

GRANDON GILL, KATHLEEN LONG, DENNIS WALPOLE

USF WEB CONTENT MANAGEMENT SYSTEM¹

Once we decide on a CMS, then the fun really begins...

Kathleen Long, Director of Web Services Information Technology at the University of South Florida (USF), reflected on the issues that she faced in spearheading the content management system (CMS) for the institution's website. Once completed, that system would virtually define the university's public presence—replacing a hodgepodge of locally generated web pages developed by IT staff, faculty and even students. The new system would allow USF to better brand itself. It would eliminate the duplication, obsolescence, and inconsistency that could frequently be found in the existing site. In short, it would convey the quality of the university to potential students, parents, the community, and other critical stakeholders. Such a transformation was long overdue.

Getting there was the challenge. The decisions that needed to be made were both short term and medium-to-long term. In the near future, the IT group—in close consultation with its stakeholders—needed to decide upon a CMS vendor. Nine different systems had been considered. These varied greatly across many dimensions, which included open-source versus proprietary platform, underlying technology (e.g., Java, PHP or Microsoft's .NET), level of experience with large university sites, and, perhaps most critically, the level of support available. How should these different characteristics be prioritized in the decision-making process?

Over the longer term, she faced the question of how the system would be implemented and what policies would need to be established. A research university environment was very different from a typical business, or even a teaching-focused educational institution. Faculty members, many of whom had tenure and might well be stars in their own fields of research, often had created their own content—even entire websites. Attempting to force such individuals to change was likely to be a losing proposition. How, then, could they be incentivized for the good of the institution as a whole? An extensive network of regulations governed what information could, and could not, be accessible on the public site. How could compliance be assured? USF was highly unionized. They could become very prickly if their voice was not heard. And then there were the students. Would a tightly or loosely controlled system best meet their needs? A variety of policy options for the new site could be considered relating to mandatory, voluntary, and excluded uses. What would a sensible set of policies and implementation plan look like? The only thing about which Long could be certain is that it would need to be the product of consensus.

¹ Copyright © 2012, *Informing Science Institute*. This case was prepared for the purpose of class discussion, and not to illustrate the effective or ineffective handling of an administrative situation. Names and some information have been disguised. Permission is granted to copy and distribute this case for non-commercial purposes, in both printed and electronic formats. This material is based upon work supported by the National Science Foundation under Grant No. 1043919.

Content Management Systems

A content management system is an application intended to manage a website or a collection of subdomains. To understand the role of such systems, it is useful to look at how the Internet has evolved over the past two decades.

Evolution of the Web

The World Wide Web (WWW or “web”) describes that segment of the Internet that operates using the *hypertext transfer protocol* (http), originally developed by Tim Berners-Lee in the late 1980s. Although the original web incorporated many innovative ideas, two of these in particular stand out: the use of the *hypertext markup language* (HTML) to represent documents and the ability to link documents together using hyperlinks. These advancements represented a major improvement over earlier systems for indexing and delivering documents, such as the *gopher* system named after the mascot of the University of Minnesota. Originally developed for pure text documents, the web experienced an unprecedented surge of popularity after the Mosaic browser, which displayed images, was introduced.

Very soon after use of the web began exploding, it became clear that care had to be taken to manage websites—collections of HTML documents hosted on a single domain (e.g., www.mydomain.com). It was extremely easy for erroneous or obsolete links to be contained in such sites, leading to a poor user experience. For smaller sites, one approach to addressing this problem was to use application software such as Microsoft’s *FrontPage*, which allowed the site as a whole to be displayed as a system of pages, with links being updated automatically as pages were moved, added, or deleted. For large sites, such as enterprise-wide sites, special tools were used for site building and, quite often, responsibility for individual subdomains was decentralized to business units or departments.

By the late 1990s, a new type of content—referred to as dynamic content—increasingly came to dominate organizational websites. This content was typically a mixture of HTML code and programmatically constructed code that incorporated information from databases every time a web page was displayed. Technologies such as Microsoft’s *Active Server Pages* (ASP), Java programs deployed using the J2EE enterprise platform, and pages using the PHP hypertext processor are all examples of approaches that could be used to develop such content. This growth in capability further increased the need to manage the complexity of a website. Web development was becoming less of an art and more of an engineering discipline. Many of these websites were also intended for use within organizations, producing a schism between internal and externally facing websites.

Shortly after the millennium, a new type of web application rapidly began gaining popularity. While there had always been a social dimension to the web—one site, *Geocities*, hosted millions of personal websites and communities—a new class of application was being developed that allowed users to add content to a stream of communications that then appeared on a website. Blogging sites such as *LiveJournal* proliferated. Discussion group sites, such as those hosted by Yahoo!, boasted millions of users. Collaborative sites, such as *Wikipedia*, permitted experts to post and review information. Sites for sharing personal information, such as Friendster, MySpace, Facebook, LinkedIn and many more fueled an era of dynamic, interactive social computing, sometimes referred to as Web 2.0.

These innovations presented their own special challenges. Because dynamic and social content were commanding the most attention, the more static pages that characterized the original web—and still played an important role in conveying information—continued to be added to sites, often with little or no central coordination. Because the sites required no attention once put in place (unlike dynamic and social sites), they typically remained in place long after they were obsolete. Even when they were taken down, search engines such as *Google* would often save the sites and pages they found (a process referred to as

caching), meaning that the life of a particular web page was effectively indefinite. For many organizations, this led to websites whose overall form was chaotic, and whose function was marred by numerous defects, such as browser inconsistencies and bad links.

Content Management Capabilities

About the time that Web 2.0 was taking root, organizations began turning to *content management systems* (CMS) as a way to manage increasingly chaotic websites. While the nature of these systems varied widely, a typical CMS offered a collection of capabilities that included:

- The ability to define and customize templates that provided an entire website with a consistent look-and-feel that would be relatively insensitive to what browser a user chose.
- A user-interface that was largely form-based, meaning that content could be added or edited on the site without in-depth knowledge of HTML.
- The ability to deliver the same content in a form appropriate to different browsers. For example, a mobile user would see a site presented in manner better suited to the smaller screen of a smart-phone.
- Built-in consistency checkers that detected bad links or content that could not be viewed.
- Ability to incorporate some data-driven content, such as calendars or lists of news items. Often, additional dynamic content could be supported through the use of plugins—small applications, often developed by 3rd parties, that could communicate with the CMS using an application program interface (API).
- Hooks that allowed it to link to other applications, such as the Google API.
- Ability to define and access libraries of common content (e.g., graphics, videos, styles) that avoided duplication and obsolescence.
- Management tools to control access to different pages and sections of pages.
- Management tools that allowed user roles to be defined and workflow processes for changing site content to be established.

Naturally, capabilities such as these came with a cost. While CMS applications themselves were not necessarily expensive (many were built upon free open source applications), the use of a CMS greatly reduced the ability of individuals responsible for particular pages to implement a unique look and feel. Indeed, ensuring consistency across a whole site was the core purpose of the CMS. Developers of pre-CMS sites, however, were often less than pleased when their work was discarded in favor of “the standard”.

