Web Presence and Services Program Requirements
Requirements for the Web Content Management System (WCMS) Tool, Function and Design, Governance, Services Definition, and Other Unmet Needs

3/17/09
Version 4.0
Authors: Marion Bashista, Ann Berry-Kline, Leslie Geary
Document Control

Change Record

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Version</th>
<th>Change Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/3/09</td>
<td>Marion Bashista</td>
<td>1.0</td>
<td>Document created</td>
</tr>
<tr>
<td>2/11/09</td>
<td>Leslie Geary</td>
<td>1.0</td>
<td>Tool requirements added</td>
</tr>
<tr>
<td>2/11/09</td>
<td>Marion Bashista</td>
<td>1.0</td>
<td>Web Presence requirements added</td>
</tr>
<tr>
<td>2/13/09</td>
<td>Marion Bashista, Leslie Geary, Ann Berry-Kline</td>
<td>2.0</td>
<td>Updates per review by Ann</td>
</tr>
<tr>
<td>2/17/09</td>
<td>Marion Bashista, Leslie Geary, Ann Berry-Kline</td>
<td>2.5</td>
<td>Updates and new version for Web Council</td>
</tr>
<tr>
<td>2/27/09</td>
<td>Leslie Geary</td>
<td>3.0</td>
<td>WCMS functional and technical requirements updated.</td>
</tr>
</tbody>
</table>
| 3/17/09 | Leslie Geary                | 4.0     | *Updated functional and technical requirements based on document review period.*  
*Addition of Appendix A: RFP Requirements Text.  
*Insertion of page numbers |

Reviewers

<table>
<thead>
<tr>
<th>Sign Off Date</th>
<th>Reviewer</th>
<th>Position</th>
<th>Sign Off</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ann Berry-Kline</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Distribution

<table>
<thead>
<tr>
<th>Copy Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table of Contents

1 EXECUTIVE SUMMARY .............................................................................................................1

1.1 Background.............................................................................................................................1

1.1.1 Requirements Gathering Process ...................................................................................2

1.1.2 Requirement Definitions .................................................................................................2

1.2 Overarching Themes from Campus Groups ..........................................................................3

1.2.1 Easy to Use........................................................................................................................3

1.2.2 Consistency of Look and Feel Across Campus..................................................................3

1.2.3 Importance of Local Design Flexibility ..........................................................................3

1.2.4 Importance of Authoritative Content ..............................................................................3

1.2.5 Robust Basic Tool is Highest Priority .............................................................................3

1.3 Key Requirements ..................................................................................................................4

1.3.1 WCMS Tool.......................................................................................................................4

1.3.2 Web Presence (Non-Tool)................................................................................................4

1.3.3 Other Web Related Unmet Need.......................................................................................5

2 WCMS FUNCTIONAL REQUIREMENTS ..............................................................................6

2.1 Content Creation......................................................................................................................6

2.1.1 Authoring and Editing.......................................................................................................6

2.1.2 Date Controls.....................................................................................................................8

2.1.3 Metadata............................................................................................................................9

2.1.4 Custom Forms...................................................................................................................9

2.2 Content Requirements............................................................................................................10

2.2.1 Content Types and Sources............................................................................................10

2.2.2 Content Repositories.......................................................................................................11

2.3 Content Delivery...................................................................................................................12

2.3.1 Publishing Targets ...........................................................................................................12

2.3.2 Web Standards ................................................................................................................13

2.3.3 ADA Compliance .............................................................................................................13

2.3.4 Web Site Visitor Experience............................................................................................15

2.4 Site Management & Quality Control ....................................................................................16

2.4.1 Change History................................................................................................................16

2.4.2 Publishing Workflow.........................................................................................................17

2.4.3 Site Reports or Notifications............................................................................................18

2.5 Application Administration..................................................................................................19

2.5.1 User and Group Management..........................................................................................19
3 WCMS TECHNICAL REQUIREMENTS ................................................................. 25

3.1 Technology Stack ........................................................................................................... 25
  3.1.1 Application User Interface ......................................................................................... 25
  3.1.2 Supported Technologies and Hosting Requirements .................................................. 25
  3.1.3 Security ....................................................................................................................... 26
  3.1.4 Additional Security Requirements for Hosted Systems ............................................. 28

3.2 Application Development and Administration .............................................................. 28
  3.2.1 Development and Integration .................................................................................... 28
  3.2.2 Technical Application Administration and Support .................................................. 29

3.3 Performance, Business Continuity, and Disaster Recovery ........................................... 31
  3.3.1 Performance and Business Continuity ....................................................................... 31
  3.3.2 Disaster Recovery ....................................................................................................... 32

4 WEB PRESENCE (NON-TOOL) REQUIREMENTS .................................................... 33

4.1 The Web Function and Design Project .......................................................................... 33
  4.1.1 Redesigned Suite of Templates .................................................................................... 33
  4.1.2 Information architecture ............................................................................................. 33
  4.1.3 Authoritative Content Sources ................................................................................... 34
  4.1.4 Web Specific Style Guidelines .................................................................................... 34

4.2 Governance Requirements ............................................................................................. 34
  4.2.1 Standards and Policies ............................................................................................... 34
  4.2.2 Authoritative Content Sources ................................................................................... 34
  4.2.3 Cross-divisional User Groups ..................................................................................... 35

4.3 Web Services Requirements .......................................................................................... 35
  4.3.1 Training and documentation ....................................................................................... 35
  4.3.2 What is the service and how do I get it? ..................................................................... 35
  4.3.3 Costing Model ............................................................................................................ 35
  4.3.4 Defined Roles and Responsibilities ............................................................................ 35
  4.3.5 Applications Library ................................................................................................. 36
  4.3.6 Adequate Resources .................................................................................................. 36

5 OTHER WEB RELATED UNMET NEEDS ..................................................................... 37

5.1 Enterprise Content Management Tools .......................................................................... 37
  5.1.1 Collaboration Space ................................................................................................... 37
  5.1.2 Digital Asset Management ......................................................................................... 37
  5.1.3 Document Management ............................................................................................ 37

5.2 Other Functionality ....................................................................................................... 38
  5.2.1 eCommerce ................................................................................................................. 38
6.2.2  Intranet ................................................................. 38
6.2.3  Integration With Campus Applications ................. 38

6  APPENDIX A: REQUIREMENTS TEXT IN RFP ......................... 39
6.1  Critical Requirements .............................................. 39
6.2  High Requirements .................................................. 41
6.3  Medium Requirements ............................................. 45
6.4  Low Requirements ................................................... 50
1 Executive Summary

In October of 2008, Chancellor Blumenthal formally charged University Relations and Information Technology Services (ITS) to co-sponsor a Web Presence and Services Program “to address a range of web presence factors including updated content, design, service offerings, and support technologies.”

Several high priority components identified in this charge are particularly relevant in the selection of a web content management tool:

- Designing an updated, effective, and consistent look and feel for UCSC’s web presence
- Selecting and installing a centrally-supported, easy to use web content management tool
- Defining design standards, security strategies, and compliance activities for sites defined as official UCSC web sites

1.1 Background

The Web Presence and Services Program coordinates a series of web-related projects designed to improve our web presence. Areas of focus include campus processes, roles, tools, and services that support campus web pages, as well as the design and function of the central campus site.

The Web Content Management System (WCMS) project was charged with gathering requirements, selecting a web content management system tool and implementing that solution. The requirements were gathered from a series of interview sessions.

The requirements gathering process gathered current and future business needs and other web presence information from divisions and departments across campus. Some of these business needs may be satisfied with a web content management system and others may not.

Two categories of requirements are detailed in this document.

- WCMS tool requirements are those that would be satisfied with a web content management system. The selection committee will review business requirements and select candidate tools that meet as many needs as possible.

- Web Presence (Non-Tool) requirements are those outside of the application tool. They cover a broad range of campus needs outside of the tool. This set of requirements provides the Web Presence and Services Program Leadership Team and Web Council with input for use in other components of the Web Presence and Services Program.
1.1.1 Requirements Gathering Process

The Web Presence and Services Program and Web Content Management System (WCMS) Requirements Gathering Team engaged campus constituents for 2-hour requirements gathering sessions that ran from mid-November 2008 through the end of January 2009. Campus constituents were also requested to complete a simple survey regarding their web publishing needs.

Seventeen requirements gathering sessions were held with the academic, academic support, and other campus constituencies with web interests. Approximately 300 staff and faculty attended the interviews. Areas interviewed included:

- Academic Divisions – Arts, Humanities, Social Sciences, Baskin School of Engineering, Physical and Biological Sciences, University Library, University Extension, Silicon Valley Initiatives, and an Open Forum for Academic Units
- Academic Support Divisions – Business and Administrative Services, Student Affairs, Chancellor’s Office, University Relations, and Information Technology Services

The vast amount of information captured was reshaped into this document which describes technical and business requirements for the WCMS tool, service definition needs, function and design needs and tolerances, enterprise content management needs, and a list of governance topics for Web Council and Web Steering that may result in standards, guidelines, and policies.

The WCMS Requirements Gathering team members for this project included:

- Marion Bashista (ITS Portfolio Management Group)
- Ann Berry-Kline (ITS Portfolio Management Group and Project Manager)
- Jill Esteras (Division of Social Sciences Department Manager)
- Leslie Geary (ITS Portfolio Management Group)
- Andrea Hesse (ITS Divisional Liaison for Humanities)
- Sean Mikawa (Office of Admissions)
- Christina Navarro (Division of Physical and Biological Sciences Department Manager)
- Bomi Patel (ITS Divisional Liaison for the Chancellor’s Office)
- Julie Rogge (ITS Local IT Specialist for the Arts Division)
- Angela Thalls (ITS Local IT Specialist for Arts, Humanities and Instructional Computing)
- David Turner (ITS Portfolio Management Group and Web Presence and Services Program Manager)

1.1.2 Requirement Definitions

Requirements gathered were categorized to describe how important each requirement was according to number of times captured per session. Definitions for each requirement captured are:
• Critical - minimum requirement - applications that do not meet this function as stated will not be considered
• Very Important - highly desirable - applications that do not meet at least partial functionality are not likely to be considered
• Important - identified as desirable - applications may be selected over others on the basis of this functionality
• Nice to have - identified as functionality that may add value but is not likely to drive application selection

1.2 Overarching Themes from Campus Groups
The following themes were reiterated in all functional requirements gathering sessions and have implications for WCMS functionality. Specific requirements related to these general themes are identified further in subsequent sections.

1.2.1 Easy to Use
Content managers without exception emphasize the absolute requirement for a simple intuitive interface.

1.2.2 Consistency of Look and Feel Across Campus
All interviewees identified the need for a more consistent look and feel across campus. Some divisions also identified a need for added consistency across division and department pages. The WCMS must support centralized control over branding elements.

