the tools that they're already using, and yet others have one tool that they've heard of and just want.
- ⚡️ Some people see this as the most significant decision of the project and one where they can have a say.
- ⚡️ Some people see this as a way to influence how large budgets are spent.
- ⚡️ Some people think that, if you make a bad call on the product, the project's doomed, so they put too much emphasis on the process and drag it down.
- ⚡️ Some people just think that they're good at system selection and plug in because they have experience.

All these reasons have some validity and some invalidity. My overall feeling is that selection gets all the attention because it's so tangible and easy to understand. Money is spent and a product is chosen. The other parts of the project are more difficult to get your hands around and not so straightforward.

For whatever reason, system selection can be a real hassle if you don't put serious effort into reining in all the people who want to decide, as follows:

- ⚡️ **Depend on the consensus that you built** during the mandate process. If the process didn't yield a list of who gets to decide, you at least should have gotten a group of people with enough authority to help you choose a small but evenly distributed group with no members who've decided before the process starts.
- ⚡️ **Ratify a group process before you begin.** Make sure that everyone agrees to it and agrees that the winner wins. Too often, I've seen a clear winner emerge in this process, only to be shot down by people who don't like it "for some reason." If they can articulate a reason, it can be added to the list and scored with the rest of the criteria that you use. If they can't articulate why they like or don't like a product, their opinion shouldn't count for much. Don't create process on the spot as you need it. This leads to a lack of respect for the process and rigging by people who can figure out how to manipulate the criteria and scoring.
- ⚡️ **Decide on a fair and impartial scoring mechanism** for each scoring step that you include in your process. Agree to the range for each scale and agree to what constitutes each value in the range. Then each person can make a judgment in the strict context of the scale and scoring criteria that you established.

Tip

The more numbers that you have in a scoring scale, the more room you have to differentiate between products - and the more work that you must do to define what each number means. I prefer scales that have three values for subjective measures (bad, okay, and good) and no more than ten for more objective measures. (Zero is no functionality, and ten is far more than you'd ever need.) Make your scales as small as possible while still retaining enough room to differentiate products.

☞☞ **Agree to stick to what your process decides.** It helps no one to add up all the scores and then ignore them because they don't come out right.

Note

If you find yourself stymied by the continual subversion of your objective process, you may try giving up. Maybe an executive decision just needs to be made about the product to use. Maybe your offer to give up gives new incentive to those who need to respect the process (or, at least, gives incentive to their supervisors). In the end, it's more important that some product be used than that it be the absolute best choice.

As far as who should be in the reviewers group is concerned, I can think of two types of people: those who represent a significant organizational interest and those who have the expertise or perspective to judge vendors. Obviously, you should shoot for people who have both qualities. I'd resist people who represent a constituency but have no basis to judge between vendors. As much as someone may want them to serve as a reviewer, they can slow down the process and cause a general lack of respect for the rest of the reviewers from the vendor and the organization.

How to select a product

I've seen all manner of processes used to decide on the best system to buy. In many cases, you can't call them processes at all: You call around, see what other people are using, and go with that. In most cases, however, there's a process, and it centers around a list of requirements, vendor presentations, and some sort of selection committee. The process that I present is comprehensive and exhaustive. If you're planning to spend a lot of money on your CMS, it's worth the extra effort. On the other hand, you should balance the full treatment that I give you with the timing and depth that you think you need. In the full process that I describe, there can be a lot of questions to ask and answers to process. The process runs as follows:

- ☞☞ Create a high-level overview of the CMS project that you can pass around.
- ☞☞ Canvass the market for the products that seem to fit.
- ☞☞ Make the first cut from the list of candidates, selecting those that seem worth really pursuing.
- ☞☞ Send a Request for Proposal (RFP) to those who make the first cut.
- ☞☞ Select a small number of finalists by scoring the RFP responses and any follow-up questions that you ask.
- ☞☞ Have technical drill-down meetings and check references from each of the finalists.
- ☞☞ Have a presentation from the remaining candidates.
- ☞☞ Make a final decision by combining the scores for the references and presentations with those for the RFP.

The following sections give more detail to the process.

A high-level overview

To start the selection process, it's useful to create a short project overview that you can include in early correspondence with the vendors. The overview also should orient your selection committee to the major points toward which you're aiming.

To create the overview, take the high points from your project mandate, requirements, and logical analysis. Here's a sample overview:

Our company is seeking to implement an enterprise-level content management system that aids in the creation, management, and publishing of personalized content to the Web and beyond. At the highest level, the system is successful if it does the following:

- ☞☞ Provides a solid framework behind the staff and processes involved in creating and tagging content.
- ☞☞ Supports a wide and diverse contributor base with tools and processes appropriate to skill level and commitment.
- ☞☞ Provides a full-featured repository where content components and file-based resources can be stored, tracked, updated, targeted, combined, and archived.
- ☞☞ Provides a flexible, template-driven publishing capability that can format and output any combination of content and file resources to standard databases and file formats (including XML, HTML, flat text for e-mail messages, and QuarkXPress for print output).
- ☞☞ Provides strong, easy-to-use workflow tools that can guide the entire creation, storage, and publishing process.
- ☞☞ Supports a robust publishing, testing, and deployment environment.
- ☞☞ Provides a full-featured but extendable personalization capability for targeting and custom publishing of content.
- ☞☞ Integrates with our existing internal systems and current Web infrastructure.

The product or products selected are successful if they do the following:

- ☞☞ Have "out-of-the-box" functionality capable of covering a wide range of the preceding requirements with minimal customization.
- ☞☞ Have standards-based customization and extension capabilities that enable full integration with existing systems and rapid development of additional functionality."

Canvassing the Market

You can begin your search for the appropriate software by performing a broad overview of the CMS market, looking for products that address your general needs. Colleagues and the Web are good places to start.

Here are a few tips for conducting your search:

- ☞☞ **Get recommendations.** Try sending your overview via e-mail to anyone you can think of who may have an opinion. The recommendations give you a place to start and a set of products to which you can compare the ones that you find yourself. Interestingly, this technique may also turn up biases and inside favorites from the people in your organization whose opinion you ask.
- ☞☞ **Keep your analysis close at hand.** Nearby, you should have your requirements, logical design, and the selection criteria that you think are most important; look at them often as you

do your search. This helps you avoid the disorientation that you're bound to feel as you follow endless links and get lost in the rhetoric and marketing spin of the Web sites that you visit.

- ⚡️ **Develop a short set of questions that you ask at each Web site that you visit.** This helps you stay on task and give an even, standard treatment to each product. As you may suspect, I like these three: What collection processes do they cover, what management functions do they cover, and what publishing functions do they include? Use these or create your own, but don't let the site control how you find information or what information you look for.
- ⚡️ **Consider core products vs. peripherals.** I'd roughly estimate that, in mid-2001, 40 or 50 products do enough really to be considered content management systems. (Early last year, there were half that many.) On the other hand, there are hundreds that advertise some connection to content management. Develop a quick eye for the products that don't have enough core content management functionality to qualify for your search.
- ⚡️ **Keep a side list of your special needs.** This list should go beyond standard content management concerns (special marketing needs, unusual publications, particular integrations, and so on). You may come across products that don't address all your needs but do hit one or two. You may find these products helpful later on, either as add-ons to the system that you purchase or as good examples of how a particular need can be met.
- ⚡️ **Document your search.** If you stay on a site for more than a minute or two, there's something of interest there. Paste its URL into a file so that you can track your way back to it later as you say, "I know I saw that somewhere..."
- ⚡️ **Don't spend too much time on this process.** You can easily get lost for days and days. Continue keeping up with the subject on the Web, but don't hold up the rest of the process for more than a day while you choose your initial list of candidates. If in doubt, include a company on your list. You can easily remove them later.
- ⚡️ **Start a file.** In fact, start both a physical and a computer file on each interesting company. You eventually accumulate a lot of material on the ones that make it to the end of the process.

The preceding points address Web sites of CMS product companies. My informal estimate is that far more than 90 percent of your Web search results come from such companies. The rest are from service organizations that do content management projects and from articles, publications, and conference presentations on the subject. Although I've not been overly impressed with the depth or breadth of coverage in the pages I've seen in this category, they're certainly worth perusing for general information and, especially, for comrades who may have faced problems similar to your own. I've always had good luck contacting and getting help (and even friendship) from people I've heard of on the Web.

No need to be stingy in this process: Include any product that looks seriously interesting. If you end up with a list of more than 10 or as many as 20 products, you're on the right track.

Making the first cut

With a little effort, you can winnow down the large number of product companies that have something to do with content management to the handful that seem to address your particular needs.

If you have enough lead time, you can send an e-mail message to or call each company on the list and request a complete marketing package. In addition, go back to each site and collect as much relevant information as you can about each candidate.

Tip

If you request marketing packets, make sure that you ask them to include all white papers, case studies, demo programs, sample documentation, industry analyses, and pricing sheets that they have. These tell you much more than the glossy sheets that are standard fare for marketing folders.

Don't be afraid to summarily dismiss some candidates as you collect information and find out that they really don't fit. If you have time, you can also search the Web for each product name. This finds you reviews and other commentary on the product (as well as any flames or accolades that the product's received publicly).

Tip

Note the names of people who've commented on a product. You may need to talk to them later about their experiences.

The core of the first cut is a preliminary set of evaluation criteria (usually a spreadsheet). Start by synthesizing your requirements and logical analysis with the questions listed in this white paper as well as any others you came up with, to create a high-level cut on your selection criteria. Go back over your notes and links from your market survey and narrow the list to no more than 10 to 20 criteria to apply to each of your initial candidates. Use simple criteria and simple scoring rules (short scoring ranges). Have all your reviewers apply these criteria and, as quickly as possible, score each candidate product, using all the resources that you provide as well as any others that they may uncover and share with the group.

In this process, I prefer a small set of quick passes through the list to find products that easily drop out. People shouldn't expect that they need to do complete and exhaustive analyses. They don't have time for a complete analysis and may become very frustrated. Rather, ease them into the process by doing a number of quick passes. With luck, the first pass or two drops out half or more of your initial list of products. This may be all that you need to make the first cut.