CMS Suites and Vendors

CMS applications came in many forms, and with a wide range of price tags. At the simple end—targeting individuals and small businesses—there were customizable applications such as *WordPress*. Originally developed as an open source platform for blogging, by Version 3.0 the product had evolved to incorporate nearly every feature necessary to be considered a true CMS: the ability to define templates, simple user interface, a huge library of 3rd party plugins, widgets that allowed it to reach out to other applications and to utilize different types dynamic content, media libraries, and so forth. Where such tools tended to be weakest was in their ability to manage complex user roles, business processes, and complex security

structures. Applications with these capabilities tended to be aimed at the enterprise. Although the origins of some of these products may have originally been open source, they tended to be sold by vendors that had heavily customized them and frequently specialized in particular industries.

Choosing between alternative CMS suites could be very challenging because of the dizzying array of different features and technologies incorporated in each. For example:

- Some CMS systems were open source, some were built on open source, and some were entirely proprietary in their construction. The degree to which this mattered was likely to depend upon the technical skills (and aspirations) of the adopting organization.
- Underlying technologies upon which the system was built varied widely. Common approaches included .NET, J2EE, and PHP. Each of these varied in the degree to which it could handle large traffic volumes and in server requirements (e.g., .NET required Microsoft Windows-based servers). In addition, a typically CMS stored its content in a database; some could easily link to nearly any SQL-based database whereas others required a particular flavor, such as Oracle or MS-SQL.
- Interface capabilities ranged from the easy to operate but limited in flexibility to the highly customizable but difficult to operate.
- Security and backup capabilities that ranged from the manually-invoked to the automated and enterprise-ready.
- Workflow management capabilities ranged from the limited to the advanced.
- The size and experience of vendors was different. Some vendors were large; others were small but had extensive experience in particular areas (e.g., government, higher education). Some had strong development experience while others were more focused on supporting standard installations.

Interestingly, the price of such systems was generally not as high as might be expected for enterprise scale applications. Typically, such systems cost well under \$100,000. What that meant was that the nature of the system and the vendor tended to matter more to the large enterprise purchaser than the price-tag. A summary of some of the better-known vendors, prepared from vendor websites, is provided in Exhibit 1.

University of South Florida

Over its short lifetime, the *University of South Florida*, located in Tampa, has experienced a rise in stature matched by few other institutions in the world. Founded in 1956, the university has transformed itself into a system soon to consist of four separately accredited institutions (the Tampa/Main campus, USF St. Petersburg, USF Sarasota/Manatee, and USF Polytechnic in Lakeland) that cover most of the west-central region of the state. It is the 8th largest public university in the U.S. and is among 108 institutions classified by Carnegie as research university/very high (RU/VH).

As shown in Exhibit 2, in 2010-2011 the USF System had a budget with revenue totaling just over \$1.5 billion. Direct state support (General Revenue) was under 19% of the total, a percentage that had been cut almost in half over the previous decade. Tuition and Fees accounted for 12% of the total. Student Financial Aid, used to pay tuition and other expenses, such as room & board, represented another important source. This category included state and federal sources, such as Florida's *Bright Futures* program, which provided full or partial scholarships for academically strong students from Florida High schools, Pell Grants, and Stafford Loans. Contracts & Grants, mainly a consequence of funded research, accounted for

as much revenue as state support and tuition combined. In fact, the extraordinary expansion of USF's in this category was a major point of pride for the institution.

Spurred by the university's past successes, for the past several years the president and provost of USF had set their sights on a new goal: achieving membership in the Association of American Universities (AAU), an organization founded in 1900 that limited itself to the top-tier of public and private universities in North America. Unlike accreditation, AAU membership did not offer any tangible benefits from a funding or eligibility for loans/grants standpoint. What it would accomplish, however, would be to cement USF's position among the top three institutions in the state (the much older *University of Florida* and *Florida State University* were the remaining two) and serve as an attraction in recruiting desirable students and faculty members.

According to the USF provost, the school had already achieved many of the benchmarks normally met by AAU members—particularly those involving funded research. Three areas where USF was falling particularly short were:

- *Student success*, as measured by 6-year graduation rates. As was often the case for universities that had experienced rapid growth, many students were first-in-family-to-attend-college and many worked as well. As a result, these ratios were substantially lower for USF than for most AAU schools.
- *Faculty to student ratios*, where its ratio of 1:27 was nearly twice that of many AAU members. USF had long been hampered by a shortage of funds with which to hire sufficient faculty members to bring it to parity with the better funded AAU members. Here, part of the problem was the unusually low tuition rates set by the State of Florida. As a result of budget pressure and a reorganization of the state university system, it appeared that some flexibility to raise tuition in the future might become available.
- *Graduate to undergraduate ratios*, with USF having far more undergraduates in proportion to graduate students than would be typical for an AAU university. Currently, the university stood at about 23% graduate enrollment overall; ideally, it would have preferred a ratio closer to 1/3.

USF was actively trying to remedy these deficiencies in a variety of ways. One key element of this strategy was using its information technology more effectively.

Information Technology at USF

The Information Technology area at USF was headed by Michael Pearce, who was a Vice President and Chief Information Officer (CIO) for the USF system, which included branch campuses. The IT activity was funded through a variety of mechanisms that included:

- A base expense budget of roughly \$12 million (academic year 2010-2011) that was funded by the university, heavily weighted towards personnel costs supporting roughly 100 full-time-equivalent (FTE) employees (FTE computes employees based on fraction of work week assigned for part time employees).
- Auxiliary activity expenses of roughly \$33 million that were largely matched by revenues from activities such as the *USF Computer Store*, which accounted for about half of the total, and various fees, such as those build into residence halls for IT support. Roughly 100 FTE employees were associated with these activities.
- Instructional enhancement activities costing roughly \$11 million. These were funded by a mandatory technology fee paid by students each semester. Because this fee specifically prohibited salary from being funded—with the exception of one FTE administrator—there were no personnel associated with this activity.

Much of the core activity of USF IT revolved around four major systems, each of which supported a different need. These systems were:

1. *Blackboard*: A learning management system (LMS) that was used to deliver course content to students and to facilitate interactive activities between students. Login identity to Blackboard was also passed to various other systems—such as the extensive collection of databases supported by the USF Library System—and the list of such systems was growing as part of a *Single Sign On* initiative that was supported, in part, by the technology fee. USF’s Blackboard system was recognized as one of the largest and most complex installations of its kind in the world.
2. *OASIS/Banner*: These systems maintained student registration information and grades. Access to students and faculty was provided through controlled interfaces that were accessed through the USF website or through Blackboard.
3. *PeopleSoft*: An ERP-based system that was principally used for human resource purposes, such as payroll and activity reporting that was critical in managing personnel who often wore many hats (e.g., instructor, researcher and administrator) and were funded from a variety of outside sources, such as grants.
4. *SharePoint*: A portal-based application developed by Microsoft that facilitated the sharing of documents and the management of workflows. This was used principally by USF employees for internal purposes and could only be accessed through a login ID and password.

A number of smaller systems were also under the control of USF-IT, such as the system supporting faculty email. Where such systems were not mission-critical and did not require special security, outsourcing was often considered. For example, USF’s student email—originally supported by a disorganized network of local servers—had been entirely outsourced to *Google*. USF-IT was also dramatically reducing the number of computer labs being supported. In their place, it had installed many of most commonly needed products—such as Microsoft Office, the suite of Adobe products, and various statistical and mathematical packages—on virtual application servers that allowed students to access the software remotely over the internet. Not only did this dramatically reduce the amount of support and maintenance required, it also meant reducing hardware dependence. Using remote terminal services, for example, meant that a Windows product could be accessed from a Macintosh, a Linux platform or even—in some cases—from a smartphone.