1.2.3 Importance of Local Design Flexibility
In addition to consistency across campus, all interviewees also stressed the need for individual design flexibility. Academic Divisions, in particular, require unique design to better distinguish their programs and successfully attract students, faculty, and researchers. The WCMS must allow both central control of some look and feel elements while also allowing local flexibility.

1.2.4 Importance of Authoritative Content
Each interview group identified the need for definition of authoritative content. It is desirable that the WCMS support the identification and use of authoritative content.

1.2.5 Robust Basic Tool is Highest Priority
Most interviewees feel that selecting a robust, user-oriented, and cost effective tool is the highest priority and that advanced functionality is a secondary consideration.
1.3 Key Requirements

1.3.1 WCMS Tool

Key functional requirements elicited from campus constituencies align well with the following objectives identified by campus leadership that include consistency in branding, distributed and easy content management, increased quality and currency of information, compliance with accessibility standards, and system stability and security. The highest priorities voiced in nearly all campus interviews largely echo the objectives identified at the outset.

1.3.1.1 Critical Functional Requirements

- Intuitive authoring tools
- Support for multimedia and syndication
- Mechanisms to manage currency of content
- Approval workflows and role based privileges
- Simple web forms
- Effective search engine
- Compliance with web and accessibility standards
- Support for integration with other systems
- Versioning, Archiving, and Audit Trails
- Mechanism for using rather than duplicating authoritative

1.3.1.2 Critical Technical Requirements

- Entirely web based content manager access
- Platform and browser independent
- Compatible with supported operating systems and databases
- Able to utilize campus authentication methods
- Meets critical security requirements
- Open standards and APIs
- Vendor support and use of best practices
- Ability to scale and meet performance needs
- Capable of effective and efficient recovery from failure

1.3.2 Web Presence (Non-Tool)

Requirements gathered are categorized according to the components comprising the Web Presence and Services Program for ease of distribution and reference. Requirements captured were then broken down into topics or themes.
1.3.2.1 **Function and Design Requirements**
- Redesigned suite of templates
- Information architecture
- Authoritative content sources
- Web specific style guidelines

1.3.2.2 **Governance Requirements**
- Standards and Policies
- Authoritative content source
- Cross-divisional user groups

1.3.2.3 **Web Services Definition Requirements**
- Training and documentation
- What is the service and how do I get it
- Costing model
- Defined roles and responsibilities
- Applications Library
- Adequate Resources

**1.3.3 Other Web Related Unmet Need**

Requirements were captured that were outside of the Web Presence and Services Program and its various projects. These represent other campus needs and potential integrations with web related content.

- Intranet
- eCommerce
- Integration with Other Campus Systems – AIS, FIS, LMS
2 WCMS Functional Requirements

The following functional requirements, collected from a wide variety of potential WCMS user constituencies across campus, align well with Web Presence and Services Program goals.

2.1 Content Creation

2.1.1 Authoring and Editing

2.1.1.1 WYSIWYG (Critical)

Ease of content editing was the most consistent and highest priority requirement voiced during requirements gathering. Due to a lack of technical expertise on the part of a majority of staff responsible for web maintenance today, an easy and intuitive content authoring interface that employs a familiar formatting toolbar or WYSIWYG editor is critical. An example of a simple WYSIWYG editor is below.

Other important functionality as part of a WYSIWYG editor includes:

- Ability to edit the HTML directly
- Spellchecker with customizable dictionary
- Automated compliance with Section 508 accessibility requirements

2.1.1.2 Form-Based Content Editing (Critical)

For structured content like a news article or faculty directory listing, which includes standard elements, forms-based content editing is critical. For example, standard elements for a news article might include Title, Author, Date, Body Text, Keywords, etc.

2.1.1.3 Navigation Control (Critical)

The WCMS should not impose navigation items on content authors. An example of current frustration with an existing WCMS system is that navigation is organized alphabetically and that top navigation bar categories were uneditable.
2.1.1.4  **Intuitive Content Search (Very Important)**

The WCMS tool should provide content authors and managers an intuitive interface for finding the content they wish to edit. The ability to find content is a given. Key to this requirement is usability, a subjective but nevertheless important consideration for UCSC where authoring is widely distributed.

2.1.1.5  **Preview (Very Important)**

The ability to preview content updates as they will appear to the public prior to publishing live is important particularly during the creation of new pages.

2.1.1.6  **Content Ownership (Very Important)**

The ability to associate an owner with a content object is especially important for automated notifications and effective management of information and staff.

2.1.1.7  **Link Check (Very Important)**

It is ideal if the application provides automated detection of broken links. External free tools are also available to detect broken links.

2.1.1.8  **Import Files (Very Important)**

The WCMS should be able to import multiple image, document, or other files via some sort of mass upload mechanism in addition to supporting individual file upload. This requirement is particularly important during site migrations.

2.1.1.9  **Staging Environment and Sandbox (Very Important)**

The application should support a staging environment of some kind that allows not only preview of a single content item or page prior to publishing but also a private area in which content managers can try new content ideas without affecting the production environment preview of a set of pages or entire web site prior to publishing.

2.1.1.10  **Clipboard (Very Important)**

The WCMS should have a clipboard system that allows easy cut and paste from one part of the system to another.

2.1.1.11  **Undo (Very Important)**

The WCMS should allow users to “undo” operations if they make a mistake. This is distinct from rollback to a previous version.

2.1.1.12  **Interfaces for Simple vs. Advanced Users (Very Important)**

Some applications either have or allow configuration/coding to support different user interfaces for simple vs. more sophisticated users.

2.1.1.13  **Context Sensitive Help (Important)**

It is desirable that the WCMS has an integrated context sensitive help system built into the product.
2.1.1.14 **Image Editing (Important)**
Relatively important to authors is a function to change the dimensions of an image. Another function that may be helpful would be a way to optimize/reduce the file size for web display. More advanced image modifications such as cropping, text overlay, and modifying color were not considered important.

2.1.1.15 **In-Context Editing (Important)**
Ability to browse to a particular page for content updates is considered important, especially for staff that make infrequent updates to their pages and require the easiest possible way to make quick updates.

2.1.1.16 **Content Check-In, Check-Out (Important)**
Although many staff members are solely responsible for editing content in their area of responsibility, a system that facilitates content authoring by non-technical subject matter experts should prevent concurrent updates.

2.1.1.17 **Clean Paste from Word Processing Docs (Important)**
Copying and pasting text from a word document into a WCMS can be problematic due to formatting and other information stored in the original document – a function that removes this information would be beneficial so that editors don’t have to cut and paste into a text editor first.

2.1.1.18 **Word Document Conversion (Would be Nice)**
The ability to convert a MS Word document into web content would be a benefit to some users.

### 2.1.2 Date Controls

2.1.2.1 **Expiration Date (Very Important)**
Some types of content are no longer pertinent after a certain date and should be removed from public view; event information is a good example.

2.1.2.2 **Review Date (Very Important)**
UCSC web contributors are often too busy to regularly review site content for freshness. Having a review date associated with content that reminds authors via email to update content is important. Review dates or time periods should be optional and adjustable to account for a wide range of review periods.

2.1.2.3 **Publication Date (Important)**
The ability to delay publication of content to the live site allows authors to develop content ahead of time and automate publication.
2.1.3 **Metadata**

2.1.3.1 **Tagging Content via Metadata (Critical)**

Metadata is utilized to classify and add meaning to content. Tagging is critical for effective site searching, organization around particular categories, establishing relationships by time, subject, or other category allowing users to specify parameters for searching, sorting, or filtering information. Tagging would be used, for example, to identify the subject matter of a news article, the owner of content, or an “alt tag” for an image as required by Section 508.

2.1.3.2 **Custom Metadata (Critical)**

Custom metadata fields are critical to allow content classification specific to UCSC business needs. This customization implies not only the use of variable set vocabularies but also the ability to add additional fields and the ability to determine the type of field (text, date, dropdown menu, radio button, etc.).

2.1.3.3 **Default Settings (Important)**

It is desirable that the WCMS allow content managers to set custom default settings for particular content types to make repeated content creation require fewer clicks.

2.1.3.4 **Global vs. Local (Important)**

The ability of divisions and departments to define individual metadata fields specific their own business need without intervention by a central support person is potentially important depending upon the complexity of their content.

2.1.4 **Custom Forms**

2.1.4.1 **Custom Forms that Generate Email (Critical)**

The need for custom forms across campus is ubiquitous and critical. The criticality of this function for a WCMS depends upon the complexity of the form and the extent to which this feature is available in these systems. A cursory review of WCMS products suggests that this type of functionality is common.

2.1.4.2 **Custom Forms that Store Data in a Database (Very Important)**

Some WCMS applications offer simple data collection forms designed within the interface to store information in the WCMS database. A WCMS is likely to allow for collection of data to an external database via application coding within a web page that interacts with an external database. Some WCM systems provide for “easy” creation of web forms that store information in the WCMS database.

When discussing web forms, it is critical to differentiate between “web applications” and web content management systems. Most modern applications are accessed via the web and involve moderately to highly complicated functionality that is clearly beyond the scope of a WCMS which is primarily designed to manage content for display on a web site.

There is a desperate campus need for custom web based forms; the extent to which this is appropriately managed by a WCMS depends upon the complexity of processing needs and the nature of the WCMS.
2.1.4.3 **Captcha (Very Important)**

A Captcha is a type of challenge-response test designed to prevent unwanted spam generated by automated completion of web forms by computers rather than humans. A common type of CAPTCHA requires that the user type letters or digits from a distorted image that appears on the screen.

2.2 **Content Requirements**

Various types and sources of content can be associated with a web site in addition to text.

2.2.1 **Content Types and Sources**

2.2.1.1 **Documents (Critical)**

Documents such as PDFs must be supported. It is desirable that authors can associate metadata with documents. Advanced document management functions are not required as part of the WCMS, although the need for a Document Management System (DMS) across campus is significant.

2.2.1.2 **Images (Critical)**

Insertion of image files must be supported. It is critical that users of the WYSIWYG editor be able to upload images. Ability to associate “alt tags” to images should be supported. Additional metadata to classify images is desirable. A common repository of images for public selection would be nice but is not a requirement.

2.2.1.3 **Structured Content (Critical)**

It is critical to be able to create types of web content that is structured. A news article, for example, might include a title, author, abstract, body text, categories, etc., which should be entered as distinct components (as opposed to being simply part of the page text) to enable context specific display of information. For example, one may want to display article title and abstract on their homepage, title as a link on an index page, full content on the article page, and limited components for display on a mobile device. Other types of structured content include staff directories, press releases, and event descriptions.

2.2.1.4 **Multimedia (Critical)**

Multimedia including video (streaming and stored) must be able to be displayed on web pages. The WCMS should allow video from external sources (example: YouTube) to be embedded.