The first cut should yield between five and 10 candidates. With five, you risk having too few to make it to the end of the process; with 10, you risk having to spend too much time collecting information. You decide what's appropriate for your organization and schedule.

The sales presentation

After the first cut, it's time to get serious. Your group should have enough practice reviewing and scoring to be comfortable with going deeper, and you should have a group of good candidates.

At this point, some groups choose to have a preliminary demo from each of the candidates. This is a nice idea for becoming familiar with the products and the organizations behind them.

If you choose to have demos, I recommend the following:

- ⚡ **Don't require preparation of your team or the vendor.** Ask for the standard presentation. This serves a number of purposes. It gets the process going quickly, the company can send its usual talent, and your people can either attend or not, depending on their schedules. It gets the standard presentation over and done with so that, at the next meeting, you can get down to details quickly. It gives the vendor's representatives the opportunity to present their best face and walk you through the stuff that they're most proud of. In the next meeting, you talk much more about the parts that the vendor's not so proud of.
- ⚡ **Aim to understand the product, not question it.** Save your probing questions for later and become as immersed in the product as you can. Be positive, to draw the presenter out and to have her give you the fullest demo possible. Each product has lots of interesting and good features that you should try to understand and appreciate.
- ⚡ **Freely discuss your requirements and needs.** The vendor's presenter is trying to qualify you as a prospect as much as you're trying to qualify her company. Try to answer her questions as openly as possible so that she can get a solid feeling for what you've

accomplished (which, by now, ought to be quite a lot) and what you want to accomplish in the end.

✍ ✍ **Find out the names of the people inside the product organization** that you can contact directly for specific information. If you're lucky or persistent, you can build a good list of expertise for future reference whenever you're having trouble getting an answer.

The RFP

Your main job here is to create the complete selection criteria list and form it into an official Request for Proposal (RFP). You may call it an RFP, or you may prefer to be less formal, but the idea is the same. You create a complete scoring system (most likely a spreadsheet that extends the simple one that you use to make your first cut) that lists all the questions that you need answered to come to a list of finalists, along with a place to score each finalist.

Note

Vendor sales representatives may like to do more of a consultative sell, sitting down and talking at length with you about your issues before trying to propose how the solution would work. You can use the first demonstration meeting to accomplish this or schedule separate meetings. But make sure that it doesn't end up slanting your process toward the company that sells the hardest at this early stage.

Usually, my preference is to ask a lot and expect a lot of my questions to go unanswered. Others choose to ask only the most essential questions and insist on complete responses. I like the former approach because it provides a lot of discussion and input to the next stage of the project. Besides, even if you ask only essential questions, you're not guaranteed to get complete responses without a lot of probing and pushing.

Tip

Just because you get a bad answer doesn't mean that the product really can't do it. Assess the credibility of the answerer and, if you want, give the company a break by digging deeper until you find an answer that you can trust. Similarly, answers of "yes" or "we do that," count for little unless you dig deeper.

If you choose to send a complete set of questions, warn the vendors that you're giving them a big job and make sure that you give them ample time to respond. Of course, no matter how much time you give them, most wait until the last minute and then panic if they can't complete the document.

Selecting finalists

I've learned not to expect too much from RFP responses. They often fall far short of what I'd prefer to see. I believe that the problem usually is that I expect too much detail and the respondents have too little time and training to respond adequately. You may think that, after a while, companies would build up a pool of stock responses to respond in full to almost any question that you may have. So far, this has not been my experience. Still, this is a very young field that's growing so quickly that it's understandable that vendors can't spare the time or talent to respond at the first request.

That's why I believe that it's a good idea to plan to have follow-up meetings with the candidates that make the RFP cut. Before you schedule these meetings, see whether any candidates drop out due to lack of response or unacceptably poor performance on the RFP. For those that make a reasonable effort and have answers that are on the target, if not on the bull's-eye, more discussion is warranted. By this time, you've undoubtedly been noticed by the companies, and if you're a big enough fish, you have their eager attention as well.

Again, in these meetings, I favor multiple quick passes through the questions over a single, in-depth pass. On the first pass, focus on the issues that the candidates did particularly well or poorly on in their RFP response. If they did poorly, was it because of a lack of responsiveness to

the question or because they can't do what you want? Make a list of all the weak points. Do these points amount to a disqualification from the process, or with some additional time, may they be resolved favorably? If more time is warranted, divide your list into topics. (The criteria categories that I propose in the section "Sorting Through the CMS Selection Criteria," later in this white paper may help here.) Rather than asking the same person to answer the same questions again, see whether you can find the right person inside the product company for each type of question. Providing your questions in writing and letting your sales contact know that continued participation hangs on deeper responses from a qualified expert may help bring in the right people.

You can use e-mail, phone calls, one-on-one meetings, or group discussions to move each question to a final score. You may find it helpful to divide your team by expertise or interest to ensure the most efficient coverage of the questions.

Tip

Don't divide your team by product if you know that there are favorites. Instead, divide them by question type so that each person sees all products.

At some point after your first pass through the RFPs and before your follow-up is complete, you should officially score the responses. A spreadsheet with the tallying built-in is helpful. Make sure that each reviewer is fully aware of and in agreement with the process. An answer of "I don't know" or "I need more discussion" ought to be allowed. The reviewer herself can decide whether her reservations are enough to postpone the final scores. I've generally had several people score each product based on those team members' ability to judge the products and vendors. Then I've used the spreadsheet to average the scores to come to the final score for each criterion.

You want to prioritize each question that you score by giving it a weight. A question with a weight of 10, for example, is counted ten times as heavily as a question with a weight of 1. Make sure that you establish these weights *before* you begin scoring.

Note

How much of your process to reveal to the vendors is always an issue. I'm of two minds. On the one hand, I think that the more they know about the criteria, how they're scored, and what you're questioning, the better they can respond. And most use the information that you provide to try to serve you better. On the other hand, the more that they know, the more some of them may try to interfere and bias the process toward themselves. You decide.

Your schedule and allotted effort should decide how many finalists you can choose. Two is low, and five is high. For the finalists, the process requires much more effort on both sides.

I emphasize again the points that I made earlier about who gets to decide. If you want a clean, decisive process, decide beforehand what your process is and stick to it. If you continually change your criteria or decision process, you frustrate your team and the vendors and you end up with no result or a result that's questioned and subverted.

Technical drilldowns

With a small pool of finalists, the time is right to get down to the small details. Do a more thorough job of analyzing the RFP and follow-up questions from the finalists, and schedule one or more meetings where your technical experts and theirs gather to envision the relationship. Experts can be of a variety of stripes, as follows:

☞ **Business experts** can gather to discuss terms of the contract that you may sign.

☞ **Editorial experts and information architects** can come together to discuss the collection system and metadata modeling facilities.

☞ **Programmers** can meet to understand how the system may be developed.

⚡⚡ **IT folks** can envision the architecture and deployment of the system.

⚡⚡ **Publications staff** can deliberate on the system's capability to produce the appropriate output.

As part of this set of meetings, you may want to schedule additional demos of the product or other background sessions to get your team fully up to speed on the capabilities of the products.

These discussions should result in a clear idea in the minds of your team of how you'd complete your tasks with the different systems. At this point, you should feel comfortable sharing your full requirements and logical design with the vendors to give them as much as possible to go on and to test the strength of their understanding and desire to work with your team.

The vendors should understand that they need to move beyond the sales staff to bring in people from their development groups or from their professional services groups who can really contribute. It help to bring these people in if you're clear, during these meetings, that you're testing the relationship that you hope to have with the company that you finally choose to work with.

I favor having each meeting result in a diagram that each side believes represents what the system may look like or do, given your needs and their product. Although the content of the diagrams change based on whom you're meeting with, the concept should hold across all the meetings that you hold. If you do this, you thank yourself later. You can hang all the pictures next to each other to help you choose from among a possibly confusing array of competing approaches.

Envisioning the project

In the technical drill-down meetings, you've worked through most of the issues that were troubling. If you have, you're now ready to figure out just how the project would progress with each potential product.

You can start by adding detail to the RFP responses that you receive and rescoring them based on the information that you receive through follow-up meetings. If you can get past the bad feeling that they could have told you in the first place all the stuff that you found out later, you can capture and organize a lot of the detail that you discover and still have a single place to grade the company (in your scoring matrix). Make sure that you mark the revisions that you make to the RFPs so that you can tell what you added and what they originally said.

Compare the first scores that you came up with to the revised ones, as follows:

⚡⚡ The second set of scores are more reflective of what the product and the vendor are really capable of doing.

⚡⚡ If the second score is higher than the first, you had a bad sales rep.

⚡⚡ If the second score is lower than the first, the difference is the amount of spin added to the original RFP response.

Your biggest task is to create modified project plans for each prospective product. Work through your existing project plan with representatives from the product company. Work through the complete plan, but pay particular attention to the parts of the project that the product can't do out of the box, as follows:

⚡⚡ **For the major tasks that require custom development**, develop an extension to the project plan that has enough information in it to be used as a basis for cost and schedule estimates. Try to get as good an approximation as you can of the amount of customization that you need to do to augment the stock functionality of each product. You can perhaps get the vendor to separately estimate doing the custom development.

☞ ☞ **For the major integrations**, try to get as specific an explanation as possible of how the integration would occur. Get whatever time and cost estimates that you can and try to plan the integration in the same way that you planned the major customizations. You can also possibly get an estimate from the vendor for integration services.

You can update your risk assessment, once per potential product, noting any new, heightened, or reduced risks that using that particular product entails.

Note

If you think that this is a lot of work, you're right. The only consolation is that one of the project plans that you create now is the one that you can actually go forward with later.

The vendors can be very helpful to you in this endeavor. They have a lot of great experience doing CMS projects and can bring a great level of detail to your plan. If they're not helpful or know less than you do about your plan, that's good information to know as well.