The development priorities of USF-IT were heavily influenced by USF’s strategic plan. The overall plan was, in turn, translated into a set of IT strategic goals, shown in Exhibit 3. One of these goals in particular—*Goal 5: Communication Strategy*—was motivating the development of two new systems. The first of these, a *customer relationship management (CRM)* system, was focused on maintaining more effective communication with students and alumni from the time they enrolled through their entire alumni period. The second, a *content management system (CMS)*, was focused on rationalizing the existing hodgepodge of disconnected web pages that currently existed on USF’s public website.

The CMS Initiative

When Kathleen Long joined USF in 2011 as Director, Web Services Marketing and Communications, the institution’s web presence was a veritable cornucopia of mismatched elements. Among the issues that were quickly identified:

- There were at least 350 subdomain URLs (e.g., domains that ended in .usf.edu). Of these, at least 150 appeared to serve no useful purpose or had been abandoned.
- There was a lack of consistency across local websites, with a particular problem being failure to adhere to USF’s branding and identification standards.

- Numerous examples of broken links could be identified.
- Sites that would only run properly under a particular browser—or would not run properly under any current browser—were common.

From a branding perspective, the situation was unfortunate. Even more important, however, was the fact that many students—particular prospective students—found themselves quickly getting lost in the site. Users became particularly frustrated when they were searching for something specific. The problem was not that the information was not on the site. The problem was that the huge volume of irrelevant and outdated information that obscured what was needed. Too often, visitors just abandoned their search. This might, in turn, mean that a promising applicant simply dropped USF from further consideration.

Kathleen Long

Kathleen Long was quick to point out that the situation at USF was far from unique. She had over 20 years of combined marketing and technology experience in higher education and in the corporate realm. Her professional background included strategy development, project management, writing, graphic design, and information architecture. She had earned a Bachelor's degree in English from Towson University, Baltimore, MD and a Master of Arts degree in Media Design from the University of Baltimore. Prior to joining USF, she had most recently served as the Media and Web Services Manager for Anne Arundel Community College.

She joined USF in April 2011 with the title Director of Web Services. Her responsibilities included oversight of the technologies that supported the university's website, portal, mobile application, and e-commerce platforms. In that role, she worked closely with university communications and marketing, spending considerable time engaging users in the requirements gathering process. She was also responsible for the IT department's marketing and communications efforts.

Obstacles to CMS Adoption

Long's experience with higher education websites also meant that she was very aware that such institutions faced certain challenges that were far greater than would typically be encountered in a corporate environment. These were largely a function of the makeup of universities, which were dominated by faculty and students.

At universities, students were frequently employed as part-time labor. Financial aid work-study, the limited opportunities available to international students, and the funding budgets of grants all contributed to this. Where a student so employed had strong technical skills, he or she was often enlisted to create or maintain websites. While often technically adept, such students tended to lack experience in building large-scale systems, including the importance of adhering to development, documentation, and testing standards. The resulting sites were often high in creativity but were correspondingly fragile, inconsistent with other university sites, and difficult to maintain. This problem was made worse by the fact that students often stayed around for only a year or two. Far too often, a proper turnover of the site was never made to the individual subsequently charged with maintaining and updating it. Sometimes, even critical information—such as system passwords—was lost in this process.

Faculty members and administrators posed a different type of problem. While far less transient than students, they were used to a very high level of autonomy. High profile research faculty members, in particular, were used to writing their own ticket and did not always respond well to guidance from others. When these individuals needed a website, they simply went out and created it, or arranged to have it created. If they liked the result, they could then become very attached to it. Any criticism of the site or attempts to bring it under IT control would be met with strong resistance. Because a research institution's prestige

tended to rest heavily upon its ability to attract and retain the best possible faculty, any attempt to overcome such resistance by force was guaranteed to be a losing battle. As Long noted, wryly:

Many of the behaviors I see every day in higher education would get you in trouble in a corporate environment. The organizational units and individuals working in higher-ed tend to pride themselves on their individuality. Adherence to mandated standards is generally not high on their priority list.

A particularly daunting challenge presented by higher education was a culture built around consensus. It was rare that stakeholders, particularly faculty members, would simply take a vote and move on. Instead, where disagreements were found to exist, the preference was to further discuss it. And discuss it... And discuss it some more... And so forth. The end result was that all but the simplest decisions tended to be made at a glacial pace. This decision-making style, while beneficial in some contexts, was a poor fit with the fast-changing pace of IT.

Evolution of the CMS Initiative

Upon joining USF, Long's principal assignment had been to determine how USF's public presence on the web could best be enhanced. She began by assessing the existing status of the university web site. What she found was succinctly summarized in an entry to the IT blog that was set up for the CMS project and maintained by Alisha Ales:

The current USF website is made up of hundreds of disparate sites that seem to have little relation to one another. Many of these sites are filled with out-of-date information, broken links and inappropriate content. Unfortunately, even relevant content is often so poorly organized that people who visit the website find it difficult to get the information they need.

The blog further summarized the web's role with respect to one of USF's most important constituents, and its existing shortcomings:

The first step is recognizing that the USF website is a marketing tool—in fact, it's USF's number one marketing tool. We all need to remember that the purpose of the USF website is market the university to our primary audience: prospective students.

We know that prospective students are looking for very specific information when they visit university websites. Primarily, they are looking for information about academic programs, admissions requirements, and cost to attend / financial aid.

Prospective students are also interested in knowing what kinds of jobs they will be qualified for after they complete a specific degree program, as well as information about campus life, housing, dining, parking, internship opportunities, student employment and other resources.

Finally, prospective students enjoy viewing videos and photos of the campus, learning about student activities and events, and reading news stories and profiles about faculty, current students and alumni who have found success at USF...

One of the most significant problems with the current USF website is that it is filled with irrelevant content. When you feature multiple navigations, long lists of "related links" and broad paragraphs of text, you make it difficult for your audience to find the information they need to take action.

Redundancy is another significant problem on the current USF website. When you post the same set of links in four different places on one web page, or post the same content over and over again on multiple pages, you confuse your audience.

Likewise, when you recreate content that already lives somewhere else on the USF website—say, parking information—you not only clutter up your web page, but you also run the risk of failing to note when the original content changes, leaving you with a web page filled with out-of-date, inaccurate information.

Identifying the problem was the easy part, however. Indeed, since nearly every major U.S. university had a variation of the same problem, it was to be anticipated. The challenge was developing a strategy to address it.

Complicating the issue was the fact that the USF web site cut across virtually every constituency within the university community. The university administration was heavily impacted, though its concerns for admissions and student success. Faculty members were impacted because what they presented on the university site could be an important influence on potential students, particularly at the research-intensive graduate level that USF was determined to expand. Facilities and concessions could play an important role in distinguishing USF from other competitors, both within the Florida state university system and, increasingly, outside of it. And, quite naturally, how enrolled students presented themselves on the USF web was likely to be great interest to high school students who had a large number of possible universities to choose from.