It is desirable that multimedia files be associated with metadata which implies a basic level of digital asset management functionality. It is desirable that the WCMS captioning video and have control the media player. Advanced multimedia management is not required by the WCMS; however the need for Digital Asset Management (DAM) systems across campus is significant.

2.2.1.5 **Non-English Characters (Very Important)**

The WCMS should support basic non-English language character sets, for example, letters with diacritic marks.
2.2.1.6 **Syndicated Content (Very Important)**

The application should allow web authors to incorporate RSS Feeds into their web sites. It is desirable that the tool can read and parse syndicated content.

2.2.1.7 **Shared Content (Very Important)**

Shared content allows an authoritative source of information to share content with other users of the system in a way that the information is created and maintained only by the responsible office. Information reused on other sites is updated automatically.

One content sharing opportunity emphasized by most constituencies was a campus person directory.

Examples of other potential content sharing opportunities include course information, financial aid and admissions information, student handbook information etc.

One pain point for the academic divisions is the need to share information for interdisciplinary programs, courses, and faculty.

Another identified content sharing function would be to aggregate content from various originating departments into a single page. There are examples in which different departments maintain components of information that an end user legitimately expects to see in one place.

Note that methods for content sharing are diverse and content sharing, if extensive, requires significant planning.

2.2.1.8 **Integrations with External Applications (Very Important)**

External applications may be required to populate web pages or add functionality to the WCMS. Potential kinds of information from external UCSC applications might include course, staff profile, event, etc. data. Potential sources of data include AIS, CLP, Campus Calendar, etc. Note that integration may be as simple as consumption of content syndicated by the external application or quite complicated involving significant programming. Potential external applications to extend functionality might include search, web analytics, business intelligence, and other third-party products.

2.2.1.9 **Slideshow of Images (Important)**

Ability to incorporate a slideshow or rotating/random display of images on a site would be ideal.

2.2.1.10 **User Access to Widgets (Important)**

If common simple widgets or modules are available to plug into the application, it would be ideal if WCMS users could insert these themselves rather than requiring an administrator or programmer. An example might be a calendar or calculator widget.

2.2.2 **Content Repositories**

2.2.2.1 **Taxonomies (Very Important)**

It is important to have central and potentially local taxonomies for standard campus vocabularies.
2.2.2.2 Authoritative Content (Important)

Ideally, a WCMS would have a repository of authoritative content for content sharing (see “Shared Content” above).

2.2.2.3 Images and Logos (Important)

It would be a benefit to the campus to have access to a central repository of images. This type of functionality may be more appropriately delivered via a number of technological solutions from a specialized Digital Asset Management System to a manually maintained web site containing approved images and logos. Whether this function is provided by the WCMS or not, the need for access to images was consistently voiced across campus.

To the extent that a WCMS provides an image repository, the following characteristics would be beneficial:

- Association of images with metadata
- Globally accessible and department-specific image repositories
- Access to individual images based on role or group privilege

2.2.2.4 Multimedia (Nice to Have)

It would be a benefit to the campus to have access to a central repository of multimedia files. A Digital Asset Management solution may be a more appropriate technology for this function.

To the extent that the WCMS provides a multimedia repository, the following characteristics would be beneficial:

- Association of multimedia with metadata
- Globally accessible and department-specific multimedia repositories
- Access to individual multimedia files based on role or group privilege

2.3 Content Delivery

2.3.1 Publishing Targets

In addition to delivering content on a web page, WCMS applications are often capable of publishing to multiple presentation formats.

2.3.1.1 Syndication (Critical)

The WCMS must be capable of publishing content for syndication – typically for RSS feeds. Publishing should be driven by metadata like categories, date, keywords, etc. An example of this functionality currently in place is the News and Events tool implemented by the Public Information Office. Articles are categorized and RSS feeds are created based on category. Categories are nested such that, for example, a feed specific to articles related to the “Music” department are automatically included in the general Arts feed. Ability to syndicate ATOM and podcast feeds is ideal as well.
2.3.1.2 **Print (Very Important)**
The WCMS should allow for print friendly formats of web content.

2.3.1.3 **PDF (Very Important)**
The WCMS should allow for publishing web content to pdf.

2.3.1.4 **Handheld (Important)**
It is increasingly important for the University to publish handheld compatible versions of web content. The University Police and Fire Department identified this as a distinct emerging need.

2.3.1.5 **Publishing Different Versions of Content (Nice to Have)**
Some interviewees suggested that the ability to publish different versions of the same content to multiple distinct websites (i.e. for distinct audiences) may be beneficial.

2.3.1.6 **Desktop Publishing Applications (Nice to Have)**
Several users identified that it would be helpful to be able to publish web content to desktop publishing applications. As more and more units are managing authoritative content on the web, the ability to publish out to another application for generation of print publications may become more useful.

2.3.1.7 **Language Translation (Nice to Have)**
Translation of content to other languages was not identified as an important function but several users commented that it might be useful.

2.3.2 **Web Standards**

2.3.2.1 **WC3 (Very Important)**
Web content delivered by the WCMS should meet WC3 standards.

2.3.2.2 **Valid HTML (Very Important)**
The WCMS should generate valid HTML.

2.3.3 **ADA Compliance**

2.3.3.1 **Image Alt Tags (Critical)**
The application must allow for alt tags to be applied to images. The application should allow administrators to configure the system to enforce author compliance.

2.3.3.2 **Admin Control Over Styles and Templates (Very Important)**
It is important that administrators have granular control over the WYSISYG editor controls so that only appropriate styles and formatting is used.
2.3.3.3 **CSS Controlled Layout (Very Important)**

It is highly desirable that the WCMS use layout controls like CSS to make pages standards compliant, making the website accessible for people using alternate browsing methods such as handheld devices, Braille browsers, or text only browsers.

2.3.3.4 **Mechanism to Control Scripting Accessibility (Very Important)**

The following is actual verbiage from the Section 508 and WCAG 6.3 standards.

Sect 508, 1194.22(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

WCAG 6.3 (not cross-referenced by Section 508) Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

2.3.3.5 **Option to Include Summary Text on All Tables (Very Important)**

Tables’ summaries are textual descriptions of the contents of a table and help make tables that present data understandable for people using screen reading technologies.

2.3.3.6 **XHTML (Very Important)**

Publishing content as XHTML will help ensure it is standards compliant. The WCMS should emphasize semantic XHTML - where well-known and meaningful elements (h1, ul, li) are used in favor of idiomatic use of div and spans.

2.3.3.7 **Option to Make Image Alt Tags Required (Important)**

Requiring alt tags can be problematic if required of images that are, for example, blank spacers or purely decorative graphics like a curved image making up part of a background. In these cases an empty alt tag is appropriate. In the event that the application can distinguish between these types of graphics, this requirement should be considered “Very Important.”

2.3.3.8 **Option to Make Table Summaries Required (Important)**

Requiring table summaries is problematic when tables are not used to present data; for example, when they are used for formatting purposes.

2.3.3.9 **Video Captioning (Important)**

Captions are text versions of the spoken word. Captions allow the content of web audio and video to be accessible to those who do not have access to audio. Though captioning is primarily intended for those who cannot hear the audio, it has also been found to help those that can hear audio content and those who may not be fluent in the language in which the audio is presented.

Common web accessibility guidelines indicate that captions should be:

- **Synchronized** - the text content should appear at approximately the same time that audio would be available
- **Equivalent** - content provided in captions should be equivalent to that of the spoken word
**Accessible** - caption content should be readily accessible and available to those who need it

On the web, synchronized, equivalent captions should be provided any time audio content is present. This obviously pertains to the use of audio and video played through multimedia players such as QuickTime, RealPlayer, or Windows Media Player, but can also pertain to such technologies as Flash, Shockwave, or Java when audio content is a part of the multimedia presentation.

2.3.3.10 **Accessibility Reports and/or Checker (Important)**

The ability of a tool to adequately scan a site for Section 508 compliance is questionable. There are external applications that can be used to test for compliance but, again, their ability to adequately test a site is questionable. See: http://jimthatcher.com/testing.htm

### 2.3.4 Web Site Visitor Experience

2.3.4.1 **Search (Critical)**

The WCMS must either provide search capability or integrate with third party applications to achieve search functionality.

The tool or integrated solution must provide individual sites with site-specific search.

Some users identified that the ability to choose between site specific, UCSC-wide, or entire web when conducting a search would be ideal.

Some identified the desire for full text searching.

2.3.4.2 **Friendly URL (Critical)**

The WCMS must generate friendly URLs.

2.3.4.3 **Domain Names (Critical)**

The WCMS must support multiple (a large number) of distinct URL domain names (cnames). An example of a URL with a distinct domain name might be http://arts.ucsc.edu (as distinguished from http://www.ucsc.edu/arts).

2.3.4.4 **Secure Socket Layers (SSL) (Critical)**

The application must support pages protected by SSL encryption.

2.3.4.5 **Password Protection (Critical)**

The WCMS must support password protected sites.

2.3.4.6 **Search Engine Optimization (SEO) (Critical)**

The WCMS should generate sites optimized for search engines. SEO can be complicated. The requirement here refers to a basic level of optimization.

2.3.4.7 **Sitemap (Very Important)**

The WCMS should be able to automatically generate a site map and, ideally, an A-Z index.
2.3.4.8 **Breadcrumbs (Important)**
Ideally the WCMS would generate breadcrumbs automatically.

2.3.4.9 **Expandable Sub-navigation (Important)**
It is desirable that the WCMS allow for multi-tier, expandable navigation items.

2.3.4.10 **Email Address Encryption (Important)**
It is desirable that the WCMS prevent email spam by, in some way, protecting email addresses on websites from automated scanning mechanisms.

### 2.4 Site Management & Quality Control

#### 2.4.1 Change History

2.4.1.1 **Versioning (Critical)**
Versioning refers to the cataloging of changes to content over time. Versioning of content is critical to many campus users in order to refer to past versions of the content. Functions closely associated with versioning are:

- The ability to roll back to a previous content item version. (Very Important)
- The ability to easily compare changes between versions.
- The ability to add comments when changing content to describe the change or reason for the change.
- Systems may allow authors to publish different versions or “editions” to different areas on the web site. This was not a requirement identified by interviewees.

2.4.1.2 **Archive (Important)**
Archiving information is a complicated topic and may refer to providing

- Archives of information for the public
- A kind of a web record retention mechanism for internal purposes

Both types of archives were identified during requirements gathering as potentially important functions. Further definition of user requirements would be necessary to understand exactly what type of retention strategy is desired.