References

Rightly or wrongly (rightly, I believe), most companies don't give you references until they're sure that you're a highly qualified prospect. The reason that they wait is that they have precious few clients willing to put out the effort that a reference requires, and they want to reserve them for the highest value. There are some companies that don't have many references because they're not well liked, but that's not something that they're likely to tell you. With any luck, you weed out these vendors in your first cuts.

If you've followed anything like the process that I've laid out, you have more than enough proof at this point that you're serious. References are just about the last step that you take on the path to your decision.

What you want from references is people like you who are using the product to do the things that you want to do. You need to be very lucky to get such a perfect match. More likely, you find people who speak kindly of the product (same as you'd do if you agreed to be a reference) and are in a different industry or have only some of the same issues that you face.

One mistake that people make with references is to leave the choice entirely up to the vendor. What can you do to come up with two or three people who are similar to you and are using the products that you're reviewing? Following are some things that you can do to reach that goal:

☞ ☞ **Check the Web.** In your initial search, did you find any people who had opinions on the product? If so, ask them to whom you can speak. If you didn't fully search the Web for the products that are finalists, do it now.

☞ ☞ **Check the logos on the vendor's site.** Product companies proudly display the names of their major clients. Who in these organizations can you talk to?

☞ ☞ **Check conferences and journals.** Although content management is still a bit obscure, most technology magazines have done at least one story on it. In addition, there are content management conferences where people do case studies on the use of one product or another. Maybe you can find just the person that you need to talk to in one of these venues.

The point is that you need comparable installations to help you decide which product is best for you. Whether you get those from the company or by yourself is irrelevant. Besides, if you tell the companies that you're doing your own search, they may be a bit more helpful in the references that they provide.

After you have the references, you need to schedule a conversation. If you can go to them, you get a lot more time and effort from them. Plus you can sometimes see the system running and talk to all sorts of people. If you can't manage to make it to the reference's site, a phone call may still be enough.

The biggest issue is how much time and effort you can get from the references. You should need no more than an hour or two if the person is sufficiently briefed on her own system. Prioritize your questions to make sure that, if you're suddenly cut off by a higher priority issue, you've at least gotten the main points out.

To me, the best approach is to recognize that these references are your peers. Even if you choose a different product, here's a person who's in the same boat as you. What do you have to offer her? At the very least, you should have a little sympathy for her problems. More likely, you have some ideas or information that she can use as well. If you can establish a relationship rather than being a single inconvenience that the reference must endure, you get more and better information. More important, you may get a friend that you can collaborate with over the course of your career in content management.

What should you ask the reference? The same questions that you've been asking yourself and the vendor. Use the reference in conjunction with the technical drill-downs to solidify your feeling about how it would be to use this product. Of course, general questions such as the following are always a good start:

- ☞ Did the vendor deliver what she said that she would?
- ☞ Did the product do what you expected it to do?
- ☞ Have you gotten the support that you needed?
- ☞ Was the training any good?
- ☞ How close are your requirements to ours?
- ☞ What problems did you have with the installation?
- ☞ With what are you integrating?

Some of the most telling answers come from asking questions such as the following:

- ☞ What was the most surprising or least anticipated part of the implementation?
- ☞ If you had to suggest one improvement to the product, what would it be?

These may be good icebreakers, but make sure that you get quickly to the key issues to make the best use of the limited time that you probably have.

Vendor presentations

By now, you're very close to a decision; in fact, you may have a clear leader who all but has the contract. But there's no contract yet; only a lot of discussion. You also likely have a list of issues that never have been successfully resolved for each vendor. Finally, you may have people with the authority to sign the purchase order who'd like a chance to meet these people before giving them a lot of money.

You can combine all these needs into a final vendor presentation that has the following purposes:

- ☞ **Final resolution of any outstanding issues.** What you should aim for is that there are no outstanding issues that would be embarrassing to talk about at such a weighty meeting.
- ☞ **A full discussion of the terms of the agreement** that you may sign with the company. Prior to the meeting, you should get sample contracts and pricing and service schedules that you can review and comment on.
- ☞ **An executive review of the company.** This gives your sponsors (or whoever else wants to attend) the chance to meet with whatever executives the product company can send. Having the big guns there on both sides can have an amazing effect on the issues that seemed unsolvable before. In addition, you have the right people present to discuss any strategic

aspects of the relationship that you may form. Can the two companies help each other in more than just content management?

☞☞ **A cost estimate.** You should work to have a final price tag for the product available at the meeting. Alternatively, you can set this meeting as the point at which the price is set. You may have a lot of hoops to jump through to arrive at a price. In my experience, CMS companies aren't forthcoming with cost estimates. That's another good reason to set this meeting as a deadline. You also may want to have all the customization and integration estimates that you prepared ready as well at this meeting to verify them with the company's official representatives.

This is decidedly not a meeting to demo the product or have the sales people present their slide shows. It's a down-to-earth, do-we-want-to-work-together meeting. The time has passed for features. It's time to make a final determination of the company itself.

Tip

Make sure that you brief your sponsors on the need to avoid wasting meeting time on demos. Find out whether they'd like a demo, schedule a separate time before this meeting, or invite your sponsors to one of the earlier meetings where they can ask functionality questions.

Centering the discussion on the contract has the effect of putting a seriousness and reality on the discussion that no other context can yield. There's nothing like saying, "So can we insert a sentence here that binds you to the conditions that you just agreed to?" to make the point hit home and finally be decided.

The result of your final vendor meetings should be that your sponsor group (and the people who sign checks, if they're not the same) should form an opinion about the companies that you're reviewing. If their opinions match yours, you're finished. If the two don't match, you have a bit more work to do.

Making the final call

If you've followed the process to this point, you have enough information to make a selection, as follows:

- ☞☞ You have large files on each product.
- ☞☞ You have RFP responses that are augmented and scored.
- ☞☞ You have diagrams showing each possible approach.
- ☞☞ You have prospective project plans for each product.
- ☞☞ You have risk assessments for each product.
- ☞☞ You have references from peers using each product.
- ☞☞ You have cost estimates.
- ☞☞ You have agreement in principle on a contract for each product.

You have the information that you need, but you also need the organizational will and process to decide and move on. Your selection process should have produced a numerical winner in the RFP scores, as well as project costs. In addition, you have one or more subjective assessments from your team and sponsors on each of the other deliverables from the process.

If you're really lucky, all these align and point toward a single winner. Otherwise, you have the following options (in order of my preference):

- ☞☞ **Drive for consensus** by hashing through the disagreements in open debate until all opinions converge enough for a decision. You can expect a lot of debate but a solidly supported decision.

✂ ✂ **Create an objective scoring** method for the subjective factors. You can create a scale for each factor and a weight for each subjective factor relative to the weight of the RFP score and the product cost. Then create a calculation that sums the scores for each reviewer and product and calls a winner.

✂ ✂ **Escalate the decision** to someone who can make and enforce it. Provide the decision maker with all the information that you've collected and let her decide.

Clearly, the process that you choose has more to do with the nature of your team than the information at hand. A well-coordinated team with mutual respect and a strong unified vision of the project can easily reach consensus. A team that feels bound by the process more than by respect for each other is likely to accept an objective scoring. Even for a well-functioning team, the objective scoring method may be fastest and easiest. If the team isn't at all well functioning, the only good course may be to punt the decision to someone higher in the organization. That at least ensures that the decision is made and everyone abides by it.

Of course, the worst time to decide how to decide is at the end, when emotions may be high. At the beginning of the selection process, you should assess your team's inclinations and decide how to decide.

Taking Stock of the Deliverables

I've spent ample time on the selection deliverables in the course of describing the process, so I just provide a few additional comments in the following sections.

Product files

It's a good idea to have the files that you created for products in a central place where the whole team can access them. I also favor having them jointly owned by all team members, who can add to them as they find good material. Don't forget to attach a URL or other identifier to all materials so that you can track back to their sources as needed. You undoubtedly need a directory as well as a physical file for each product. I've had luck with a big binder, where all materials were three-hole punched and the directory location for computer files was written on the cover.

You can include the notes that you compiled from content management Web sites in a separate file. If you're way more organized than any group I've ever worked with, you can continue to add to the links file and periodically publish it as a resource for people in your organization to learn about content management.

The selection criteria and RFP

I'd suggest starting with a spreadsheet that catalogs all your criteria and their weightings. You can continue to develop the spreadsheet until it's time to send out the RFPs. At that point, you can send a spreadsheet to each vendor, but it's more friendly to convert the spreadsheet into an electronic document (Microsoft Word is an accepted standard) and send it that way. The conversion takes a little extra effort, but it makes the RFP easier to work with (at least for the CMS product vendors). In addition, you can add a cover page and company background that's difficult to do well in a spreadsheet.

I've found it particularly useful not only to provide a long list of questions in an RFP, but also to include a short description of what you're trying to accomplish at the start of each section. You can draw from your logical design to create these introductions. By doing this, you not only provide needed context to the person answering the questions, but you also give yourself a way to make sure that you've covered all the most important questions that your logical design may have raised. As you put in the introductions, ask yourself, "Do the questions that I ask here cover the implementation of what I just said that we want to do?"

You may also consider including parts of your project mandate and the high-level overview that you wrote at the beginning of the selection process (which I present in the section "A high-level overview," earlier in this white paper. Finally, consider augmenting your lists of fill-in questions with scenarios. Suppose that you have some authors who seem particularly problematic. (Maybe they use strange tools or are remote). In addition to (or, in some cases, instead of) asking your authoring questions, you can explain the situation with your authors and ask the vendor to propose a solution. You may get mixed results, as scenarios require more thought and effort by the respondent than short answers, but it may be worth a try to see who has the most creative solutions.

The score card

You can build your RFP scorecard directly into the spreadsheet that you devise for selection criteria. If each reviewer gets a copy of the spreadsheet file, each has her scores immediately tallied for her. To combine the scores of all reviewers, you can create a master spreadsheet that looks inside each reviewer's copy and tallies the full results, or you can just copy and paste the individual scores into a new sheet.

Tip

Make sure that you have a way to link questions in your RFP to your scoring matrix. I prefer to number the sections in the RFP so that I can refer to them exactly in the scoring matrix.