CMS Project Timeline

To deal with the complex stakeholder relationships, a number of activities had to be initiated. The first, taking place in early September 2011, involved establishing two committees: the *CMS Workgroup* and the *CMS Advisory Group*. The workgroup consisted principally of IT staff, with representatives from USF Communications & Marketing. This group was to be primarily responsible for implementing the actual system. The CMS Advisory Group had the role of strategic oversight and met monthly. It consisted of representatives from various colleges within USF, the library, and the administration. It was co-chaired by Long and Stephanie Harff, from University Communications & Marketing. The group had originally been called the *Web Rebranding Advisory Committee*, but its name was quickly changed to reflect the fact that there was almost no debate regarding the need for a CMS.

The second key decision made at the outset of the project was to exclude student-initiated and maintained websites from the original CMS mandate. While these sites certainly presented a face of USF to the public, attempting to exert a significant level of control over them was likely to run into many obstacles, some of which could easily be insurmountable. Turnover of the management of the sites was necessarily high, since a student was likely to graduate within a year or two of taking over maintenance of a site. That meant that keeping site administrators trained was likely to be nightmare. The university had to be highly sensitive to the desires of students, who were required to pay a mandatory technology fee and felt that they should have considerable input into how that money was spent—which made them resistant to attempts to dictate their own activities. Finally, prospective students could very well react positively to the high level of creativity exhibited in student sites. It was not clear that the informing benefits accompanying a concise well-designed USF web presence would necessarily justify the costs—both actual and emotional—of forcing student sites into a rigid set of templates.

In early October 2011, just over a month after the CMS project was initiated, a pair of requirements drafts had been developed for the system. The first set of requirements was general, focusing on system capabilities from the standpoint of users and managers, presented in Exhibit 4. The second, presented in Exhibit 5, was more technical, emphasizing issues such as database, backup, and security.

The next major activity involved ranking the various requirements. It was unlikely that any vendor would offer a package that met every requirement precisely. Thus, some means of prioritizing capabilities was necessary. Input was gathered from both groups. By 1 December 2011, both requirements and the ranking

criteria had been finalized. By 15 December 2011 28 possible vendors had been screened and, using the ranking criteria, 9 initial finalists were identified (the list shown in Exhibit 1). At that time, the advisory group also added representatives from admissions, research and design, and additional webmasters from various colleges.

Long indicated that the term “finalist” needed to be carefully interpreted. Because of a variety of regulations governing university purchasing processes, not being named a finalist would not prevent a vendor from offering a project proposal. The main consequence of being a finalist was being given an invitation to demonstrate the vendor’s particular solution. 7 of the 9 finalists responded to this invitation, presenting in January 2011 and early February 2011. Graded on a scale of 1 to 10 with respect to overall fit, one vendor scored around 9, while three scored 7.5 or higher. What this meant, from a practical standpoint, was that USF had identified at least some solutions that appeared to meet its needs.

The next step in the process was to produce an official invitation to negotiate (ITN). This public document would allow virtually any vendor to submit a proposal. Of the 34 page document, roughly 30 pages outlined USF purchasing procedures, policies, and directions for submission. Two and a half pages alluded to the specific document that vendors needed to prepare that related to the system. These are included as Exhibit 6. The ITN was scheduled to open on 2 April 2012 and vendors could then submit for at least 120 days (up to July 31st). Until that date, USF could not legally make a final vendor selection.

Current Situation

Until a vendor had been selected, actual installation of the CMS could not begin. That did not mean, however, that Long could rest easy. In many respects, the selection of a software package was likely to be the easiest part of the CMS process. When things would start to get *really* difficult was in the course of the process of trying to convince the owners of hundreds of locally-built web USF installations that they needed to migrate to the new CMS. Part of this effort involved communicating with various constituencies. This activity was ongoing and had been since the project was initiated, as illustrated in Exhibit 7. But faculty members and other high-ranking members of the university community frequently did not pay close attention to a change until it actually demanded effort on their part. It was only then that objections might be registered.

Long was quick to point out that most of the community had been very receptive to the changes that would be needed. The university was well aware of the many ways in which a strong web presence could benefit it in the long run. That was not in doubt. Nevertheless, the process of moving towards that end needed to be considered carefully.

Looking forward, there were a variety of strategies that could be employed in implementing the CMS. In general terms, some of the possibilities were as follows:

- *Directed*: Ultimately, the use of the CMS was going to be mandatory, no matter what resistance was encountered. The project had the strong support of the CIO and the Provost, who was in charge of the academic units of the main campus. Drawing upon their authority, there was little doubt that individual units within the university would, however grudgingly, move to the new system. The potential danger this presented was that efforts would be half-hearted and that considerable foot-dragging would occur. No matter how good the templates or organization of the university’s site, it could not compensate for limited or incomplete information supplied by users. On the positive side, this approach would likely be the quickest to produce a consistent and appealing USF site.
- *Phases*: Implementation could be conducted in a series of phases, according to a timetable. Under this plan, specific USF units would move to CMS while others waited their turn. A certain amount of phasing was inevitable, as some units would be needed to pilot-test the system before a full scale roll-

out took place. The advantage of a planned phase it was that it would reduce the demands on the IT and communications groups during the implementation and would allow issues detected in one phase to be addressed (and, hopefully, resolved) prior to starting the next phase. On the negative side, it could leave the USF web site as a bundle of inconsistencies over the course of a relatively long implementation cycle. It could also lead to considerable jockeying with respect to who would go first and who would go last. Moreover, it was not altogether clear whether it would make sense to start with the hardest or the easiest units.

- *Modular*: Another possible broad strategy would be to break down the CMS into component functional elements and roll these out one at a time across the entire university. For example, each academic department had its own website. A single or small number of templates could be designed for this purpose and then all departments could be brought on at once. The advantage of this approach would be that rapid consistency across the university would be achieved each time a new element was rolled out. Like the directed approach, however, it could maximize the amount of widespread resistance encountered, particularly where academic functions were involved.
- *Islands of innovation*: A different approach sometimes proposed for information systems was more client-driven. In this approach, a variation of the phased-in approach, specific units would volunteer to undergo the full transition to CMS. Other units, then able to see the full benefits of the project, would want to jump on board, leading to a user-driven push to get the system completed once it reached a certain critical mass. While great-sounding in theory, the drawback with this approach was that it could not be counted on to work. On the other hand, it did not involve trying to force tenured faculty and administrators to adopt a system against their will, which was unlikely to work either.

There were also a variety of other issues that needed to be considered as part of CMS implementation planning. For example, the plan was for USF to adopt consistent URLs, e.g., **usf.edu/mydomain**, as opposed to having numerous sub-domains, e.g., **mydomain.usf.edu**. During the transition, the question was whether to run the new and old sites in parallel. On the one hand, running in parallel would ensure that information was available during the transition. On the other hand, it would temporarily add to already major problem of duplicated and inconsistent information on the web.