It is important to note that as more and more significant original content is maintained on the web site as opposed to within source documents, the need for versioning to track the evolution of content becomes more critical especially if record retention schedules and standards are defined. It is, however, also important to distinguish between a need for an archive and the need for recovery of existing web content in the event of a system failure or accidental deletion.
2.4.1.3 **Audit (Very Important)**

Tracking and auditing of changes to content includes cataloging of information like date of change, user responsible for change or approval, changes made, comments, etc. in an interface available to site managers.

2.4.2 **Publishing Workflow**

2.4.2.1 **Workflow: Simple, Flexible, Locally Administered (Critical)**

Publishing workflows allows for separation of duties between distinct WCMS users and includes an automated flow of tasks from user to another in defined stages. Simple types of workflow privileges include the right to author content, approve content, or publish content.

The majority of UCSC content managers want the option to set up simple workflows for all or some of their content in order to improve content quality.

Equally important to campus users, however, is that workflow does not unnecessarily delay publishing of information. Workflow must allow for some sort of override if changes are urgent and must provide a feedback mechanism if a user doesn’t complete their assigned workflow task in a timely manner.

2.4.2.2 **Workflow: Roles (Critical)**

With very few exceptions, workflow needs were described as simple. Critical roles and privileges identified include:

- Author – creates content
- Reviewer – approves content for publication
- Publisher – publishes content

Other roles identified as potentially important include

- Contributor – updates existing content
- Metadata Contributor – updates metadata
- Editor – reviews content for accuracy, grammar, voice, etc.

2.4.2.3 **Email Notifications and Alerts (Critical)**

The WCMS must be capable of sending email notifications of assigned workflow tasks. Ideal would be email alerts regarding workflow items that have not been completed in a timely manner.

2.4.2.4 **Workflow Queue (Very Important)**

The WCMS should have an interface that shows a user the workflow status of all items currently in a workflow that includes them.

2.4.2.5 **Comments (Very Important)**

It is important that workflow participants can save comments regarding a change or action.
2.4.2.6  **Cross Unit Workflow (Nice to Have)**
Some users mentioned that the ability to create workflow actions across functional units would be beneficial.

2.4.3  **Site Reports or Notifications**

2.4.3.1  **Site Analytics (Very Important)**
The WCMS should provide site analytics including hit counts, referrers, etc. or should integrate with external tools like Google Analytics or Web Trends.

2.4.3.2  **Content Inventory (Very Important)**
Some users felt that a report identifying all content or pages within a site and characteristics such as owner, review date, etc. would be helpful.

2.4.3.3  **Broken Links (Important)**
Identifying broken links during editing is covered in Section 2.1.1.6 above, however, many users felt that a report of site-wide broken links would be helpful.

2.4.3.4  **Search Analytics (Important)**
Some users felt that it is a benefit to understand user search criteria to identify common keywords, percentage of successful searches, etc. in order to continually improve metadata.

2.4.3.5  **Automated Content Relationship Tracking (Important)**
It is desirable that the WCMS maintain link integrity by tracking the relationships (or links) that exist between content items in the system such that if an item is moved or expired from the site, all related items are automatically updated by the system without end-user intervention.

2.4.3.6  **Orphaned Content (Important)**
Some users felt that a report of site-wide orphaned pages would be helpful. Orphaned pages are defined as pages that have no referring links. In some cases, orphaned pages are intentional. In other cases, they may either be outdated or, alternatively, wrongly missing referrers.

2.4.3.7  **Staleness (Important)**
Some users felt that a report or notification identifying potentially stale pages would be helpful. Stale pages are defined as pages that may require review for accuracy or currency. Stale pages might be those not updated within some pre-defined period, those with a passed “review” date, or other criteria.

2.4.3.8  **Content Status in Workflow (Important)**
Some users felt that a report identifying content or pages in some stage of workflow would be helpful. This report might identify content item/page, owner, status in workflow, task assignee, etc.
2.4.3.9 **Content Owned by Deactivated Users (Nice to Have)**

Identified as helpful would be a report or notification when site content no longer has an active owner due to reassignment or departure.

### 2.5 Application Administration

Administration, in this context, refers to configurations managed within the application interface and do not necessarily require a technical user. See WCMS Technical Requirements (below) for application technical support requirements.

#### 2.5.1 User and Group Management

1. **Basic Roles or Privileges (Critical)**

   Roles are ways of defining user functional privileges (read, write, edit, approve etc.) in the system. The number and privileges of roles required depend upon staffing levels, governance decisions, and local needs. Basic required roles include:

   - Author
   - Reviewer/Approver
   - Publisher
   - Site Manager/Administrator: manages local tool configurations including user role or privilege assignments, workflows, local taxonomies, local reporting, etc.
   - Template Designer
   - Global Administrator: manages global configurations that cannot be managed at a local level. Examples of potential global configuration items:
     - Custom role definitions
     - Global taxonomies
     - Assignment of local administrators
     - Global workflow options
     - System-wide report definitions

   Mostly likely, a programmer role should be available as well to allow permission based separation of code from content.

   User roles and privilege requirements should be further analyzed with constituents in the context of tool capabilities.

2. **Groups (Critical)**

   Different WCM systems define groups in different ways. Some simply manage roles. Others provide access to particular areas of the site. Still others make other distinctions.

   A clear minimum requirement for a WCMS for UCSC is the ability to assign responsibility for limited sections of the UCSC web site preferably in a hierarchical manner such that a division administrator, template designer, or other role may have access to all division web sites whereas access for other employees is limited to smaller areas of responsibility.
A beneficial function would be to identify work groups or teams that can access content or participate in workflows regardless of file system directory permissions.

2.5.1.3 **Granularity and Flexibility of Roles (Very Important)**

Roles should be granular enough to allow for adequate rules based on multiple division and department needs.

2.5.1.4 **Content Managers Not Directly Affiliated to the University (Important)**

Some content authors will not be UCSC staff, faculty, or students. It may be that the system should accommodate individual accounts in addition to a central authentication mechanism. (Note that this is largely a policy question but the system should not have limitations that in some way preclude non UCSC staff or students from managing content.)

2.5.1.5 **Multiple Roles Per User (Important)**

It is ideal to allow one user to have a different role when assigned to a distinct area of responsibility. For example, a user might be limited to updating a given page or site and assigned with greater or lesser privileges for another page or site.

2.5.1.6 **Custom roles (Important)**

It may be advantageous to have the ability to apply custom roles that would allow for increased granularity of privilege assignment. Whether this is a critical requirement or not depends upon the level of granularity present in default roles and permissions and governance decisions as to the ability of local entities to customize their role definitions. Complicated user roles can lead to trouble.

### 2.5.2 Administrative Quality Control Options

2.5.2.1 **Limiting WYSIWYG Options (Very Important)**

The WCMS should allow global or local web administrators to limit WYSIWYG functionality to prevent unacceptable formatting. Examples might include limiting text color and size to style sheet options or preventing blinking of text.

2.5.2.2 **File Size Limits (Important)**

This ability to set limits on file sizes is potentially important to prevent upload of massive image or video files. Huge files could impact performance or storage.

2.5.2.3 **Site Storage Limits (Important)**

This ability to set limits on the size of individual web sites is potentially important to manage storage limitations or performance.

2.5.2.4 **Content Item Discoverability Control**

It is desirable that the WCMS have the ability to control the discoverability of content items by external search engines.
2.5.3  **Local Administration**

2.5.3.1  **Manage Users and Groups (Very Important)**
Local administrators or site managers should be able to assign user roles, group memberships, and/or privileges within their own domain.

2.5.3.2  **Work Flow (Very Important)**
Local administrators or site managers should be able to create and assign custom workflow.

2.5.3.3  **Subsites (Very Important)**
The WCMS should allow for sub-sites within a site that are self-contained with their own navigation and content hierarchy.

2.5.3.4  **Taxonomies (Very Important)**
Local administrators should be able to create local taxonomies, vocabularies and/or metadata options to accommodate standardization of local verbiage.

2.5.4  **Templates and Style Sheets**

2.5.4.1  **Global, Central, or Master Templates (Critical)**
The WCMS must support global templates supported centrally that lock down certain areas or design elements to maintain a consistent campus look and feel.

2.5.4.2  **Multiple Templates (Critical)**
The WCMS must allow for multiple templates to provide multiple design options.

2.5.4.3  **Local Design Flexibility (Critical)**
Master templates must also allow for design elements locally including custom navigation items.

2.5.4.4  **Template Library (Very Important)**
The WCMS should provide some mechanism for end-users to select among template options when creating a new page.

2.5.4.5  **Template Update Propagation (Very Important)**
Changes to global templates should be propagated to child pages throughout UCSC.

2.5.4.6  **Style Sheets (Very Important)**
In addition to centrally-supported, global style sheets, the system should support the ability to add a stylesheet to individual pages or a sub-site.

2.5.4.7  **Creation of Local Templates (Very Important)**
Local template designers would like to be able to create and maintain local templates for pages. Local templates might be used for units that are significantly separate
from core university business. Possible examples might include the University Extension, University Affiliates, Summer Session, Research Entities, etc.

2.5.4.8  **Global Templates with Added Local Elements (Important)**

It would be desirable that the WCMS allow hierarchical establishment of template rules. For example, in addition to elements locked down centrally, a local template designer may be able to lock down items based on local rules. For example, a division may want to promote local identity by requiring additional look and feel consistency within its units.

2.5.4.9  **Template Creation Complexity (Important)**

Templates ideally would not require complex programming to create and maintain. Templates could be created with a simple text editor and do not require additional software to create/maintain.

2.5.4.10  **Workflow for Template Approval (Important)**

It is desirable that templates can be submitted to a workflow approval process.

2.6  **Other Functionality**

2.6.1  **Bundled MicroApplications**

The following functionality may be included in some WCMS systems or may be available as third-party widgets or integrated third-party applications.

It is worth noting that third-party application vendors who specialize in these technologies are more likely to provide a far more robust set of functionality as well as more timely updates.

2.6.1.1  **Event Calendar (Important)**

Event calendars were identified consistently as a primary pain point. In some cases, the need was for a campus-wide event management system. In other cases, simpler local calendar needs were voiced. It is unclear whether these local needs would be reasonably met with a campus-wide event system or whether a WCMS is an appropriate delivery mechanism. Some WCMS systems have some sort of calendar functionality and others have been used to build campus wide calendar systems. If any such functionality were to be included in a central WCMS solution, more detailed requirements elicitation would be critical. Multiple interviewees mentioned compatibility with CalDAV.

2.6.1.2  **Real-time Chat (Nice to Have)**

Some groups that provide office-hours type services to students mentioned that it might be nice to offer real-time chat with their customers.

2.6.1.3  **Email Newsletter Subscriptions (Nice to Have)**

Some campus units send out email newsletters to various constituencies using a variety of tools including external services dedicated to this function. Because some WCM systems include this functionality, it is worth mentioning here.
2.6.1.4  Blogs, Wikis, Forums, Social Networking (Nice to Have)

Some users interviewed mentioned various interactive functionalities such as blog, wiki, forum, and other social networking tools as something currently used or desired. Because some WCM systems include this functionality, it is worth mentioning here.