If you decide to score subjective factors, you can simply add them to the RFP spreadsheet file and sum them with the rest of the RFP questions.

The design diagrams

In design, a picture is worth more than 1K words. By making a diagram deliverable from each technical drill-down meeting, you accomplish the following:

- ☞ ☞ Ensure that the meetings focus on design rather than become unfocused discussions of what's possible.
- ☞ ☞ Ensure that some solution is reached. Meeting notes can meander and never conclude. A picture can be more or less precise, but it always tells a complete story.
- ☞ ☞ Can compare one product easily to the others. Each diagram is a possible solution to the same problem.
- ☞ ☞ Have good input to the implementation process that comes next.

Convinced? If so, here are some guidelines to help you make the most out of these diagrams:

- ☞ ☞ **Title them.** Preferably, the titles should be the names of the systems that they describe (such as "PageMaker Publication System" or "Repository Structure"). You also may include a short description of the problem that each diagram addresses.
- ☞ ☞ **Label all major shapes and lines.** It's amazing how quickly people forget what all those parts are and do.
- ☞ ☞ **Include inventories in the margins.** Lists such as staff, CPUs needed, and connections all help you later develop plans from the diagrams.
- ☞ ☞ **Use a standard iconography.** If everyone agrees on the picture for a database, LAN connection, Web server, and the rest of your symbols, your diagrams are easily read by anyone on the team and more easily compared to each other.

The decision report

At the end of the selection process, you've reached a major milestone in the project. You're finished with the last big decision. From now on, the decisions are all about *how*, not *what*. To mark the milestone and also to inform the organization, you should prepare a short decision report.

Include the following in the report:

- ✍✍ The high-level overview that you wrote at the beginning of the process.
- ✍✍ An overview of the process that you followed.
- ✍✍ Major factors that led to the decision.
- ✍✍ An overview of the product selected.
- ✍✍ Information about the company selected.

This report should be distributed at least to all the people who participated in the project so far, and, ideally, to anyone who may participate in the CMS in the future.

If there's enough interest, you may want to schedule an open session with representatives of the product selected so that anyone can come and see a demo and ask questions.

Tip

After all the hard work that you do on product selection, don't forget to reward yourself and your team. This decision is a clear cause for celebration. Don't forget the people at the product company. If you used this process, they worked as hard as you did and deserve a party, too!

Risk assessment update

At the beginning of system selection, your risk analysis has mostly high-level organizational issues and stuff about how you possibly can't do what everyone wants to do. As a result of the selection process, you can add a set of more concrete risks and mitigations to your assessment. Table 16-1 provides an example of a simple product risk assessment to get you started. (Effect and probability are on a scale of 1 to 5, where 1 = low and 5 = high.)

Table 16-1 Product Risk Assessment Example

Area/Risk	Effect	Probability	Score
-----------	--------	-------------	-------

Design/Planning

Technical training and implementation, support not timely	3	1	3
---	---	---	---

Development

Major development effort required	3	3	9
-----------------------------------	---	---	---

Inadequate developer documentation	2	3	6
------------------------------------	---	---	---

Integration

Major technical issues related to integration	3	2	6
---	---	---	---

Integration results in poor site performance	3	2	6
--	---	---	---

Personalization inappropriate or inadequate	1	3	3
---	---	---	---

Rollout

Usability/training issues	2	2	4
---------------------------	---	---	---

Ongoing Support

High level of end-user support required	2	2	4
---	---	---	---

Inadequate end-user documentation	1	3	3
-----------------------------------	---	---	---

Incapable or unresponsive help-desk staff	2	2	4
---	---	---	---

Scalability issues, significant hardware requirements	2	4	8
---	---	---	---

**Futures **

Vendor's technology becomes obsolete	2	2	4
--------------------------------------	---	---	---

Vendor becomes financially unstable	2	3	6
-------------------------------------	---	---	---

Vendor experiences high level of change/turnover	2	4	8
--	---	---	---

Vendor's technology direction becomes misaligned with ours	2	2	4
--	---	---	---

Vendor's market direction becomes misaligned with ours	2	0	0
--	---	---	---

Overall Score 78

Project plan update

If you did your homework in the selection process, you need little extra work at the end to update your project plan. Make sure that you go through it again, however, and fill in all the spaces that you left blank because you ran out of time.

If you've added people to your team from the product company, make sure that you add these new staff members to your plan. They probably have their own project plan, so you must work together to integrate the two. In any case, you should maintain a master plan where the all staff and schedules come together to complete the rest of the project.

Taking Stock of Your Staffing Needs

The CMS product selection process begins with the build-or-buy decision. This is a big decision with lots of political and technical overtones. I believe that the decision should be presided over by as high a person as you have on the core team. In my taxonomy, that's the content manager. The content manager most likely at least wants to supervise this activity and pass the

recommendation by the sponsors before making a final decision. The content manager or her equivalent probably also should be the one to write and circulate the high-level overview.

I favor having a single person head up the vendor-selection process and others participate as reviewers or outside observers. My preference is also to have the person running the process refrain from being a reviewer as well. There's just too much conflict of interest between coming up with the evaluations and doing them. Also, the leader may have her hands full managing the opinions of the reviewers and may not need the extra hassle of offering opinions as well. Finally, as main contact to the vendors, it's useful for the leader to be able to say, "It's not my decision!" Whether she participates in the selection or not, the leader should take responsibility for the following:

- ⚡ **Doing or supervising the market canvass** that begins vendor selection. It may be easier for one person to do this, but it may be faster and more politic for a group to do it.
- ⚡ **Overseeing the first cut.** Although the reviewer group should make the cut, it may need a lot of help keeping to the process and closing on a decision.
- ⚡ **Creating the final set of selection criteria.** With ample input from the reviewers, the leader can still be responsible for forming the final selection matrix.
- ⚡ **Creating the RFP.** The mechanics and styling of the RFP document should come from a single source.
- ⚡ **Overseeing the process of creating scoring metrics.** Again, help and prodding may be needed to get a final agreement.
- ⚡ **Overseeing the selection of finalists.** Here, the impartiality of the leader really comes in handy.
- ⚡ **Deciding which technical drill-downs need to happen** and receiving the design diagrams that result. Diagrams result only from constant but amicable pressure. Of course, input to the decision should come from all quarters.
- ⚡ **Overseeing the vendor presentations.** You need an officiator of these events who understands both the vendors and the organization well enough to negotiate between them.
- ⚡ **Overseeing the final call.** You hope here is that facilitating this process doesn't go beyond process and prodding.
- ⚡ **Producing the decision report.** The selection leader is in the best position to create this report, but the content manager, or her equivalent, ought to be the one who circulates it and gets feedback from it.

Here are some of the other roles that you may need to fill in the selection process:

- ⚡ **Reviewer group.** Earlier in this white paper I discussed staffing the reviewers group, so I'm not repeating that here. I add only that you should aim for between four and 10 reviewers for optimum team performance.
- ⚡ **Scheduling** is a big part of selection. I suggest enlisting administrative help to make sure that all the vendors and reviewers are where they're supposed to be and know what's required of them at all points in the process.
- ⚡ **Technical drilldowns** require an appropriate analyst to lead them and participation from other analysts and subject-matter experts (implementers) who can drill as deeply as needed on a topic and help construct the diagrams.

Sorting Through the CMS Selection Criteria

There's a vast number of questions that you may want to ask a vendor. In the sections that follow, I outline a number of them. As you read through these questions, I'm sure that you can think of

some of your own and also, for better understanding, ways to reword the ones that I supply. Recognize that not all these questions are relevant to your needs, so choose them accordingly. Make sure, however, that you take into consideration future needs and ask the questions that may not apply today but that could conceivably apply a year from now.

These questions are the same ones that you must answer later to implement your system. In this white paper I call these questions *selection criteria*. After you've selected a CMS product, however, they serve as physical design constraints. You can use them to specify how your system is to be created from hardware, CMS functionality, third-party software, and customizations that you develop.

The selection criteria in the sections that follow ask about how certain functions are accomplished in a particular CMS. Use the logical design that you created previously to augment these questions. In addition to the following questions, use your logical design to explain what you're trying to accomplish and to ask vendors questions about what you want to do and why.

For each of your criteria, it's also important to decide what the ideal responses to each of your questions would be. Doing this makes it much easier for your reviewers to score by comparing the vendor's response to the ideal response. I include some suggested "right" answers to the criteria that I propose.

Business criteria

In addition to meeting the technical and process requirements you may have for your CMS, you want to make sure that the CMS product company that you decide to work with meets your business criteria for a successful relationship. After you have its product installed and part of your e-business initiatives, you're quite tied to the fortunes of this company. You don't want to find out then that it wasn't the company that you understood it to be.

Here are some of the business criteria by which you may want to judge CMS product companies:

History: You can ask vendors for a short description of the history of the company. Understand where they came from. Did they morph from a different industry (document management, for example)? Did they "ride a WebPub" from concept to product, bootstrapping from a service to a product company? Did they raise a lot of capital and burst onto the scene as a startup? It's less important which path they took than that you understand their history and how it may affect their future.

Business profile: You can ask for general business data, including the number of years they've been in business and have been producing the proposed product. What's the ownership of their companies? What are their capitalization strategies? What have their earnings been in the past and what kind of growth have they seen? You may be surprised at how small even some of the major players are. If you're dealing with public companies, you can certainly check their growth and analysts' guesses of their viability in the future.

Customer profiles: One of the best questions you can ask of vendors is how well they know their audience. If they can send you the profiles of the accounts that they see as key to their success, you can decide whether you're one of them. If, on the other hand, you're nothing like

the number of completed and ongoing installations they've done that are similar to yours. Which use the current version and which use earlier releases?

Next versions: Given that next versions are going to be needed (a near certainty), you should ask vendors about their upgrade policy, including how new releases, upcoming features, hot fixes, betas, and patches are communicated to customers and released. Ask how often to expect updates or upgrades and how much they cost.

Binding agreements: For the set of issues that you may come up with that aren't resolved in the current version but are promised in future versions, what sorts of binding agreements or penalties is the vendor willing to take on? These binding agreements are one way to ensure that the vendor is serious about what's in the next version.