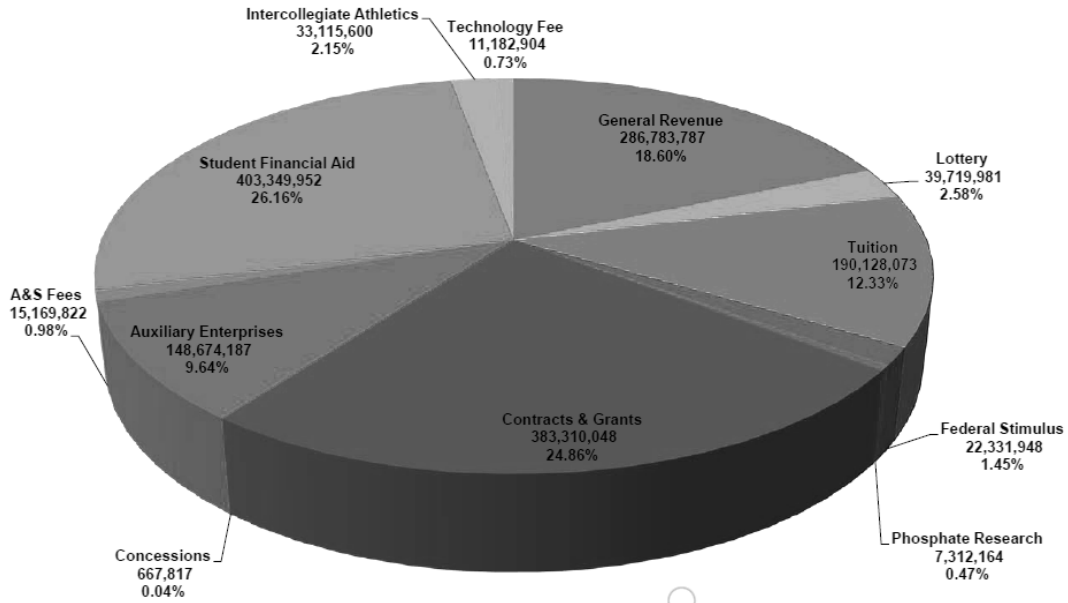
Exhibit 1: Summaries of Major Vendors

Vendor	Core Technologies	Comments
dotCMS	Build on J2EE open source core.	Has higher education specialized version. Clients typically perform a lot of the customization required.
Terminal4	Built on J2EE core. Capable of managing scripting in a variety of formats, including JSP, PHP and ASP.NET.	Has some higher education clients, including high profile sites such as <i>Oxford University</i> . CEO of company participated in the demonstration.
Alfresco	Java content platform. Incorporates Drupal open source platform.	Does not appear to have version specialized for higher education (does have a government version). Much of their support appears to come in the form of consulting engagements.
Acquia	Provides releases and extensions to Drupal open source platform.	Provides pre-configured version for higher education. Did not visit USF to provide a demonstration.
Jadu	Releases parallel versions in PHP and Microsoft .NET.	Has higher education specialized version. U.K.-based company.
Orchard Project	Built on ASP.NET. Entirely open source project.	Software is free and must be downloaded and customized from Codeplex repository. Did not visit USF to provide a demonstration.
Ektron	Built on Microsoft ASP.Net platform.	Featured a large number of higher education customers, including local <i>University of Tampa</i> . Had additional optional modules for linking to social networks, marketing and eCommerce.
Omniupdate	Built on Java-based CMS core that relies heavily on XML templates to manage other web assets. Also offers hosted version.	Specializes in higher education, with over 550 institutions using the product. CMS emphasizes managing existing web resources through templates rather than fully replacing web pages, allowing for more gradual implementation.
Sitefinity	Built on Microsoft ASP.NET platform.	Had some higher education customers. Third party vendors were typically required to assist in development and customization.

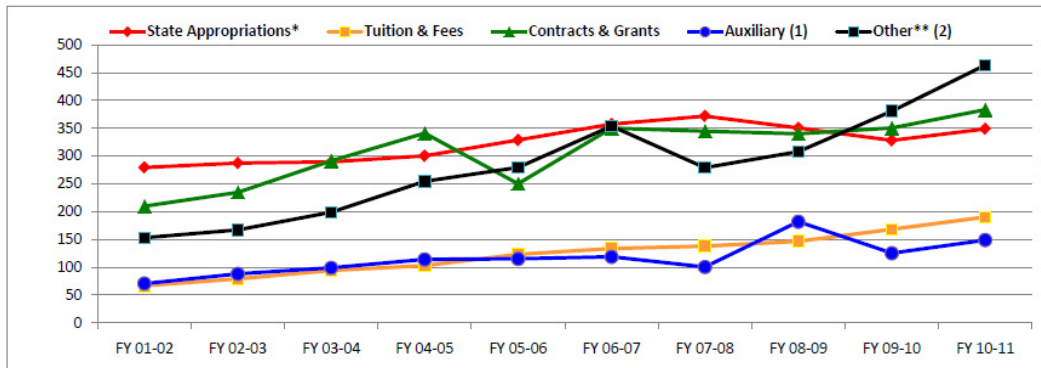
Source: Compiled from vendor company websites and demonstration evaluations by the case writers.

Exhibit 2: USF Budget Revenue Sources and Trends

University of South Florida System
2010-11 Operating Budget by Funding Source (Excludes Direct Support Organizations and Capital)



University of South Florida System
2010-11 Operating Budget
10 Year History of Budgeted Expenditures by Funding Source
(Excludes DSOs, Capital and Phosphate Research)
(In Millions)



Funding Source	2001-02	2002-03	2003-04	2004-05	2005-06	2006-07	2007-08	2008-09	2009-10	2010-11
State Appropriations*	\$ 279.5	\$ 287.3	\$ 289.4	\$ 300.3	\$ 328.6	\$ 357.2	\$ 371.9	\$ 350.4	\$ 328.0	\$ 348.8
Tuition & Fees	66.7	79.1	94.1	102.9	122.9	133.5	137.7	146.5	168.0	190.1
Contracts & Grants	209.5	234.7	291.4	340.9	250.0	350.0	344.8	340.0	350.3	383.3
Auxiliary ⁽¹⁾	70.4	87.9	98.8	113.8	114.9	118.6	100.2	181.7	125.0	148.7
Other** ⁽²⁾	152.8	167.2	198.8	254.4	279.5	353.6	278.9	307.7	380.7	463.5
Total	\$ 778.8	\$ 856.1	\$ 972.5	\$ 1,112.3	\$ 1,095.9	\$ 1,313.0	\$ 1,233.5	\$ 1,326.4	\$ 1,351.9	\$ 1,534.4

*State Appropriations exclude Phosphate Research Trust Fund.

**Other includes Activity & Service Fees, Intercollegiate Athletics, Concessions, Financial Aid and Technology Fee.

⁽¹⁾ Increase in Auxiliary Enterprises in fiscal year 2008-09 is a result of additional budget authority for cash reserves.

⁽²⁾ Increase in Other in fiscal year 2006-07 is due to an increase in the Stafford Loan Program.

Exhibit 3: USF Strategic Goals

Strategic Goals

The following are the strategic goals that IT will focus on from 2010 to 2015 to support the USF System and its member institutions in achieving their missions and strategic goals.

Goal 1: Student Success

Enhance student success by providing “anywhere, anytime” access to 21st century learning tools and resources our students and faculty require to succeed.

Goal 2: Customer Service

Establish service management and communication processes that are responsive to our customers and drive increased satisfaction.