2.6.1.5  Survey (Nice to Have)

Several users mentioned the ability to conduct surveys and polls. Some WCM systems offer survey functions or the easy creation of custom forms that might be used as a survey.

2.6.2  Personalization

Personalization, in this context, refers to the generation of custom content or presentation for web users.

Note that personalization is particularly relevant to sales oriented web sites (i.e. Amazon) where previous searches, interactions, interests, etc. of web users are stored. Personalization is relevant to many intranets and is often better served by portal software. Personalization is complex and often resource (staff and system) intensive. Implementation often requires complex business process definition, content organization, and testing to achieve desired outcomes. Often substantial custom development is required.

Personalization is complex enough that separate personalization software solutions are often purchased and integrated with a WCMS.

2.6.2.1  Campus vs. Public (Important)

Currently the UCSC homepage distinguishes between campus or public visitors based on whether the originating IP address is associated with UCSC. To date, both campus and public pages have mostly or always contained identical content. The Public Information Office does not know, currently, whether this functionality would add value or not.

2.6.2.2  Roles Based Personalization (Pre-Defined Groups) (Important)

This type of personalization may be easier to implement and may not require specific software solutions. Site owners or administrators in this model assign specific end users the ability to view custom pages or content. Multiple users identified the ability to password protect pages for individual campus web users (using discrete authenticated sessions rather than shared passwords) was identified multiple times as a desired function.

2.6.2.3  Preferences Based Personalization (Nice to Have)

One issue clearly and frequently reiterated across campus involves information overload and the multiple potentially discrete audiences (faculty, staff, prospective students, current students, parents, etc.). Some method for simplifying the web experience based on these types of roles might be desirable depending upon the complexity of the mechanism. Ease of content management, a far more critical and ubiquitous requirement across campus, may conflict with personalization goals.

A possibly simpler approach is “customization” which involves identifying distinct audience segments and creating custom web sites for each.
Other potentially simple mechanisms not requiring specific technology solutions may be implemented to help distinct audiences find content.

### 2.6.2.4 User Homepage Customization (Nice to Have)

Homepage user customization allows users to customize their own home page with gadgets or content items most important to them. This feature is typically part of a specialized portal solution rather than a WCMS.
3 WCMS Technical Requirements

3.1 Technology Stack

3.1.1 Application User Interface

3.1.1.1 Browser Based Access (Critical)

The WCMS must provide full content management functionality via a web browser interface. The application must not require a fat client. The application interface must be accessible remotely and should be accessible wirelessly.

3.1.1.2 Platform and Browser Agnostic (Critical)

Additionally, the WCMS must provide full content management functionality and consistent user interface regardless of end user platform (PC, Mac, and Linux systems) as well as across standard web browsers for those platforms.

3.1.2 Supported Technologies and Hosting Requirements

3.1.2.1 Operating Systems (Critical)

One of the following operating systems must be supported by the application. Ideally, an application would be operating system agnostic.

- Windows 2003
- Windows 2008
- RedHat Linux Rhel 5.2
- Solaris 10

3.1.2.2 Databases Supported (Critical)

The vendor must certify the selected database as supported for the WCMS. The selected database and version must be supported by the database vendor and must be certified and supported by the application vendor. Preferably, the application would be database agnostic. The following databases are supported by ITS. No particular versions are required but the application would ideally support the latest accepted release from the database vendor.

- Oracle
- MySQL
- Microsoft SQL Server
Some important database considerations will be licensing costs, typical storage requirements, replication or mirroring, licensing for production/dev/test/staging environments, mechanisms for scaling, hardware requirements, NAS or SAN storage, etc.

3.1.2.3 **Web Servers (Critical)**

Supported web servers include:

- Internet Information Server (IIS)
- Apache

3.1.2.4 **Authentication (Critical)**

- **Shibboleth** is strongly preferred but is dependent upon not only the vendor ability to support it but also whether ITS is ready to offer it on the timeline required for implementation.
- **LDAP** is an alternative that may exist. ITS has not defined whether and under what circumstances integration with LDAP will be approved.
- **Kerberos** is an alternative if neither Shibboleth or LDAP are possible.
- **Local Authentication** should not be considered a viable option.

3.1.2.5 **Server and Network Architecture (Very Important)**

- Ideally, the application is certified to operate in a VMWare environment.
- The application should support load balancing of the web servers

3.1.2.6 **Messaging Protocols (Very Important)**

The software being proposed should fully support standard messaging protocols (e.g. SMTP, MIME)

### 3.1.3 Security

3.1.3.1 **Secure Authentication (Critical)**

UC requires that passwords must be transmitted in encrypted format. The system must be SSL compatible and should be configurable to switch from https on login to http for other pages. The system should also be configurable to switch to https mode for some pages or sections and back to http for others.

3.1.3.2 **Regular Security Patch Release (Critical)**

The vendor must release timely security patches as necessary. Ideally, the vendor can demonstrate proactive identification of potential security vulnerabilities.

3.1.3.3 **Session Timeout (Critical)**

The application must allow for session timeouts for both application users and application administrators.
3.1.3.4 **Firewall (Critical)**

The application must function behind a firewall and must not require open ports or protocols disallowed by campus security.

3.1.3.5 **Safe Error Handling (Critical)**

The product must not be susceptible to open failures like authentication bypass in the event of an application failure.

3.1.3.6 **Secure Transmission of Data (Critical)**

If the system contains restricted data, the application must support secure transmission. It is highly desirable that if the system contains any sensitive data, the transmission is secure. A WCMS likely should not contain restricted data or highly sensitive information; however, the need for password protected sites reiterated across campus suggests that some information is considered relatively private.

3.1.3.7 **Secure File Transmission (Critical)**

Any transmission of files directly to the servers should be secure. WebDav over SSL is preferred.

3.1.3.8 **Security Breach/Intrusion Support (Very Important)**

The vendor should provide support in the event of intrusion or other security-related incident. Ideally, this support would be 24/7.

3.1.3.9 **Password Strength (Very Important)**

To the extent that passwords are managed by the application rather than by UCSC authentication systems, the application should enforce good password strength in the following areas:

- Length (very important)
- Character types (very important)
- Lockout after unsuccessful attempts (important)
- History (nice to have)
- Expiration (including for initial or temporary passwords if applicable) (nice to have)

3.1.3.10 **Logging (Very Important)**

To the extent that authentication is handled by the application:

- Authentication attempts (very important)
- Activity audit trail (content changes and administrative/configuration changes including role changes)
- Log retention practices should be documented.

3.1.3.11 **Input/Data Validation (Very Important)**

The application should protect against SQL Injection or similar security issues related to input of data.
3.1.3.12 **Antivirus (Important)**

Ideally, the WCMS will provide some level of antivirus capability for uploading content via the application interface.

### 3.1.4 Additional Security Requirements for Hosted Systems

If the application is hosted remotely, the following additional requirements apply:

3.1.4.1 **Data Security Compliance (Critical)**

The vendor must be able to comply with each of the UCSC Data Security requirements in Appendix DS. These can be found at:

http://purchasing.ucsc.edu/forms/datasecurityappendix.pdf

3.1.4.2 **Minimum Connectivity Requirements (Critical)**

The vendor must be able to comply with UCSC’s Minimum Network Connectivity Requirements Policy, available at:

http://its.ucsc.edu/security/policies/minreqs.php

3.1.4.3 **Security Incident Response (Critical)**

The vendor must provide security breach response practices consistent with ITS security requirements. Ideally, the vendor utilizes robust intrusion detection systems.

3.1.4.4 **Backups (Critical)**

Data, including content, configurations, and logs must be backed up and retained in a way deemed acceptable by the university. Backup and retention practices must be documented. It is important to know where information, including backups, will reside. Foreign hosting may present a problem and should be evaluated.

3.1.4.5 **Uptime Expectations (Critical)**

The vendor must meet expectations of uptime for both the website and the content management functions. The vendor must perform system changes including those to servers, operating systems, databases, WCMS application, etc. in a manner approved by ITS with respect to managing potential disruptions of service.

### 3.2 Application Development and Administration

#### 3.2.1 Development and Integration

3.2.1.1 **Well Documented Open APIs (Critical)**

The application must provide for a robust mechanism for integrating external applications/data or extensions. The vendor should provide clear and complete documentation about best practices to prevent integration failures due to application upgrades etc. It is ideal if the vendor can demonstrate that previous upgrades have not caused failures to extensions.
It is important that UCSC resources are or can be made available for any required programming languages. Some current languages on campus include PHP, ASP.NET, ColdFusion, Perl, C, and Java.

3.2.1.2 Development and Test Environments (Critical)

For application development, development and test environments are critical. It is very important that effective mechanisms for publishing code changes from dev to test to production are available as well as the ability to roll back changes. It is very important that dev/test systems have safeguards against performing production specific functions. For example, dev and test instances must not send email notifications to client users during testing.

Staging environment may be a requirement as well.

3.2.1.3 Ability to Develop Additional Functionality (Very Important)

Developers report that it is important to be able to develop modules to extend functionality.

3.2.1.4 Embedding Code on Pages (Very Important)

Developers want to be able to embed code on pages and isolate that code from static content editors to prevent accidental code changes.

3.2.1.5 Web Services Compatible (Very Important)

The application should be able to use and serve external web services.

3.2.1.6 Open Accessible Database (Very Important)

It is desirable that the database schema is available and accessible to allow for custom queries for reporting purposes (not to allow customization of the core product.)

3.2.1.7 External Databases (Very Important)

It is very important that the WCMS can support integration with external databases via code within web pages to allow for dynamic web site data.

3.2.1.8 Application Code (Important)

Programming languages should not be proprietary.

Programmers report that it is ideal if the code base is exposed so that developers can understand how the application functions. A lack of this openness has caused difficulties in the past with custom reporting, for example.

3.2.1.9 Developer Community (Important)

Developer tool kit, training program, knowledge base, user groups, forums, user conference etc. are desirable.

3.2.2 Technical Application Administration and Support

Application administration in this context refers to technical administration provided centrally and may involve working via the web interface and/or managing installation and configuration of the application via server access.
3.2.2.1 **Support for UCSC Business Model**
The WCMS must support a business model with centralized application support/administration but distributed site management and administration.

3.2.2.2 **Application Support (Critical)**
The vendor must provide application training during implementation and on-going technical support.

3.2.2.3 **Cross Browser and Platform Compatibility (Critical)**
To the extent that application administration is accomplished via the web interface, the application must be platform agnostic (Windows, Mac, Linux all supported) and must meet

3.2.2.4 **Export (Critical)**
The WCMS should support export of web content.