References: Ask for references at the sites of the projects that are like yours. If possible, the references should include a business as well as a technical contact.

It's a lot to ask of vendors that they supply this much information (especially on top of the technical requirements, which are even more voluminous). As you negotiate the delivery (or lack of delivery) of these items, notice as much what the process is like as what's delivered. Is the vendor accommodating? Is the information available, or does she need to make it up? Do any vendors stall on items that they don't give you, hoping that you stop asking, or are they up front about why they can't deliver? Finally, do they just say what you want to hear, or do you detect a measure of sincerity and genuine desire to work with you? The impression that you get right up front is likely to carry through the rest of the relationship.

Overall criteria

The selection criteria in this section apply to the CMS as a whole. You find selection criteria sorted by collection, management, and publishing later in this white paper

Project process and staff

The goal of asking the questions listed in the table below is to assess not only the process that the company recommends, but also how well the company knows its own product implementation cycle. If it can provide you with this information in an organized and well-considered form, you can assume that it has the project process down.

Describe the general steps involved in an implementation of your product.

Does it seem overly simplified and does it diverge widely from the steps that I outline in

What factors are necessary to ensure a successful deployment?

Do the vendor offer any particular

seen and surmounted a lot of problems?

What's the typical time frame for a deployment and what can be done to minimize the time and effort required?

Does the vendor's timeframe include the full customization that probably needs to be programmed in or just general installation and some basic template setup?

Provide an example of project plans and deliverables from projects that are close in scope to ours.

The response tells you whether the vendor's completed projects like yours, how organized the project process is, and how strong the deliverables are.

Describe the various project roles (both your own staff and your customers') and the numbers of people involved in each role. Describe what roles people had before the implementation and how their roles changed after implementation.

How complete and realistic are these lists? Do they miss any categories?

What professional services does your company provide?

Many CMS product companies can provide full service, from design to maintenance. Each, however, has its own particular way of doing so.

For a project such as ours, what sort of professional services do you recommend?

You likely get a description of the standard services packages that the company offers. Try to get past this to the particular project composition that it sees for your project.

Can you provide a list of your service staff positions and their billing rates?

You can compare this list to the staffing list that I provide and your own staffing plans to decide what you may need.

Development environment

The questions in the table below should give you a fair appraisal of the kinds of development tools that are allowed and preferred in working with their product. Keep in mind that some of these criteria are fairly technical in nature, so if you're not too familiar with software development, make sure that you have a technical resource review this section.

What development languages and tools were used to create your product?

It's good if the company used the more common Web-ready tools, such as Java and C++.

Please provide an overview of the ways programmers can customize or extend your product, including the languages and tools needed.

Here you're looking for a clear description as well as the tools that you want to use. Nonproprietary development languages and tools are preferred if you don't have ones that you particularly want to use.

What APIs are provided as standard with your product?

An extensive, well-documented API is preferred. Best is that the company builds its own user interface from the same API so that you can make sure that it's well tested and provides complete coverage of the product's functions.

What skills and tools are needed to create and modify basic templates for authoring, submitting, and publishing content?

Basic templates shouldn't require programming skills. Ideally, the tools are those that a person who can use a graphical HTML editor can easily learn and use.

What skills and tools are needed to extend and integrate templates beyond basic content inclusion and layout?

Nonproprietary development languages and tools are preferred if you don't have ones that you particularly want to use. You may prefer layout tools that are targeted at the kind of skills that a graphic designer may have. If a programmer is required to create page layouts, you may have trouble.

What languages and tools does your system support for automating administrative functions?

You want to hear that all the administrative functions of the product, including workflow, user administration, database management, and reporting, can be called independently by you, using Web technologies that you want to use (ASP, JSP, CORBA, COM, J2EE, and the like).

What languages and tools does your system support for automating content authoring and acquisition?

At the very least, you should be able to modify collection Web-based forms and embed them in your own Web applications. It's better if the product offers out-of-the-box integrations with the authoring tools that you like. (Microsoft Word is usually the most important, but integration with HTML editors and graphic design applications is becoming increasingly common.) You should be able to extend and modify

you like.

Localization

Localization is a complex issue, often barely recognized by CMS products. If you have localization needs, first understand exactly what they are. Don't expect a product company to tell you how to localize. Given your plan, however, it should be able to tell you how its tools can help. The table below lists a series of localization-related questions.

Please provide an overview of how your product supports publication localization. What specific tools and processes do you provide?

The overview ought to tell you how well the company understands localization and how deep its support for it is.

How is translated content produced and synchronized with the source content?

Ideally, the system maintains linkages between translated components to manage updates. It may be capable of handling content in different languages, including support for double-byte character sets or Unicode, international date formats, and global currencies. The screens that the content maintainers use must be easily translatable into different languages

Are there any facilities in your product for multiple selection images or other media?

You may need various versions of graphics and other media that are localized for your various regions.

Please give a full example, with references and URLs, of a customer that has successfully used your system to create a global localization and translation solution.

A successful implementation by a customer who agrees to be a reference is a strong point in the company's favor.

Do you have any translation aids in your product, or do you integrate with any?

Translation aids help you put the master and translated versions of content side by side for comparison and difference checking.

Can you version and track changes at the element level?

It's really helpful to be able to send some component elements for localization separately from others that don't need to be localized.

Is there template functionality for sniffing the origin nation and redirecting?

The CMS may be capable of reading HTTP headers and helping you decide the region of origin of Web visitors.

Do you provide multilingual UI for content contributors and producers?

Your international contributors may need, and are certain to appreciate, anything that you can do to localize their CMS interface.

Training

The questions in the table below assess how complete, convenient, and affordable training on this product is.

Please provide an overview of the training that you provide.

You're looking to see a well thought-out response that shows that the company has put effort and resources into training and that it's professionally delivered.

What's the availability and cost of training, and how frequently is it offered?

You want to make sure that the training schedule doesn't affect your implementation timing.

Is training held at your site or ours?

Some may offer on-site training for large enough groups. Otherwise, travel costs need to be included in budget considerations.

What CMS staff positions do you train for?

There's almost always end-user and administrator training, but is there training for developers, managers, information architects, IT staff, or designers?

Technical support

Bad technical support can make you very sorry that you chose a product. The questions listed in the table below should help you assess what sort of support to expect. Remember to be reasonable in your expectations of the support that you receive. It's not reasonable to expect the company to provide one-hour turnaround on all requests, anywhere, anytime. Recognize also that the better the support that you require, the more expensive it is. The correct response from the vendor to unreasonable support expectations is, "We can provide whatever level of support that you're willing to pay for."

Please provide an overview of the technical support that you provide.

As with training, you want to see a well thought-out response that shows that the company's put effort

professionally delivered.

Do you provide support, or do you work with any partners?

If the company's small, it's better if it's partnered with a larger support organization that can bring its support up to enterprise levels. On the other hand, you want to see that the company can provide the specific expertise to the partner to augment the partner's ability to address problems.

What's the geographic coverage of support, and at what hours is it available?

Obviously, the company should cover the same territory as your organization does.

In what languages is support available?

This may be an important consideration for you if you're embarking on a global deployment.

What's the guaranteed response time for a support request?

How willing is the company to guarantee anything having to do with support?

Is there a single point of contact for support, or do people need to know whom to call for what?

A single point of contact with fast triage and routing to the appropriate contact is preferred.

What kinds of self-help facilities do you provide?

FAQs, known bugs and workarounds, and tutorials for beginners can save a lot of money for the support organization.

What guaranteed expertise is available on the first call, and what are the escalation procedures? How high in the technical organization can an issue be escalated?

The optimum cost-effective solution is to have enough skill on the first call to correctly route the problem. You want to see that, from the first call, there's problem tracking and a logical and reasonable set of criteria for escalating to each new level of expertise.

Describe your support licensing plans and associated costs. Do you offer a pay-as-you-go plan as well as contracts?

You may prefer a cost-per-use model that you can charge back to contributing groups rather than a fixed fee.

What comes as part of the

The extent of services for setup and the cost of

the costs associated with ongoing support after the initial purchase and installation?

consideration the "total cost of ownership," not just the initial purchase price.

Are there any provisions for downtime or loss of revenue from systems impaired by a CMS bug?

Depending on how mission critical your system is, this sort of seriousness may be necessary. A willingness to enter into this sort of agreement shows that the company has extreme confidence in its system (or isn't smart enough to know that these sorts of agreements put it at grave financial risk).

How wide is the support? What software and hardware is covered?

How are problems that may be outside the CMS software supported? Are extensions using the standard API supported? Are template bugs supported? Are platform, operating system, Web application server, and database and network connectivity issues supported? For issues that aren't supported, how is it determined where the problem lies and how does the staff then tie into the appropriate other support organization?

Deployment

The questions in the table below help you understand how the product can be rolled out in your environment.

What operating systems does your product run on?

The right answer is the operating system that you use!

What operating systems are required on the client machines for the database client and authoring tools? Describe any client-side software needed to use the system.

All basic functionality should be available through a Web interface so that the client needs only a Web browser. Client software should be available, but not required, as a supplement to a browser interface to enable a richer and more responsive UI.

Which Web browsers are supported for authoring, management, and administrative functions?

You hope that it's the ones that you use.

Describe the skills and size of the team needed to successfully deploy and maintain your product.

An open-ended question that you can use to assess the company's process and savvy on this issue.

What models do you employ for distributing the management and publishing environments across business units that aren't in the same facility?

If you need for people to contribute from around the world and for them to publish locally, you want to assess how the CMS can help you keep this distributed environment organized and minimize redundancy.

Is your system best deployed inside or outside a firewall? Describe the difference in requirements or performance if your system is deployed inside or outside a firewall.

This question helps get at the distinctions of using this system for an intranet vs. an Internet site. It also uncovers any problems that you may experience with remote contributors accessing the CMS.

Describe the process that you'd use to deploy your system across an organization with a structure such as ours.

Another open-ended question to see how well the company can understand and provide for your specific environment. You want to add a list of the significant factors that are present in your organization.