Goal 3: Research

Provide the research community with the technologies, resources, and administrative solutions for recognition as a major 21st century research university.

Goal 4: Administrative Effectiveness

Improve administrative efficiency and decision making by providing integrated administrative systems and information that enable streamlined processes and effective management.

Goal 5: Communication Strategy

Develop and deliver a consistent, branded, system-wide information technology communication plan that promotes awareness and use of IT services and encompasses the informational needs of various stakeholder groups.

Goal 6: IT Infrastructure

Establish a sustainable, secure, compliant, robust and reliable IT infrastructure that enables a consistent technology experience for all users.

Goal 7: Governance and Funding

Refine the governance structure and establish a funding strategy to effectively align and maximize IT resources to achieve the priorities of the USF System and its member institutions.

Goal 8: Employee Recruiting, Retention & Development

Become a nationally recognized IT workplace through the attraction, development and retention of world-class personnel.

Source: USF IT Website

Exhibit 4: Excerpts from General Requirements Draft

Template Options and Requirements

Options

Basic templates:

- 1 or 2 variations containing consistent headers and footers, color schemes, fonts (www.usf.edu/brand) for home pages, and standard secondary and third-tier pages with adaptive layout/formatting for all pages

News template:

- Includes basic template requirement, but is distinct with more autonomy such as:
 - Ability to attach a thumbnail as well as a larger image for feature stories
 - E-mail newsletter creator (ie: *InsideUSF*)
 - Links section for related or recommended articles

Specific services content template:

- Multi-paged public-facing content (ie: General Counsel, Registrar's Office, etc.)

Forms template:

- To allow the user (CC or CM) to create a form for user submission

Bio Template:

- A template that would allow the CC or CM to create a page containing biographical information for a faculty or staff member

Outside applications:

- (ie: eScheduler) to be branded per CMS templates

Requirements

All templates options to include:

1. WYSIWYG text editor (wireframe [idea](#))

- Removes software product formatting
- Formatted font colors and types
- Formatted bullets
 - Ordered lists
 - Heirarchies
 - Outlines
- Basic Microsoft Word capabilities to include spell check, grammar check, word count, cut, copy, paste, etc.
- Tables
- CSS/live page preview
- Capability to insert links

2. Navigation and menu controls

- Navigation options via pluggable widget that can be placed in a designated area on a template if desired (vertical, horizontal, flat, nested, animated)
- Page audit/orphaned page or link search (*automated sitemap to highlight the existence of an orphaned page/link*)
 - Alert audit (*CC can have the option to delete a page or take action based on the notification of a link discrepancy*)
- Search and taxonomy – (tagging, organization)
 - Separate searches

1. Search the entire site
 2. Individual department search (*maybe a widget or dropdown box*)
 3. One course/class search
- Google vs. built-in CMS search capabilities
 - Meta-data tagging features
 - Automated site search
 - ADA Compliance
 - Text-only functionality built-in
 - Share, e-mail and print buttons on each page
 - Breadcrumb navigation on each page
 - Access to a “USF Dictionary” to allow for the creation of words hyperlinked within content throughout the CMS
 - The words and definitions would be controlled by content contributors/content managers per department
 - When a word from the dictionary is found within content it would be turned into a hyperlink. If a user hovers over the word, a short definition would pop up to include a link to more content that relates to the word.
 - Examples: FERPA, Non-degree application, Lee Roy Selmon

Content Contributor Administration Interface Requirements and Capabilities

Administration for Content Contributor:

1. “Create New Page” Requirements:

- Choose directory
- Choose template design
- Access to image library for common images (*see more below*)
- Access to link library for common content
- Access to widget library
- Capability to add a standardized set of social media icons
- Capability to create surveys
- Capability to include department facts /important USF Facts (*ex: links to USF Strategic plan, mission, USF fact book, etc.*)
- Capability to include Google maps (Campus and Parking)
- Capability to include auto release and posting dates for content
- Option to utilize or include an image carousel - *add captions along with alt text*
- Option to include RSS feeds
- Capability to upload files/documents/standard forms:
 - PDFs
 - Microsoft Office docs (*not Access*)
 - Considerations:
 - Other file types?
 - Size limit?
 - Non-standard software file types would need special process
 - If a non-standard file becomes more prevalent, then allow upload
- Capability to upload images and video
 - Considerations:
 - Store images and video on a separate server (*image and video library?*)
 - Automated video and image formatting – *specs TBD*
 - Image and video management

- Image approval – use of images in dB only or CC/CM approvals? *process TBD*
 - Marketing library and thumbnails – *specs TBD*
 - How to handle news images - TBD (*CM would have higher level of access*)
 - [Wireframe document w/possible solutions](#)
 - Capability to preview pages by submission or display preview page before go-live
 - Generate preview e-mail or private link that’s not available on the site (soft release) to allow an approver or another entity to red-line on the screen
 - [CC/CM Interface possibilities](#)
- 2. “Edit Page” Requirements:
 - Create a new page +
 - Restore a version; store up to 3
 - Indicate difference between old and new versions
- 3. “Delete Page” Requirements:
 - Capability to recover and utilize back-up versions
- 4. General Requirements for both CC and CM:
 - Content update/change indication to allow CC and CM to track changes (*i.e.: why is content denied?*)
 - Content reminders/indications (expired event, outdated content, etc.)
 - Archiving capabilities – (include start and end dates, along with e-mail notifiers for CC and CM)
 - In-context editing to allow the CC make content updates directly on the web page
 - Capability to move a page to another template/change template
 - Capability to move a page into another directory
 - Considerations:
 - Link updates to other pages
 - Produce a nightly report that checks for broken links
 - Identify updates to entire system
 - Auto indicator – could still pose as an issue because the users wouldn’t know why, unless they have some screen or message that indicates why...
 - What’s the trail of links?
 - Error handling?
 - Capability to preview page to see what it will look like on a mobile device

Content Manager Administration Interface Requirements and Capabilities

Administration for Content Manager:

- To include Content Contributor Administration Interface Requirements and Capabilities + :
 - Capability to access user set up and permissions
 - Capability to set up workflow
 - Content approval
 - Image library upload
 - Access to widget library
 - Access to metrics and statistics including meta data

UCM Administration

- Capability to create new directories for new area/department/college
- Supply metadata for SEO purposes
 - Title, keywords, description of content
 - Tag Clouds

CMS Training Requirements

- If vendor solution, the package should include training resources
- If not a vendor solution, a non-vendor or combined vendor/development solution, training resources should also be considered as components that need to be built (should be fun and engaging)
- A resource tool will be provided for Content Contributors and Content Managers to include regularly-held Best Practices Sessions:
 - How to write for the web
 - How to edit/organize content
 - How to re-size images, etc.
 - Search Engine Optimization
 - Shared content and features training
 - Help the user develop clean site layout w/navigation tips
 - Understanding analytics
 - Using hyperlinks
 - How to use the widget library

Content Migration Considerations

- What's the process going to be for existing CMSs to the new one?
- What's the process going to be for non-existent CMSs?
- Ability to migrate from current CMS to another CMS