3.2.2.5 **Vendor Release Practices (Very Important)**
The vendor should be able to demonstrate a regular release cycle and patching history as well as a history of good quality assurance. A bug list should be public.

3.2.2.6 **Patch and Upgrade Ease (Very Important)**
The vendor should be able to demonstrate that patches and upgrades have not caused major problems for clients. This includes major problems with integrations that follow vendor provided APIs.

3.2.2.7 **Administrative Interface (Very Important)**
The administrative interface and functionality should be usable and straightforward.

3.2.2.8 **Problem Notification (Very Important)**
It is desirable that the system have a mechanism for actively notifying administrators (email, cell phone, etc.) when it detects a problem.

3.2.2.9 **Import Structured Content (Important)**
Ideally, the application would provide tools for automated migration of structured information.

3.2.2.10 **Agility in Feature Development (Important)**
Ideally, a vendor would demonstrate agility in keeping up with and extending system functionality.
3.3 Performance, Business Continuity, and Disaster Recovery

3.3.1 Performance and Business Continuity

3.3.1.1 Size of Implementation (Critical)
The application must support thousands of sites, hundreds of thousands of pages, and hundreds of WCMS users. Many sites include rich media such as video.

3.3.1.2 Load (Critical)
The application must support web site load without system degradation in the event of major (but reasonably anticipated) spikes in load as might be experienced in an emergency situation.

3.3.1.3 Scalability (Critical)
The application must scale effectively as the university grows.

3.3.1.4 Web Layer Load Balancing (Very Important)
The WCMS should support running the web layer effectively in a load balanced environment. If the application layer and the web layer are coupled, this implies that both are capable of running in a load balanced environment with typical session management methods.

3.3.1.5 Caching (Very Important)
The WCMS should support caching of web content.

3.3.1.6 Database Replication (Very Important)
The WCMS should be compatible with database replication so that a local standby system can be replicated in near real-time. In the event that the application does not incorporate a database (for example: a product that utilizes a Java content repository), the system should have an analogous mechanism for maintaining a near real-time copy in the event of system failure.

3.3.1.7 Transactional Application (Very Important)
The application should be transactional.

3.3.1.8 Database Connections
The application should have a pool of persistent database connections

3.3.1.9 Redirection of Select Queries to Read Only Replicated Instance
It is desirable that the application can redirect SELECT queries to a replicated (read only) server to separate reading and writing in case one of the two monopolizes system resources.

3.3.1.10 Database Clustering (Important)
It is desirable that the WCMS operate effectively with a database cluster.
3.3.1.11 **Application Layer Load Balancing (Important)**

If the production layer is decoupled from the delivery layer, it is desirable that the application layer run effectively in a load balanced environment with typical session management methods. (Depending upon other factors such as WCMS performance, this requirement may be more important.)

3.3.1.12 **Application Layer Mirroring (Important)**

It is desirable that the WCMS support mirroring such that another application instance is maintained outside of production in the event of primary system failure.

3.3.1.13 **Squid and Varnish (Important)**

It may be desirable if the application is interoperable with Squid for caching and/or Varnish for http accelerating.

3.3.1.14 **Mechanisms for Managing Load Spikes (Important)**

The WCMS may support more advanced mechanisms for managing load. This might, for example, be the ability to switch to an alternate simple page or set of pages in the event of extreme load that threatens system integrity or reasonable performance.

3.3.2 **Disaster Recovery**

A disaster, in this context, implies a catastrophe resulting in an outage of the data center hosting the application. Detailed requirements are not complete; therefore, the following represents a current best guess.

3.3.2.1 **Disaster Recovery Support (Critical)**

In addition to performance and business continuity requirements, the application must support a disaster recovery strategy acceptable to campus requirements (currently being defined).

3.3.2.2 **Web Site Publishing to Remote Server (Very Important)**

The WCMS should support near real-time publishing to a remote server. Emergency personnel and public information officer must have the ability to update emergency content at a minimum.

3.3.2.3 **Database Mirroring to Remote Server (Important)**

It is desirable that the database be capable of near real-time mirroring to a remote server to minimize data loss in the event of a major catastrophe.

3.3.2.4 **Application and Database Mirroring to Remote Server (Nice to Have)**

The WCMS may support mirroring the application layer in near real-time to a remote server.
4 Web Presence (Non-Tool) Requirements

Web presence related requirements were collected from interviewees during the course of the requirements gathering sessions. They are organized according to the components comprising the Web Presence and Services Program.

4.1 The Web Function and Design Project

The objective of the Web Function & Design Project is to evaluate and improve the campus’s overall web presence. Components of this work include:

- Evaluating and recommending function and design of top tier campus sites
- Establishing a campus level information architecture - organizing the way we present ourselves on the web
- Producing a suite of web templates to update and improve the look of campus web pages
- Establishing design standards for navigation, branding, interoperability, and accessibility

Requirements gathered from the interview groups follow. Each section summarizes the critical gist of the requirements heard.

4.1.1 Redesigned Suite of Templates

Interviewees at every session expressed a need for a common but flexible presentation of UCSC. Templates must balance diversity and branding and allow individuality while maintaining a consistent UCSC presentation.

Users also identified a need for theme versions of the templates for multiple platforms. Templates should not be tool specific but rather useable across campus regardless of the CMS tool used by content owners.

4.1.2 Information architecture

Information architecture (IA) is structuring, organizing, and labeling information so that users can find what they are looking for more effectively and content owners can manage it more effectively. Interviewees agree this is needed at the enterprise/campus wide level and also at local the local level. Interviewees requested help in better setting up their IA to better market and present their sites.

Interviewees expressed a deep need for an improved IA that must also be able to provide the campus with wider enterprise model of working with the many diverse parts of our campus and to tie seemingly disparate bits of information into a more
unified layout. It should also provide logical organization, a sitemap, and basic design guidelines for consistent navigation and user interfaces.

4.1.3 Authoritative Content Sources

Interviewees expressed a high need for authoritative data to be identified for use across UCSC web sites without cutting and pasting or keeping multiple sources of data. This single source for content would be accessed and displayed on multiple pages across the campus web site. Examples include

- Scalable calendar and event schedules
- Course catalog and schedules
- Financial aid cost of attendance budgets
- Administrative policies

A corresponding requirement is outlined under the Governance section of this document includes assigning a governing body to declare and set policy regarding authoritative sources.

4.1.4 Web Specific Style Guidelines

Content developers and managers come with a range of web expertise. There is a need to have a common, UCSC web specific set of style guidelines and resources to support training and provide solutions to users.

4.2 Governance Requirements

One of the key components of this program is to establish campus level governance to set strategy, provide guidance, define priorities, and oversee standards and services related to UCSC's web presence.

Campus governance requirements gathered from the interview groups follow. Each section summarizes the critical gist of the requirements heard.

4.2.1 Standards and Policies

Web specific policies are needed for branding, connecting to external services or defining a standard Application Library (Flickr, Authoriz.net, blogs, wiki, calendar, web analytics, etc.), compliance (FERPA, HIPAA, PCI, and intellectual property), ADA guidelines, personal page usage and defining what an ‘official’ UCSC web page is.

4.2.2 Authoritative Content Sources

At every session, interviewees were in favor of a governing body to declare and set policy regarding authoritative sources. Authoritative sources of content solve a number of issues with multiple sources of data or copied pieces of content. The WCMS tool may allow for reuse or sharing of authoritative content.
4.2.3 Cross-divisional User Groups

Instituting cross-divisional user groups provides campus web presence a process for sharing ideas, monitoring change, and leveraging needs for both academic and academic support divisions and their units.

Key examples cited during the interviews were needs to have Admissions work with both divisional and departmental groups to create a unified, cohesive web presence for newly admitted students; coordination between Enrollment Management and individual departments for class scheduling information; and coordination across units (e.g. Admissions, CUHS, TAPS, Colleges) during move in week at the start of each quarter.

4.3 Web Services Requirements

The objective of the Web Service Definition Project is to define, articulate, and launch (as necessary) services to support the campus web presence.

Campus web service requirements gathered from the interview groups follow. Each section summarizes the critical gist of the requirements heard.

4.3.1 Training and documentation

Interviewees from every session expressed a need for adequate training and documentation. Training must be adequately resourced with sufficient staff and accompanying documentation. Training must be ongoing for end users, functional and technical support staff. Use of FAQs and development of a knowledge base were also recommended.

4.3.2 What is the service and how do I get it?

Key to the new service is defining it and providing the campus with information on how to get it. There is a need to define levels of service (basic, advanced, student groups, limited accounts).

4.3.3 Costing Model

This is a key issue for many units. There are a number of questions to address beyond what does “it” cost. Concerns centered on costs for more complex development efforts, the cost of storage and archiving, consultant assistance, and training.

4.3.4 Defined Roles and Responsibilities

Roles and responsibilities for web authoring must be well defined. Criteria to trigger approvals and design workflow are based on clearly defined roles and responsibilities. Definitions and application of roles must be consistently applied across campus. Workflows and content approvals must be flexible and representative of each web site moving into the WCMS.

These roles and responsibilities should also address who owns what content and who can update the content. Areas of concern are core campus information (campus policies), business information, academic information (teaching and research), and student and staff information.
4.3.5 Applications Library

Input from all interview groups was a need for a centrally maintained, shared library of small web applications for campus wide use. Applications identified were web forms, events calendars, directories (e.g. alumni, and divisional/unit staff and faculty), event registration, newsletter subscriptions, RSS subscriptions, surveys, wikis, and blogs.

The WCMS tool may lend itself to implementing some of these requirements expressed however many third party applications exist that better satisfy these requirements. A corresponding requirement is outlined under Governance section of this document to approve “standard” third party applications.

4.3.6 Adequate Resources

Interviewees expressed a need for adequate resources that are not currently available in today’s environment. Resources specifically identified are needed to support:

- Services – provide assistance with design, migration, mapping business processes, local IA, working with external vendors, and incident/problem resolution
- Operations - ongoing support for application and infrastructure changes, operations changes, process for feature requests and add-ons and change management
5 Other Web Related Unmet Needs

While out of scope for the Web Presence and Services Program other technology needs emerged from the interviews.

5.1 Enterprise Content Management Tools

A Web Content Management System (WCMS) is a single part of a broader suite of tools that comprise Enterprise Content Management (ECM). Interviewees identified a number of other ECM tools needed by a broad base across campus.

5.1.1 Collaboration Space

Interviewee elaborated requirements for a collaboration tool to provide

- shared information databases
- shared documents and other objects such as video
- components such as whiteboards for brainstorming, appointment scheduling, project management, etc.;
- communication application such as blogs, wikis, etc

5.1.2 Digital Asset Management

Digital Asset Management is the storage, cataloguing, and retrieval of digital assets, such as photographs, animations, videos, and music. It includes materials to support classroom instruction and various student production media.