Collection criteria

The criteria in this section can help you assess the collection technologies and processes that come with the product or can be integrated with the product.

Authoring criteria

CMS products vary widely in their support for authoring. The questions in the table below should give you a way to compare each of the products that you review.

Do you ship with any specific authoring tools? If so, which one(s)?

Only a few CMS products ship with authoring tools (word processors, image-creation tools, HTML editors, and so on). Although it's helpful to know whether the product includes these tools, it's probably more important to find out whether the product supports the tools in use by your organization.

With what third-party authoring tools do you directly integrate?

Direct integration means that some aspect of the CMS appears within the authoring tool. Some CMS products create a menu within a particular authoring tool, for example, so that you can save content to and retrieve content from the CMS repository.

What file formats can your CMS repository store?

Many CMS products can manage files that are

don't manage the content within these files. In other words, the CMS serves as a document management system for files that it can't interpret.

What file formats can your product recognize and convert to XML or database records?

In this case, you're interested in the product's capability to open foreign file formats and parse them. The result is generally either XML that can be stored in an XML repository or relational database records with the full content of the source file in them. Some products, for example, take the text that you write in Microsoft Word and convert it automatically into XML for storage in the product's repository.

Please describe how a novice content contributor could author the textual content for a Web page and include graphics or other types of media to go along with the text.

You want the system to be simple enough for your beginners and casual authors to create simple pages with little or no help.

Please give an overview of your Web-based form system for content input.

Almost all products now have a Web interface for content entry. Some may have a non-Web system as well that has fuller functionality than the Web-based forms.

Do you enable rich HTML or XML editing within the fields on your Web-based forms?

Some products include a little editing environment right in a Web-based form so that authors can format information as they type it.

Can Web-based forms show different sets of elements based on the user's permissions?

In other words, can you personalize input forms so that different people see different parts of the same content component, based on what you want them to enter?

What support do your forms offer for linking and uploading images and other media?

If this is a common need within your organization, you want to ensure that it's easy to add media and associate it with textual content.

Can your forms be automatically created from a DTD or database schema?

Some products autogenerate Web-based forms from the structures that you've previously defined, saving you a lot of work.

extended or modified by additional HTML or other programming?

you want to customize forms and need additional features. Ideally, the forms can be customized by using a nonproprietary programming language.

Can your forms represent hierarchical relationships?

All products enable you to put a set of fields on a Web-based form. A few also enable you to nest some fields within others. This is really handy for entering the titles and text in sections and subsections.

Can authors create content while they're not connected and then submit it later?

In other words, can you use the CMS authoring tools "offline," a handy feature for authors who don't work on-site or for contributors who travel frequently?

Can authors preview content as it may appear in publications, before they finally submit it?

A preview function is extremely helpful and streamlines content creation and quality assurance workflows considerably.

Can authors view a revision and workflow history for their work?

An audit trail of workflow for each component is helpful for accountability as well as content troubleshooting.

Can authors perform a spelling check on their work?

A spelling check feature isn't yet mainstream among CMS products, so keep asking for it and eventually it may become mainstream.

Describe the training expectations required for new content contributors.

The system should require minimal training for novice or casual contributors. You can expect some training requirements for more advanced authors.

Acquisition criteria

The questions in the table below can help you decide how well a product meets your needs to gather and process existing information.

With what acquisition tools does your product ship?

Some products include advanced tools for scanning content sources and automatically capturing content. Some have syndication tools that enable you automatically to accept incoming syndication feeds.

With what third-party acquisition tools do you integrate?

Most of the larger companies have strategic relationships with other companies (often, content syndicators) to augment their abilities to acquire content automatically.

How would we migrate our existing Web sites and other publications into your system?

Is there an automated process to import an existing Web site, or does it require an in-depth content analysis and restructuring into predefined content structures? Are there any tools at all for migrating publication types other than Web-based ones?

Can your product help manage the scheduling, usage rights, attribution, and payment for acquired content?

There's probably not much beyond the standard workflow system that the company can offer, but the question may get its people thinking.

Can your product automatically retrieve records from a database available locally and turn them into content components in your repository? Can these downloads be scheduled to happen periodically, with no human intervention?

The company may depend on standard programming to accomplish this, but it should demonstrate that its people understand the issue and have worked through it. You may have a product catalog database, for example, and want to create new components after new products are added to the database.

Conversion criteria

It may be that a product comes with few advanced conversion tools. The CMS product company, however, should be able to answer all the questions in Table the table below if the company is aware of and well versed on conversion issues.

Does your product ship with any conversion tools? If so, please describe their capabilities.

Products really vary on this count. Some have no converters at all. Others have a wide range of built-in tools for text and media conversion.

With what third-party conversion tools do you integrate?

The answer to this question ought to give you a feeling for how well the company understands conversion. Does it have any partners there? Does it have an idea of how conversion fits into the process?

How does your product

As pages are requested, can the product retrieve

conversion of text and media on a Web site?

you can store your media in one format and have it converted only if it's requested in another format. This isn't typical functionality, but some products can do it.

Aggregation criteria

When I first proposed a list of metadata management tools like the following one, very few product companies even recognized that it was an issue. Today, they're beginning to see the point of managing the metadata system. Over time, if you don't manage it, the system becomes unwieldy, with conflicting lists and values, and stops working as a way to unite all your content. Although the questions in the table below are still difficult for many product companies to answer even today, I suspect that, over time, most of or all these metadata capture and management features will become standard equipment in enterprise CMS products.

Does your product include or integrate with any metadata management tools?

Some products include auto-discovery and taxonomy creation tools that help you design metadata lists and automatically find metadata in the files that you process.

Can you automatically apply metadata values to content based on its type or location?

Some products enable you to choose a directory or other branch and batch-apply a metadata value to it. In addition, some can supply default values for metadata fields, based on the type of content being tagged.

Does your product have any tools for helping the administrator combine or split metadata fields that currently exist in the system?

As you move forward, you inevitably need to combine two similar metadata fields into one (for example, you may decide to merge an author field and a contributor field into one field) or split one field into two. (A single status field may need to be split into a publishing status field and an archive status field, to give you more control.)

Can you define mandatory and optional metatags?

This may be done in the Web-based form interface, but it may also be possible to do it globally for all uses of a particular metadata field.

Can you pull metadata lists from other sources, such as a database or a flat file?

You want to see, for example, if the product enables you to draw your list of products automatically out of a database and present them in a Web-based form.

Can you write metadata lists to other external files and databases?

You want to see whether the product enables you, for example, to add to the list of products and then periodically export the updated list to the database in which it originated.

Can you automatically bring up metadata lists or values for periodic expiration or review?

As part of managing your metadata system, you need to periodically review the validity of lists or values and either update or delete them. If you do, it's nice to have a tool that helps you update all the content that's affected by the change that you're making.

Does your product tie into any accepted standards for metadata?

The Dublin Core Metadata Initiative and a host of other industry-specific metadata systems are being created and revised. At some point, you want to use one or more of them. Can the CMS help you, or does it stand in your way?

Does your product support all types of metadata types in the repository and metadata fields in your collection forms?

Management criteria

The questions in this section can help you understand how the product stores and administers content and how it creates the infrastructure of a CMS.

Storage

The questions in the table below address the basic repository structure of the CMS.

How do you store content? In a relational database? An object database? A file system? A hybrid of the preceding types?

You want to know how the company structures its content storage to know whether it's compatible with your infrastructure requirements and can handle the sorts of content that you intend to manage.

Describe how your system stores the following content relationships: multiple hierarchies, multiple indexes, cross references between components and between publication pages, and sequences.

These are all the access structures that you need to store and access. All decent CMS products enable you to build a hierarchy. See how well the products that you survey do at building more than one hierarchy for the same set of components. You may need this facility if you intend to build multiple publications, each with its own TOC.

Does your content repository support extended character sets

This is a must if you intend to localize in Asian or other double-byte languages.

international date and currency formats?

If your repository system supports XML DTDs or schemas, can you create one DTD for the entire repository? Can you create multiple DTDs that are enforced for part of the overall structure of the repository?

Does your product have the capability to check the validity of links and media references that appear within text fields in the repository?

In many systems, the hyperlinks and media references (such as an tag in HTML) that are embedded in the middle of a block of text go unnoticed and can be broken without any clue. Some systems enable you to track these links and references even inside paragraphs of text.

What indexing and searching does your repository support? Does searching respect security?

A good answer includes rich indexing and querying, metadata (keyword) searching, full-text searching, proximity searching, and relevance ranking of search results, as well as the capability to exclude results that a user doesn't have rights to see.

Administration

Any CMS that you buy should do most of what I point out in the table below. The differences are in how well it performs these tasks and how convenient it is.

In what ways can your system enable nontechnical business users to manage their own content?

This is an open-ended question designed to give the company a chance to address ease-of-use issues and how it caters to novice or casual users.

Please describe the utilities provided for system administration and operations personnel to effectively manage and support the system. Please identify any administrative functions, such as configuring workflows or publishing schedules and so on, that can't be accomplished through Web-based tools.

Ideally, all administration should use Web-based tools. If there are some administrative functions that require other software, you should know about them and make sure that they work in your context.

Can you do a global search and replace throughout the repository?

The CMS should support search and replace through free text and metatags, assuming that appropriate permissions and security rights are observed.

Can you automatically verify and manage broken internal and external links?

This is a nice feature to have.

What user roles and rights model do you use, and what tools do you provide for managing and manipulating them?

There are many variations on how user roles and rights are configured, including those that tie into your own network directory services and ones that support "inheriting" permissions based on a hierarchy of user roles.

What's your approach to data synchronization and replication across multiple databases? At what level is synchronization managed and communicated?

It's unlikely that it does a lot of this automatically. Instead, you're most likely to get replication tools that you must control to ensure synchronization.

Workflow

CMS products vary from no workflow support to integration with third-party tools to strong native workflow modules. The questions in the table below can help you correctly place the products that you survey on the spectrum.

What workflow tools does your product include?

You're looking for the vendor's description of its workflow tools. You may end up with a paragraph or two of marketing information or a good overview here.