Source: ISF IT Portal

Exhibit 5: Excerpts from USF CMS Technical Requirements Draft

Technical Functionality/Requirements

1. *Cross-platform/browser support*
 - a. Full functionality over MAC/PC
 - b. Support for common browser solutions (IE, FireFox, Safari, Chrome, etc.)
 - c. Mobile browser functionality/templates (auto-generated or manual)
 - d. Text only functionality/templates (auto-generated or manual)
 2. *Management Interface*
 - a. All functionality for Admins/CMs/CCs is done via a web-based GUI vs. installed software.
 - b. Management restriction based on browser type (PC vs. mobile).
 - c. Friendly UI for management/content publishing.
 3. *Image Management/Media Management*
 - a. Media search via metadata
 - b. Present Media categories/groupings to specific security roles (i.e. UCM might not want someone reusing media reserved for only posting on news articles)
 - c. Image replacement (individual images and/or multiple changes site/sub-site wide).
 - d. Image optimization/resizing.
 4. *Site Structure & Content*
 - a. URL Aliasing/Rewriting
 - DB will automatically clean up/rewrite links link so that it's www.usf.edu/it is displayed vs. www.usf.edu/it/article_56798*&*&^
 - b. Ability to manage site structure (i.e. move site structure with minimal impact such as www.usf.edu/computerstore/tabstyle to www.usf.edu/computerstore/) This would involve:
 - Structured to accommodate scenarios such as www.usf.edu/ucm/brand; www.usf.edu/cota/dance.
 - Automatic link updates.
 - Re-crawling of content.
 - Redirection from old links for any bookmarks.
 - 'Reserve' old address so that a new page isn't overwriting a potential placeholder.
 - Set a removal date for the redirection and address reservation.
 - c. Content-to-Template Preview
 - Preview existing content with new templates.
 5. *Open APIs*
 - a. CMs/Technical Staff should have the ability to create interfaces into/out of the CMS platform based on business needs.
 - Web service APIs (SOAP and/or RESTful) for:
 - User management (Add/Remove/Update).
 - Access Control (Add/Remove/Update Permissions).
 - Content Management (Add/Remove/Modify).
-

- Site
 - Directory
 - Page
 - Page Content
 - Assets (images, video, etc.)
- b. Javascript Library
- What is available with the CMS?
 - Can we select/opt to utilize an alternate option.
- c. Ability to create custom data libraries and/or reference outside libraries within the form functionality of the CMS.
- Connectivity to different DB platforms (Oracle, MySQL, MS SQL, etc.)
 - Data flow control (push/pull).
 - Ability to control who can create/view/update these libraries.
- d. Ability to custom code widgets.
- Widgets must be written in a general-purpose language (ASP, PHP, etc), not a CMS-specific language
 - Ability to tag-in custom widgets into WYSIWYG editor as needed.
 - Ability to include widget via insert object functionality in templates/custom documents.
6. *Newsletter (Might fit into Usability Requirement document)*
- a. Management of newsletter
- Creation based on security role/permissions.
 - Custom newsletter templates.
7. *Extensibility*
- a. Social Media/3rd Part Integration – Common integration with existing services such as You Tube, live chat, Facebook/Twitter/Google+ , etc.?
- b. Geo Mapping
- Content provided by Google
 - Geo data to be used on supported mobile devices
 - Virtual tour
 - Not campus wide
 - Potential of Residence Halls, etc.

System Maintenance

1. *Outage notification/alerting (outside of CMS but worth mentioning)*
- a. Ability to set a friendly Page not Found page
- Suggest alternate pages (“Did you mean...”)
- b. Alternate broken link/media/image icon (globally to CMS and potentially locally to sub-site)
2. *Staggered Patching for System Maintenance*
- a. Ability to patch Front End/Application servers without bringing down entire CMS solution.
3. *Development/Test-Staging Platforms*
- a. Publish to Dev/Test-Staging server or environment (if possible).

- b. Creation of a Dev/Test-Staging environment for coding, patching, and upgrade certification.
- c. Load Testing mechanism to determine if intended changes will have a negative effect on production.
- 4. *Link checking (similar to Reporting & Notification entry below)*
 - a. Check what links to/from within the CMS to ensure no dead/broken links.

System Architecture

1. *DB Platform Support*
 - a. Any of the three supported DB types (MS SQL, MySQL, or Oracle) is required.
2. *Security*
 - a. SSL Certificates
 - b. Authentication mechanism
 - Public Pages - Visitors can authenticate to participate in discussions and/or receive personalized content using the following protocols:
 - OpenID
 - Facebook Connect
 - Twitter (OAuth)
 - Admin pages – CCs and CMs must login using the NetID credentials to modify content using one of the following protocols:
 - CAS
 - SAML (Shibboleth or ADFS)
 - ws-* protocols (ADFS)
 - c. Search/scrub data for common fields (SSN, CC Number, etc.)
 - d. Form creation (potential a sub-topic on Widgets)
 - Parameterized and scrubbed entries.
 - Security against SQL Injections, Cross Site Scripting (XSS), and Cross Site Request Forging (CSRF)
 - Captcha-like commenting/submissions
 - e. Granular security model
 - CMs to be defined specific permissions tailored to their usage.
 - CCs to be defined specific permissions tailored to their usage.
 - If we are creating/hosting custom content or media for UCM, we would need to define some security roles for:
 - Audio/Video/Image management (approval, submission, metadata management, etc.)
 - Custom widget upload/creation.
 - CM/CC management account or groups to grant/remove access to the CMS.
 - Page ownership/contact information (viewable by UCM/CMs/Admins as needed)
3. *Audit & System Logging*
 - a. Log changes to content.
 - b. Security mechanism matches University cyber security policy.

- Ability to send logs to a remote log server
 - c. Log server errors.
 - 4. *Backups*
 - a. Farm/Solution backup.
 - In place/One-time full backups.
 - Scheduled backups (incremental).
 - b. Transactional roll-ups/roll-backs.
 - c. Off-site backups or backup to the cloud?
 - 5. *Scalability*
 - a. Multi-site hosting – hosting of multiple sites under the same CMS implementation (i.e. usf.edu, health.usf.edu, etc.)
 - b. Flexibility to allow for close to 100% uptime based on SLAs.
 - c. USF Cloud Scalability (On Site @ USF locations)
 - Ability to host content DB (mirroring, replication, etc.) across our 3 site architecture (Tampa, Winter Haven, and Clemson)
 - Ability to host multiple Front End/Application servers across our 3 site architecture behind the NetScaler or built-in Farm management functionality.
 - “Shared-nothing” clustering of the web/application servers is preferred for ease of scaling & failover.
 - d. 3rd Party Cloud Scalability (Off Site)
 - Ability to host an instance of the CMS outside USF’s systems.
 - e. Caching
 - Caching functionality for core content either on the Front End/Application server or a web accelerator product (from CMS or NetScaler).
 - Ability to integrate with a Content Distribution Network (Cloudfront, Akamai, etc.) for large and/or commonly requested files.
 - 6. *Import/Export Functionality*
 - a. Export content in XML format.
 - b. File import/upload functionality.
 - Workflow on this (i.e. CM’s need to approve an uploaded document and auto-deletion if it’s not approved)?
 - c. Bulk-load/Bulk-import functionality to script in existing content while scrubbing any and all styling.
 - 7. *Reporting & Notification*
 - a. Built-in reporting metrics for usage, aging, orphaned docs, etc.
 - b. Ability to create customized reports?
 - Simple UI for creating such reports.
 - c. Create custom notification based on system changes, crawling, backups, workflow, etc.
 - d. Scheduled reporting.
 - e. End-user reporting of issue(s) on a page.
 - Notification to CMs and/or page owner based on metadata.
 - 8. *Date-based snapshots*
-