UCSC has large investments in digital assets in University Relations, Arts Division, University Library, and ITS Instructional Technology Group. Interviewees discussed needs that centered on storage, retrieval, and archiving of these assets and the ability to support faculty instruction and student efforts to produce and store new content.

5.1.3 Document Management

Document management systems provide management of the creation and flow of documents via databases and workflow engines. These systems control documents from their creation through to long-term archiving.

Interviewees identified the following document management functions

- check in/check out of documents
- version management
- search and navigation for finding documents
The WCMS tool may be able to satisfy some DM requirements; it is not designed for this and may not satisfy DM requirements well.

5.2 Other Functionality

5.2.1 eCommerce
Requirements are wide ranging from the need to accept Admission’s Intent to Register fees to payment for event registration. Units are looking for standard tools to assist developers in adding an eCommerce element as a part of a web page. Developers must be assured the tools are secure and satisfy all UC and UCSC compliance requirements.

5.2.2 Intranet
This usually means a staff/faculty portal, personalized for individual needs and academic or administrative functions. It may require password protected areas for collaboration on things like HR tasks.

5.2.3 Integration With Campus Applications
Integration with other campus applications such as AIS, LMS, UNEX, and various course catalog databases is needed for shared information such as class schedules and descriptions, class lists, and student information. WCMS integration with campus applications needs to go through Web Council.
6 Appendix A: RFP Requirements Text

As part of the selection and procurement process, a Request for Proposal (RFP) was created in order to elicit vendor proposals to be scored based upon ability to meet requirements in relation to overall pricing.

- Critical requirements: Critical requirements must be met in order to be considered.
- High requirements: The vast majority of high requirements will be met in the selected product.
- Medium requirements: Medium requirements will have some impact on vendor selection.
- Low requirements: Low requirements make up a very small weight (2%) of the overall evaluation criteria.

The text of the requirements section of the RFP is below.

6.1 Critical Requirements

<table>
<thead>
<tr>
<th>Ability to Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>The WCMS product must support at least 500 WCMS users.</td>
</tr>
<tr>
<td>The WCMS must support a business model with centralized application support/administration but distributed site management and administration.</td>
</tr>
<tr>
<td>Web content management must be a key focus of the proposed product.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features and Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCMS must allow document (ie. pdf, MS Word) uploads as part of content authoring interface.</td>
</tr>
<tr>
<td>WCMS must allow image uploads as part of content authoring interface.</td>
</tr>
<tr>
<td>WCMS must be able to store structured content types. (Examples: news articles or person profiles).</td>
</tr>
<tr>
<td>WCMS must allow display of video (streaming and stored) on web pages.</td>
</tr>
<tr>
<td>WCMS must allow creation of custom forms for authoring of structured content.</td>
</tr>
<tr>
<td>Application must include a WYSIWYG editor.</td>
</tr>
<tr>
<td>The WCMS must provide “alt” tags for images.</td>
</tr>
</tbody>
</table>
### The WCMS must allow for the syndication of content.

### The WCMS must support password protected sites.

### The WCMS must either provide robust public search capability or integrate easily with third party products that provide robust search.

### The WCMS must be compatible with secure socket layers for encryption of information transmitted to and from a web site.

### The WCMS must generate friendly URLs.

### The WCMS must accommodate at least 300 distinct “cnames.”

### The WCMS must provide a LOCAL site (department, unit) administrator or “super-user” role that can manage local user privileges or role assignments and access rights via the application interface.

### The WCMS must provide a global administrator role that manages the application campus-wide.

### The WCMS must provide a LOCAL site (department, unit) role that can set up local workflows.

### The WCMS must limit user access to specific areas (possibly sites, pages, directories, repositories). For example, different divisions, departments, and units and sub units must have editorial access only to their own content.

### The WCMS must send email notifications of workflow tasks.

### The WCMS must support at least the following roles relevant to workflow: content author, content reviewer/approver, content publisher.

### The WCMS must allow the ability to set up simple linear workflows.

### The WCMS templates must allow for some locally managed custom design elements and custom navigation.

### The WCMS must provide for custom templates.

### The WCMS must provide for the creation and use of multiple templates.

### The WCMS must provide global/master templates to be supported centrally that lock down certain areas or design elements to maintain a consistent campus look and feel.

## Technology

### The application must be compatible with setting up individual development, test, staging, and production environments.

### The application must have well documented, open, standards based APIs.

#### IF SAAS: The vendor must be able to comply with the UCSC Data Security Appendix. This can be found at: [http://purchasing.ucsc.edu/forms/datasecurityappendix.pdf](http://purchasing.ucsc.edu/forms/datasecurityappendix.pdf).

#### IF SAAS: The vendor must provide security breach response practices consistent with ITS security requirements. Ideally, the vendor utilizes robust intrusion detection systems.


### The WCMS must function behind a firewall.

### Passwords must be transmitted in encrypted format.

### The application (not beta) utilized by clients today must provide full WCMS content management functionality via a web browser interface.
The WCMS must support at least one of the following authentication mechanisms: Shibboleth, LDAP, Kerberos (Note that authentication will use a single source.)

The WCMS application must be compatible with at least one of the following web servers: Internet Information Server, Apache.

If the WCMS application utilizes a database, it must be certified as supporting at least one of the following databases: Oracle, MySQL, and/or Microsoft SQL Server.

The server hosting the WCMS application must support at least one of the following operating systems: Windows 2003, Windows 2008, Red Hat Enterprise Linux 5.2, Solaris 10.

The application (not beta) utilized by clients today must provide full WCMS content management functionality and consistent user interface via major web browsers.

The application (not beta) utilized by clients today must provide full WCMS content management functionality and consistent user interface from web browsers on Mac, PC, and Linux.

The application (not beta) utilized by clients today must provide full WCMS content management functionality accessible from off-campus locations over wireless connections.

**User Experience**

User Experience: Content management functions, especially content authoring, approval, and publishing must be possible for non-technical users.

### 6.2 High Requirements

**Ability to Execute**

The supplier should be able to begin implementation within two months after a contract is complete.

How many users will the proposed system support concurrently? Has the proposed software ever been tested and approved (certified) for deployment to large numbers of users?

Has the company shown good performance (profits) in it’s last two fiscal years?

The company should provide on-site consulting during implementation. Is this included in a standard start up package?

The company should provide application installation assistance. Is this included in a standard start up package?

The company should provide consulting on system configuration in the context of client business needs. Is this included in a standard start up package?

Ability to Execute: The company should provide migration assistance and consulting. Would migration of an initial site be included in a standard start up package?

Will the next release of the product require significant changes to existing data or sites upon upgrade?

Please describe typical training provided during implementation.

Please list the universities using this product currently for an enterprise-wide implementation. It is not necessary that they serve the entire enterprise but that the scale of the implementation is large and would support enterprise-wide use.
### Features and Function

The WCMS should support basic non-English character sets, for example, letters with diacritic marks.

The application should have some mechanism for reusing authoritative content which is stored/updated in a single location for display on other pages in the system. Information on other pages should be updated automatically.

Please list file (document, image, etc.) formats that can be uploaded as part of the content author interface. (pdf, Word, Excel, jpg, gif, etc.)

Site authors or administrators should be able to incorporate external RSS feeds into their web site.

The WCMS should provide a mechanism for storing both university-wide and local (ex. Department) taxonomies or vocabularies.

Each content object should be associated with an owner(s) and it should be possible to reassign ownership.

The WCMS should incorporate an intuitive search mechanism for authors to locate content. Please describe how content authors/editors locate content to edit.

Content items should have a review date option that triggers notification to the author to review and update the content.

Content items like news stories or event descriptions should have an expiration date option upon which the content should be removed from public view.

Expiration dates should be configurable to trigger automatic removal without requiring further manual intervention.

Content items like news stories or event descriptions should have a go-live/publish date option that delays live publication.

WCMS should include a link checker that detects broken links during editing/publishing

It should be possible to add metadata tags to content items like text, structured content, and images.

Global administrators should be able to pre-define list options for commonly encountered metadata needs, for example a list of standard department names.

WCMS should allow for the addition of custom metadata fields to content objects and should offer standard field types such as text, long text, date, check box, radio button, drop down.

The WCMS should provide navigation menus that support a minimum of three levels of navigation.

Content authors/editors/approvers should be able to preview content updates as they will appear in context prior to publishing live.

The WCMS should support the creation of custom forms that send an email to an individual or group. Example might be a feedback form.

The WCMS should allow the creation of forms for simple data collection and storage.

Non technical administrators should be able to create simple forms.

WCMS should provide or be compatible with added captcha-like technology on public forms.

Custom form creator should be able to set optional validation criteria: required fields, date format, email format, url format

WYSIWYG editor should allow editing of HTML directly

WYSIWYG editor should include a spell checker

WYSIWYG editor should support 508 compliance

The WCMS should provide web site visitors with an option to print a print-friendly version of the page.

The WCMS should allow publishing of web site content to pdf.

The WCMS should allow for summaries to be associated with tables to accommodate people using assistive technologies.
Web content created by the WCMS should be WC3 compliant.

The WCMS should generate valid html.

The WCMS should provide administrators the option for making alt tags required.

The WCMS should allow administrators to control which WYSIWG functions are available for authors so that greater compliance with standards is enforced.

The WCMS should use layout controls like CSS to improve accessibility compliance.

The WCMS should provide or should be easily integrated with third party solution that provides site-specific search.

The WCMS should employ techniques for search engine optimization.

Forms for creation of structured content should allow standard form field types like text, long text, check box, drop down, date field/calendar, email,

The WCMS should be able to automatically generate a site map.

How can access to content be granted (ie. by directory, group, user, content type)?

The WCMS should either provide site analytics or should integrate easily with third party tools like Google analytics, Web Trends, etc.

The WCMS should provide site reports that identify content items.

The WCMS should provide local sites (department, unit) with the ability to view content inventory reports or summaries and review status of content in workflow.

The WCMS should provide for a global (campus-wide) privilege for defining custom roles.

The WCMS should include a global programmer role.

The WCMS should allow for hierarchical access rights to content. For example, an academic division web master may have access to all sites within that division whereas a department web master may have access to a subset of the same content.

A global and/or local administrator role should be able to limit WYSIWYG options to help prevent unacceptable formatting.

The WCMS should include a global template designer role.

The WCMS should provide local sites (department, unit) with the ability to set up local taxonomies.

The WCMS should provide for a global (campus-wide) privilege for defining taxonomies.

The WCMS should provide a staging environment that allows not only preview of individual pages but of sets of pages or a whole site prior to publishing live.

The WCMS should allow custom roles.

The application should provide versioning of content such that changes can be tracked over time.

The application should allow roll-back to a previous version

The WCMS should allow comments to be associated with content changes.