With what workflow tools does your product integrate? What support do you provide for the integration?

Even if a product has internal built-in tools, and especially if it doesn't, it's important to know whether it enables you to use strong third-party workflow systems. If you're thinking of integrating in a workflow system, make sure that you get the company to answer all the questions in this table from a support perspective, as opposed to an implementation perspective. You want to make sure that events and content status transitions in the CMS can correctly trigger events in the external workflow system.

Please describe the internal or external inputs that can be used to trigger workflow steps.

Look for the capability to trigger workflow steps by an external process, such as an e-mail message arriving or a file being added to a directory.

Please describe how your system handles the configuration of user roles for workflows.

You want to see that the product has strong user interface that gives administrators complete and easy access to the functions that the product offers.

Do you support parallel as well as sequential workflows?

In a parallel workflow, two people work on an item at the same time. In a sequential workflow, one person finishes before the other begins.

Please describe the range of system activities (such as content submission or deletion) that can be supported by workflows.

You want to find out whether the system can handle requests for content, content submission, content addition, deletion, edits for existing content and content bundles, content reviews, content approvals, content migration, migration approval, and so on.

Please describe the content or system attributes that can be used to drive the rules for automated workflow routing.

In other words, what can the system look at in deciding what to do next with a piece of content? (Submitter, content type, and status are some of the usual attributes.) The best answer is that the workflow module can use any piece of metadata that you create.

Please explain or demonstrate the process for creating or modifying a workflow process.

You want to make sure that creation of workflows doesn't require programming skills. The slickest systems have a visual workflow designer with drag-and-drop manipulation of processes. Make sure that either the interface is very complete or that it's extendable.

What types of content collection, content management, content publishing, and administrative activities can be handled by an automated workflow process?

You want to know whether the workflow module can handle all authoring and publishing requests, including content submission, content addition, deletions and edits, content review/edits, content approvals, content migration, publishing, and testing.

Do you have the capability to include an external service

If it's an issue for you, make sure that the workflow tools can manage people with only a slow Internet

graphics, or translation services) in the workflow?

through your firewall, or only an e-mail connection to you.

Can you view all content that's in a particular state for a particular user?

The best systems have rich queues that show each user what task she must accomplish.

Can you add comments at the end of each workflow stage?

This is a handy feature to have. (If not built-in, you probably want to customize the system to include a comments field.)

Do you maintain an audit trail for all workflow processes?

It would be super if you could roll back a piece of content to any step by selecting that step in the audit trail.

Using your product, can you integrate e-mail into the workflow processes?

Some systems can run the entire workflow process through e-mail. Users receive notifications in their inboxes and then click links in their messages to complete tasks in the CMS.

Is there an administrative override to automated workflow processes?

This feature is a must if you need to stop or redirect workflows in process.

At what content level can you create workflows?

Most systems can attach a workflow to a file. The better ones can attach a workflow to a component (or whatever name are given components in the product). The best ones can attach a workflow to a particular element of a component. The body element, for example, can have a localization workflow attached to it to make sure that it gets translated. The rest of the component may not need to participate in that workflow.

Management integrations

In all questions about integration, it's important that you don't settle for the typical answers, such as "We have a partnership with X," or "I think someone has done that." Rather, ask for the documentation describing how the integration was done. If that fails, ask for the name of a person who's done the integration and talk to her about how it went. At the very least, get specific information about the system on which the integration was done to make sure that it's similar to your own.

Here are some of the external systems with which you may want to integrate:

✍ ✍ Document management systems

- ⚡⚡ Media and asset management systems
- ⚡⚡ Operating system user registries (such as Microsoft Windows 2000 Active Directory)
- ⚡⚡ Enterprise resource planning (ERP) systems
- ⚡⚡ Custom databases that your organization uses for catalogs and other content storage

For each type of system with which you expect to integrate, tell the vendor the name of the product that you're using and what platform it's running on, and ask the vendor to tell you specifically how the product interoperates with yours.

The table below lists some other integration questions that you can ask.

What capabilities does your system have out of the box and with additional programming to connect to and share data with other databases?

The best case is a nontechnical interface for connecting to simple databases that can be used by administrators, plus the capability for the connection to be extended by programmers using a standard language.

Can you launch other applications from within your system? If so, how? What outside applications are supported?

Many products integrate with Microsoft Office products and Adobe Acrobat. The CMS product also may be capable of launching any viewing and editing tool that the system recognizes based on a file extension.

Do you follow any open standards for data access and data sharing?

Some products follow the XML schema or other open standards such as ODBC for data exchange.

Version control

Not all CMS products have a version control module. For those that do, the questions in the table below can help you decide whether it can work for you.

Can users check content into and out of your repository?

This is the most basic form of version control.

Does your product save earlier versions of content that can be restored later if necessary?

Make sure that it's reasonably easy for a user with permissions to restore the previous version without the need of a database administrator.

What level of granularity is supported for checking content in and out?

File level is most common; component level is the most useful; and element level check-in and check-out is necessary for more complex tasks, such as localization and collaborative authoring.

How is locking handled if content is checked out?

In the best case, you have the option to lock the component being checked out and, if you want, also its siblings or parents in the repository hierarchy.

Describe the different levels of granularity at which content can be rolled back to an earlier version. How do you support rollback of a complete site, an individual page, or a single content component?

You may need any of or all these kinds of rollback functions.

Can you perform differencing between two versions of the same content?

The best system enables you to put two versions side by side and compare each change.

Reporting

There are any number of reports that you may want your CMS to provide. The table below has some of the questions that you can ask to find out what sort of reporting a product supports.

Describe the logging capabilities of your system. What types of activities can be monitored, for both content producers and consumers? How would you track usage by user, group, or geographic area?

In the best case, you can track and log users, their time of access, and actions performed (such as downloading a particular version of a file or checking out/editing a file). You can also track system usage by user, group, geographic area, and so on. Logging and analysis usually can be performed on a Web server by third-party tracking products. You can use these programs for tracking the use of your Web publications. In addition, you should know what kind of logging the CMS provides for staff activities.

Describe the workflow tracking reports that your system can generate.

Workflow reports that are most commonly used show each component or document and its workflow status, along with other supporting information. Or you may want reports that show all components that are at a particular workflow stage.

Can your product track usage and fees associated with copyrighted media elements?

If you buy media, you probably need to track and report on where it was used and how many times it was accessed or downloaded. This is rarely part of the product, but it ought to be possible to integrate into the product.

Can your product generate a complete site map of the Web sites that it produces, including page names and statistics (size, components, and so on)?

These sorts of reports are very useful to publication teams.

Disaster recovery

If you ever have a catastrophic failure of your system, you're going to be glad that you asked the questions in the table below.

What features does your system support for content archiving and disaster recovery?

Features to look for are versioning, rollback, and automated backups. These features may be part of the database software that the CMS uses but could be controlled from the CMS administrative interface.

Can you provide any statistics on the failure rate of your product, such as average time between failures or the average percentage of system availability?

Some can provide this information and others can't. It's a good question to ask as you get to checking references.

Describe any functional and architectural features of your product that are optimized to maximize the reliability of your system.

This is an open-ended question to give the company a chance to brief you on its reliability. Some companies can provide technical white papers that describe their system architecture and reliability features.

Security

The questions in the table below can help you assess the user and access features of the CMS.

Describe how your system can leverage existing operating system or directory services security profiles to avoid needing to re-create definitions and duplicate security maintenance. With what servers and services can you integrate?

In the best case, whatever permissions or user groupings are already available in your operating system also are available and extendable in the CMS.

Do you support a hierarchical security model?

Especially if the system doesn't tie into existing security systems, you need the capability to group users and have them inherit permissions from parent groups.

Can your product help us enforce intellectual property rights and security permissions by controlling the capability to publish media?

If you work with content that has restrictions because it's copyrighted or because it's secret, the CMS may enable you to create and enforce a policy.

Performance

The questions in the table below help you assess how well the product can scale to the size that you need it to be.

Describe the minimum and recommended hardware, software, and network requirements of your product at the level of usage that we expect.

The trick here is to specify your level of usage. Usage has two parts: the usage of your Web publications and the usage of the CMS by your staff. Check both.

Please provide examples and references that demonstrate your system's capability to scale to an enterprise-wide, global implementation. These examples should illustrate your capability to scale to multiple instances on multiple servers, to scale beyond a single Web farm, and to fully support load balancing.

If this is an important criterion for you, asking this question provides an opportunity for the company to present you with its scalability story. Check the story that it provides for architectural validity and applicability to your situation.

Describe any data size or performance limitations of the content repository. Include any known limitations, such as maximum number of rules, number of unique components that can be stored, number of concurrent content contributors that can be supported, or number of concurrent content consumers that can be supported.

Make sure that you check these capacity limits for staff as well as for concurrent audience users of your major Web publications.

Describe your mechanism for content replication to regional sites.

Replication usually consists of a master version and copies that are distributed regionally to improve performance.

Publishing criteria

The questions in this section help you assess how many of your publishing needs the product can cover out of the box or with customization.

Templating criteria

The questions in the table below canvass the product's templating capabilities.

Describe your overall approach to publication templating?

This is your chance to get an overview.

What programming languages can be used to develop Web templates?

The right answer is the languages that you use. Barring that, open languages that have a wide skill base in the programmer world are the best choices.

Do you support developing templates for non-HTML publications?

Sadly, most don't. One day, CMS products will support templates in word processing and graphical layout program formats as well.

Can your system process templates in a batch mode, producing complete static sites?

In other words, can the CMS render a static, Web-based publication as well as serve dynamic sites?

Can your system process templates on demand, producing dynamic sites?

Not all products can do this.

Can your system mix batch and on-demand processing to produce sites that are partly static and partly dynamic?

This is the ideal. You want to produce as much of a publication as possible statically for performance and robustness, but still produce other parts dynamically for the flexibility that it offers.

Describe the methods that you support for dynamically creating site navigation components. How are searches, links, a table of contents, or other navigational aids built by logic in the templating system?