It must be possible to retrieve the state of the entire site on a given date. This is particularly relevant in legal cases which are based on the information presented on the site. The recovered site must be fully functional, including the hypertext links, images and other related files. It would also be useful to view a historical snapshot of the site, without having to fully recover it. To achieve this capability, the content management system must version all assets within the system, including:

- pages
- templates
- style sheets
- users
- security settings
- images
- documents

Source: ISF IT Portal

Exhibit 6: Excerpts from ITN for CMS

The vendors shall provide responses in a tabbed ring binder with the following form:

Tab A: Company viability

1. Provide company background information describing the company organization, incorporation date, and if the CMS product is designated as proprietary or open source. Include all warranty representations and product support options.
2. Provide company financial information, including annual reports, if available
3. Provide the number of full, large scale CMS product implementations at higher education institutions. For at least 5 of those implementations, list the following:
 - a. Institution name and student count.
 - b. Description of product installation scope, version and configuration.
 - c. Indicate the approximate number of pages included in the implementation.
 - d. Indicate the date and timeline of the implementation (how long did it take?) .
4. Provide the manufacturers' life cycle overall objectives. From market availability to end of life. Describe product release availability and schedules.

Tab B: Scalability, Flexibility, Reliability and Performance

In this section provide information on how adoptive the proposed configuration is to changes in technology or USF's needs over time. Provide answers to the following specific questions:

1. Does your solution allow for direct edit of content and comparative or differential views of content versions?
2. Describe how your CMS solution can be expanded up and out on both a private and public cloud. Is simultaneous web presence, both locally and outside the USF network, inherent to your solution?
3. Does this scaling include both the application and database servers?
4. Provide a diagram of how your CMS solution will accomplish this.
5. Describe how your solution promotes content changes from a development to staging to production environment.
6. Describe how your solution remediates any inconsistencies between scaled out deployments?
7. What languages can be used for widget creation?
8. Are open APIs for SOAP/Restful data calls and JavaScript libraries available for use?
9. Describe your Administrative security model and describe its granularity (role based privileges).
10. Does your solution provide multi-browser support for the administrative user? Describe the options available.
11. Does your solution provide 2nd and 3rd stage workflow? Describe available options for workflow processing. Specifically, are there options for triggering scripts and database procedures from your workflow process?
12. What authentication mechanisms are supported with your solution?
13. What types of customer reports are inherent to your solution? Does your solution require integration with an outside analytics tool? If so, please explain.
14. Describe audit and system logging capabilities that are inherent with your solution.

15. Does your solution provide the ability to migrate to/from another CMS?
Describe the process to do this and in what format the input/output files will be.
16. Describe the solution's caching mechanism.
 - a. How deep in your structure can pages be cached?
 - b. Is there separate caching configurability for various content and pages?
 - c. Does your solution support hardware, appliance caching or a combination of both? List and describe.
17. Does your system support a backup/recovery system or is this performed outside of your solution? Explain.
18. Describe the recovery options within your product if any of the following are inadvertently removed or deleted:
 - a. Content recovery
 - b. Page recovery
 - c. Site recovery
19. Does the solution support overall system health monitoring or status views?
Describe.

Tab C: Customer Usability

1. Describe the training/documentation options available for your solution and incorporate these options into implementation plan referenced in TAB F.
2. Is training and/or implementation services provided at no additional cost?
Explain.
3. Describe the capabilities of your authoring tools:
 - Allows copy/paste from other products and removes any product specific formatting
 - Allows for standard social media integration and RSS feeds
 - Allows for basic Word WYSIWYG editor capabilities and standard template features such as breadcrumb navigation, share, email and print buttons, etc.
 - The WYSIWYG editor functions can be locked down based on the administrators security level
4. How is content changed/reviewed by both the content contributor and the content manager?
5. Describe the capabilities of your image, video and document libraries (i.e. can files be uploaded and then manipulated within your solution, can this be controlled by security level, etc.)
6. Does your solution utilize a common content library and describe how a change to this common content affects all areas that are using it?
7. Does your solution provide a mechanism for reusable content? Describe the process.
8. Describe your standard widgets and the process for creating our own widgets to make available to content contributors.
9. Can multiple templates be created and controlled by security level? Describe what content contributors would see when creating a new page.
10. Describe your solution's built in workflow for content approval. Does your solution provide the ability for content auto release/removal?
11. Describe how your solution handles web and accessibility standards compliance and describe the solution's ability to create a "text only" version of a website (used for screen readers).
12. Describe the solution's search capability. Can multiple searches be created across the platform?
13. Describe how metadata/taxonomy is setup for SEO purposes.
14. Describe your solution's ability to audit for orphaned pages. Specifically, how does your solution notify others that content or a page has been relocated or removed?

Tab D: Cost

Provide in spreadsheet format a summary of costs, any and all costs associated with installation, implementation, administrator licensing, ongoing maintenance costs, recommended training costs, support costs, upgrade costs and any other costs for the solution including a cost breakdown for both (if offered) a vendor hosted and self

hosted pricing structure.

Extend the costs into the columns for years 1-3 on the spreadsheet and specify any and all costs for annual escalation and scheduled version releases.

Also, provide an average number of USF employee hours required for the implementation and initial administration of your solution and incorporate this information in the implementation plan referenced in Tab F.

Include any proposal incentives and/or value added elements with a dollar value that USF can consider in the evaluation. In the past, vendors have offered additional discounts, provided the University donations of software or hardware, free training/implementation, etc. We encourage vendors to be innovative and will consider any ethical proposals. Please outline your value add proposition in this section.

Provide information and detail any and all costs regarding the Administrative tools utilized to manage the solution, including:

- Configuration management
- Provisioning management
- Hierarchical storage management
- Performance management
- Backup/Recovery management

Tab E: Other functionality/factors

In this section provide any other functionality/factors that set your CMS solution apart from your competition.

1. List current functionality not previously described.
2. List future functionality that is being reviewed or being developed for future releases.
3. Tell us why we should purchase your solution.

Tab F: Implementation Plan

In this section provide a detailed implementation plan to initiate the offered product solution including earliest product, documentation, and manual delivery date(s), implementation milestone schedule and estimated implementation completion date. Please include a listing of all vendor requirements for initiation of each milestone.

Source: USF IT Website

Exhibit 7: Summary of Outreach Activities

Presented to:

- Provost's Retreat
- Council of Deans
- ACE Council
- A&P Council
- CTIR
- ITMC
- STAC
- Professional Communicators Group
- IT Standards Board

Met with:*Academic*

- College of Arts and Sciences
- College of Marine Science
- College of Education
- College of the Arts
- Patel Center for Global Sustainability
- Air Force ROTC
- Undergraduate Studies
- Office of Student Success
- Physical Education Department

Non-academic:

- Administrative Services
- HR
- Student Affairs
- USF System
- IT
- UCM
- Media Innovation Team
- Provost's Office
- SDS

Source: USF CMS Blog (2/2/2012)