You want to see that there are some tools for this but, moreover, that the system is open enough to enable you to extend it to whatever navigation that you may want to produce.

Does your system support the concept of component and navigation templates? If so, how?

Do your templates support integration with outside systems? If so, how?

Some tools may be provided, but you really want to make sure that the CMS

yourself.

How is personalization accomplished in your templates?

This is a very open-ended question that ought to yield a general discussion of the product's personalization tools and integrations.

Content deployment

CMS products have a variety of capabilities for distributing content from the central repository behind the organization's firewall to servers inside and beyond the firewall. The content deployment questions in the table below are weighted heavily toward the Web because that's where most deployment happens today.

How can your system be used to deploy content files to remote Web servers?

This is the basic deployment functionality. Surprisingly, many products expect you to do this with third-party products.

How can your product be used to deploy database records to remote databases?

In addition to files, you may need the CMS to send data to databases outside the CMS repository.

Describe or demonstrate a scenario illustrating how content is deployed from a staging server to a production server.

A staging server enables you to test a Web site before it goes live. Find out whether the CMS can support staging. (Also notice that many products' licensing models require that you purchase separate licenses for each server. The staging server counts as one of those.)

How can your system deploy appropriate content to various destinations, based on properties of the content?

You may, for example, want to key on various values within content components (language, for example) to determine which server to deploy the content to. Can the CMS help you here, or do you need to write your own routines?

Does your product enable you to assign default values for file directories based on content type?

It's useful to have all pages of a particular type deploy automatically to a particular output directory.

Can your product automatically deploy files referenced in a field?

If a file (an image, say) is referenced inside text that's being published, it's a good idea to have the CMS deploy the file to avoid a broken reference.

Can you set the publish and expiration dates of content to be published?

It's useful if you can have the CMS automatically release and remove content based on the rules that you establish.

Can you publish on a set schedule without manual intervention?

This is what you want for small, frequently published material (weekly e-mail broadcasts and syndications to other systems, for example).

Publication platforms

With the questions in the table below, it helps to ask for specific examples of users who are doing the kinds of publications that you want. Don't stop with the answer, "Sure, we can do that."

Can your system produce broadcast e-mail?

The system should be capable of integrating to your e-mail system and pushing content to it. Alternatively, it may integrate to a third-party bulk e-mail system.

Can your system create individualized e-mail messages?

As opposed to bulk e-mail, in which the same message is sent to a lot of people, individualized e-mail is where a lot of different messages are sent to one or a few people each. This capability ties the personalization functionality to the e-mail creation functionality.

How can your system be used to produce print publications?

You want to focus this question more on the specific kind of print publication that you need to produce.

Can your system create publications based on WAP and other XML-based formats?

Again, have a good idea of the exact publications that you want the system to create.

Can your system syndicate content?

Syndication capabilities include creating format-neutral (generally XML) content chunks and deploying them to FTP sites, remote servers, or to mass storage devices (hard drives, CD-ROMs, DVDs, and the like).

Can your system create content for proprietary electronic formats, such as Microsoft Help, Macromedia

You should query specifically for any non-HTML electronic platforms that you may need.

platforms?

Can your system create downloadable documents in Portable Document Format (PDF) or other proprietary formats?

PDF is good for downloads that can't be modified by the user. Query for any others that you may need.

Personalization

The questions in the following table help you figure out the types of personalization that you can expect to get from a product.

Does your product ship with any personalization tools?

Some don't, although most don't easily admit it. Others have extensive offerings.

Does your product integrate with any third-party personalization tools?

Most at least have marketing alliances with personalization companies. Look for documented integrations.

What kind of user interface do you provide for creating and modifying personalization rules?

Some products have very well-developed point-and-click interfaces for rule building. Others require a programmer to code the personalization rules - not a good long-term solution.

Describe or demonstrate the process for defining personalized experiences. What skills and training are needed to work with your personalization engine?

See whether you can get a concise description of the overall approach to personalization that includes a discussion of how it's actually accomplished.

How does your system facilitate the collection and storage of user profile data?

Some products have functions for monitoring the user's actions and their responses to questions and then storing the results in an accessible profile - but don't count on it.

Can your product receive syndicated data from third-party suppliers of customer profiles?

You may want to supplement the information that you collect about your users with data from commercial suppliers. Can the CMS help this happen?

your personalization engine?

HTTP header information, page history, all content metadata, user profiles inside or outside the CMS repository, user groupings, and personalization rules.

What facilities do you have for producing individualized print publications or other non-Web personalizations?

Most of the action today is in dynamic HTML page building. The same tools could be used to make any other kind of personalized publication, provided that there's template support for the publication format that you need.

Do searches, links, a table of contents, or other navigational aids reflect the personalized content that's available to a specific user? If so, how does it work?

For relatively sophisticated personalizations, the system should be able to modify these access structures automatically, based on the personalization rules.

What sort of personalization rule-building process and user interface do you employ?

Some products have well-developed user interfaces for point-and-click personalization rule building.

Can your system perform collaborative filtering?

Collaborative filtering is the process where the system recommends content based on requests of similar users (an approach popularized by Amazon.com).

Please outline the kinds of personalizations that are possible based on user profiles.

This is an open question that gets the company to talk about how personalization interacts with profiles. The product may enable you to define profiles, for example, and then target content to them.

Please outline the kinds of personalizations that are possible based on a user's actions on a Web site.

This is also called behavior-based personalization. Some systems track a user's behavior and modify the subsequent pages accordingly.

Can your system facilitate the creation or running of push campaigns?

In a push campaign, the system actively presents certain content on a Web page or in another form to a group of selected users for a specified period of time.

Publication integrations

There are a lot of systems to which you may like to tie your Web publications. As with the management integrations, make sure that you get as much detail about the ones that you care about as the company can provide. The following list features some of the systems to which you may want to connect. It may be that the CMS does some of these functions as well, so phrase your question so that the CMS product companies tell you how much of the function their products do, as well as what third-party products they integrate with if you want to integrate with one or more of the following systems:

- ✂✂ Ad management and server products
- ✂✂ Brand management tools
- ✂✂ Campaign management and promotions systems
- ✂✂ Merchandising systems, including up-selling and cross-selling tools
- ✂✂ Customer relationship management systems
- ✂✂ Lead generation products
- ✂✂ Surveys and questionnaires
- ✂✂ Web access and security tools
- ✂✂ Data logging and mining tools
- ✂✂ Product configurator systems
- ✂✂ Decision tree tools
- ✂✂ Download packaging tools (which automatically bundle together and compress electronic content for efficient download)
- ✂✂ Chat products
- ✂✂ Bulletin board and threaded discussion systems
- ✂✂ E-mail acceptor and automated response systems
- ✂✂ Event planning and registration tools
- ✂✂ Webcasts and streaming technologies
- ✂✂ E-commerce and financial transactions systems
- ✂✂ Credit card transaction systems
- ✂✂ Any of the enterprise systems that I mention in the management integration section, earlier in this white paper that also need to be accessed dynamically from the Web
- ✂✂ Conversion systems for on-the-fly conversion of content (for example, from XML to HTML)
- ✂✂ Custom database applications and custom programming objects that you may have created or may use on your Web site
- ✂✂ Web servers and Web application servers

Web UI

The Web is guaranteed to be one of your publication platforms, and the table below lists questions that can help you assess how much help you can expect from the CMS product in building Web sites.

What types of searching and indexing capabilities can be made available to an end user viewing a Web site produced by your system?

Searching isn't necessarily performed by the CMS. For static sites, a standard third-party search engine may be fine. For dynamic sites that are produced by the CMS on the fly, you need integration between third-party search engines and the CMS search functions.

What particular support do you have for producing browser-independent Web pages?

Some products include browser sniffers and supporting code to make serving browser-specific pages easier.

Does your product support user-session management, including cookies or database caching of user profiles?

This is important stuff for personalization and customization.

Web architecture

If you have the CMS active on a Web site, ask the questions in the table below to see how the product integrates into your Web infrastructure.

Which Web servers are supported by your product?

Yours, you hope.

Does your product include a proprietary Web application server?

Some products still do have their own application servers. You want to make sure that the product doesn't lock you into its own application server but enables you to integrate to one of the better commercial application servers.

With what Web application servers does your product integrate?

For any product that you expect to use for more than the next year, the company needs at least to have plans for how it intends to integrate it with industry-standard application servers.

What databases can your system use?

The right answer is the ones with which your technical team is familiar.

How does your system facilitate or perform content caching?

Caching can happen at many levels. Make sure that you understand the difference between the caching models used by your various contenders. Make sure, too, that the company can

commensurate with the amount of traffic that you expect.

How does your system facilitate or perform database connection pooling?

This may be handled completely outside the CMS. Know which parts are performed within the CMS and which parts depend on a third-party application server.

How does your system facilitate or perform server load balancing?

You'd like at least some support from the CMS for this.

How does your system facilitate or perform user session management?

Does the system rely on cookies, or does it provide other methods of tracking user sessions?

Can your system leverage user-profile information stored in directory services? Is any custom development needed to implement this functionality?

Probably there's some integration effort needed.

List the LDAP servers with which your system can integrate.

This is a concern if you're using LDAP services in your environment.

Summary

If it's done well, system selection can galvanize the entire organization behind a platform and approach to your CMS. If done poorly, it can create wide rifts between groups and result in a system that was nobody's favorite. To keep the selection process on the good side, do the following:

- ☞ ☞ Understand the position of product companies. Trust the people who are trustworthy but understand the constraints and goals that guide their behavior.
- ☞ ☞ Develop a process before you need it. If you're in the heat of decision, there may be too little energy and good will to yield a fair and binding selection process.
- ☞ ☞ Develop a set of selection criteria that ask the key questions that you need answered to compare and choose products.
- ☞ ☞ Be diligent. Rather than jumping ahead, start as wide as possible and continually narrow the field until only one choice remains.
- ☞ ☞ Be official. Include the people from outside the process who need to understand and support the process as well as those inside the process who decide. Herald your final decision and your great process.

✍️ Update your ongoing project deliverables, such as your project plan and your risk assessment.