The WCMS should provide audit trails of content editing activity and they should be easily accessible by site managers/administrators. Pertinent information would be date of change, user, changes made, comments, etc.

Each workflow action should have the option to add a comment.

It should be possible to alter or override workflow dependencies to meet real-time needs (e.g. re-route workflow tasks)

Workflow should be configurable for different content areas or items. For example, a department may require that one set of pages has a rigorous workflow while another set does not need review prior to publishing.

The WCMS should have a user interface for authors and reviewers that shows the status of items in workflow.
The WCMS should provide for some mechanism for end-users to select among template options when creating a new page.

Templates should not require complex coding and should allow a designer to quickly edit and view changes in a staging environment.

Please describe how a new template is created, edited, reviewed, and published to production. What external software can be used to edit templates? What programming language is involved in template creation?

In addition to centrally-supported, global style sheets, the system should support the ability to add a style sheet to individual pages or a sub-site.

Changes to global or master templates should be propagated to child pages throughout UCSC.

Style sheets should be available and enforced to maintain consistency of content appearance via the WYSIWYG editor.

### Technology

The software being proposed should fully support standard messaging protocols (e.g. SMTP, MIME).

The WCMS bug list should be available.

IF SAAS: Data, including content, configurations, and logs should be backed up and retained such that the system and all information can be recovered quickly. Please describe service levels offered.

The WCMS should provide some mechanism for import of structured content (ex. News articles) especially during migration.

Technology: If the application does not use a database, please describe how data is stored and recommended mechanisms for backup, failover, and disaster recovery.

Technology: Please list browsers and versions certified for full WCMS content management functionality.

It is critical that UCSC have a disaster recovery strategy in the event that the web site can no longer be hosted or maintained in its primary location. Please describe disaster recovery options for deployment of this system. Note: this is above and beyond standard on-site redundancy. Does the application support mirroring to an off-site standby system?

Web content should be easily exportable.

The application should be web services compatible.

The application should provide controls that prevent dev, test, and stage environments from performing production specific functions like sending email to end users.

The WCMS should have mechanisms for graceful promotion of code from dev to test to staging to production. Is code considered part of the WCMS content or is it typically managed in an external source control application?

The WCMS should allow programming with asp.net, php, or other language to be embedded in web pages for dynamic interaction with an external database.

Please list what programming languages are necessary for integrating external applications.

Are the production and delivery tiers decoupled, coupled, or hybrid. If hybrid, please describe.

Please attach schematics for network server architecture options commonly implemented.

Please describe recommended network server architecture and system specifications for maintaining a high availability web presence for a mid size university with hundreds of WCMS users, hundreds of thousands of pages, moderately infrequent updates, etc. Include cpu, memory, disk subsystem, database engine, etc. Also include recommended database, OS, web server, application server software and versions. If other configurations are available, why is this particular combination preferred?

Please describe the application’s strategy for managing scale. Please include information as to when performance degrades as pertains to data set, concurrent user connections, load, etc.

The application should be transactional.

The application should be able to run effectively in a load balanced environment with standard forms of session management.

The application should have a pool of persistent database connections.
Please describe any mechanisms the WCMS employs to optimize delivery if there are major spikes in web traffic due to a major incident such as a natural disaster.

The WCMS should support caching of web content.

For any authentication managed by the system, authentication attempts should be logged.

For any passwords managed by the system, the application should enforce good password strength with length and character type controls. Desirable would be lockout after unsuccessful attempts, password history, and expiration.

The vendor should provide support in the event of intrusion or other security-related incident. Is this support available 24/7?

The WCMS should support secure data transmission.

The WCMS vendor should proactively assess their code for potential security vulnerabilities. Please describe how the company identifies security vulnerabilities. How often were security related patches been issued in 2008?

The WCMS should allow for session time-outs for both application content managers and administrators.

The application should protect against SQL injection attempts for form submission.

The application should provide activity audits for content and key configuration changes, including changes to user rights and privileges.

Please describe how the system manages authentication. Are user profiles obtained from the authentication source and stored locally? What profile information can be stored? Can custom profile fields be created?

Support

Please describe how clients request support, hours of support, limitations, service desk availability, response times, etc.

Please attach your privacy policy.

IF SAAS: The university website has very strict up-time requirements. Content management functions have less strict up-time requirements. Please describe service levels offered including maintenance windows.

User Experience

Workflow creation and interface should be intuitive and easy to use. Please attach screen shots that adequately represent the interface.

Administrative interface should be usable and straightforward. Please attach screen shots that adequately represent the interface.

One of the highest priority requirements at UCSC is that the WCMS content author, approver, and publisher interface should be intuitive and easy to use. Please attach screen shots that adequately represent the interface.

Template creation process and/or interface should be acceptable to campus designers. Please attach screen shots that adequately represent the interface.

### 6.3 Medium Requirements

**Ability to Execute**

Please answer the following questions: When was the proposed software FIRST originally released to the general market?

What is the release date of the CURRENT (last) release of the proposed software?

When is the next release of the proposed software planned?

Will the next release of the proposed software require any data to be converted to a format different than the format required by the current version of the proposed software?
It is desirable that the supplier has been in business at least 3 years under the same ownership.

Percent of company’s resources (people and money) dedicated to new product development?

Percent of company’s resources (people and money) dedicated to product support?

It is desirable that the supplier has experience with design and information architecture consulting for universities and can provide references upon request.

Describe a typical high level implementation project plan for a university implementation and key success factors for implementation.

Describe tools often used to migrate existing content.

Please describe the history of the product.

Please describe key upcoming features in the next release of the WCMS.

Please describe the company’s 3-5 year roadmap.

Please describe the primary focus of the product (eg. ECM, WCM, DAM, DM, etc.)

Features and Function

It is desirable that there is some mechanism by which authors can peruse a repository of authoritative content and select relevant content for their site.

A central repository of images and some mechanism for content authors to view and select them is desirable.

A local repository of images (ex. Department) is desirable.

It is desirable that the WCMS support random/alternating images on web pages.

Metadata associated with images in an image repository is desirable.

It is desirable that the tool can read and parse syndicated content.

It is desirable that the WCMS manage metadata for videos.

If simple widgets are available to plug into the application, it is desirable that they be pluggable without coding.

It is desirable that the WCMS allow content item check-out and check-in.

It is desirable that content item can be configured with the option to notify the author for manual removal.

It is desirable that, upon expiration of a content item, links to that content object in other areas of the system be updated automatically.

It is desirable that content items like images, documents, multimedia etc. have expiration date options that remove content from public view.

It is desirable that content items like images, documents, multimedia etc. have a go-live/publish date option that delays live publication.

It is desirable that the WCMS allow authors to change the dimensions of an image.

It is desirable that the WCMS allow reduction of image file size for optimized web display.

It is desirable that content authors be able to navigate to a page and edit content in context.

It is desirable that the WCMS maintain link integrity by tracking the relationships (or links) that exist between content items in the system such that if an item is moved or expired from the site, all related items are automatically updated by the system without end-user intervention. Please describe what types of content relationships are managed (images, documents, pages) and what occurs when a content item is changed or deleted.

It is desirable that local site administrators be able to pre-define list options for common local metadata needs.

It is desirable that documents and video can be tagged with metadata.

It is desirable that public forms for data collection include a file upload function.

It is desirable that creation of simple forms for authoring of structured content does not require coding or technical skills.
It is desirable that the WCMS support publishing to handheld devices.

It is desirable that the WCMS provide some sort of accessibility compliance content review.

It is desirable that video include captioning.

The WCMS should emphasize semantic XHMTL – where well-known and meaningful elements (h1, ul, li) are used in favor of idiomatic use of div and spans.

It is desirable that the WCMS provide the option for making table summaries required; however, tables are used for formatting should not have summaries.

It is desirable that the following Section 508 and WCAG requirements are met:

“Sect 508, 1194.22(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

WCAG 6.3 (not cross-referenced by Section 508) Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

It is desirable that spacer and simple formatting graphics not require “alt” tags even if other types of images do require them.

Ability to syndicate podcast and ATOM feeds is desirable

It is desirable that the application has been used by other universities to create a campus-wide events calendar. Please provide examples.

Display of different web pages depending upon originating IP address range is desirable. (ex. Currently the campus home page distinguishes between external and on-campus visitors by IP address.)

It is desirable that password protected sites allow web visitors to authenticate individually rather than with a common password.

It is desirable that the tool or integrated solution provide the option for local, UCSC-wide, or www search.

It is desirable that the WCMS allow for multi-tier, expandable navigation items.

It is desirable that email addresses on websites be protected from automated scanners that collect email addresses for sending spam.

It is desirable that the WCMS be able to automatically generate breadcrumbs.

It is desirable that the WCMS be able to automatically generate an A-Z index.

It is desirable that the WCMS have the ability to control the discoverability of content items by external search engines.

It is desirable that the application provide some custom reporting functionality.

It is desirable that the application be compatible with third-party business intelligence software.

It is desirable that the WCMS provide a site report that identifies the status of items in some stage of a workflow and might include content item owner, duration in current status, review date, task assignee, etc.

It is desirable that the WCMS provide a site report that identifies orphaned pages.

It is desirable that the WCMS provide a site report that identifies potentially stale pages. For example, a report that identifies content items with a review date in the past.

It is desirable that the application provide site reports that identify content items with owner, date of last update, review date, etc.

Desirable analytics include: page requests per month/week/day/hour for domain, site, page; successful vs. unsuccessful requests; referring pages; search words utilized and successful searches; etc.

It is desirable that the WCMS provide a local site (department, unit) role that can create custom public site forms for submission of data (ex. Feedback form, simple data collection form.)

It is desirable that the WCMS provide both simple and more complex user interfaces to accommodate different skill levels of content managers.

It is desirable that the WCMS provide a local template designer role so that global templates can be refined for local use.

It is desirable that file size limits can be set to prevent upload of massive image or video files.
It is desirable that the WCMS provide the ability to limit the size of individual web sites to manage storage limitations or performance.

It is desirable that the WCMS provide a local site (department, unit) role that can create custom forms for content authors to create structured content (ex. News article, persona directory)

It is desirable that the WCMS allow a global administrator to limit local administrator functionality.

It is desirable that the same user might have different privileges based on the site. For example, someone is an author for one site and a reviewer for another site.

It is desirable that access rights be relatively granular to site, subdirectory, page, and other content items.

It is desirable to be able to compare and easily view changes between versions. Refers to changes to text mostly.

It is desirable that images, documents, multimedia files are versioned also.

What are the options for workflow status? Can someone with appropriate privileges create custom status options?

It is desirable that templates can be managed in the workflow.

It is desirable that content items such as images and documents can be managed via workflow.

Is the ability to prioritize items in queues provided by the proposed software?

Email alerts for workflow tasks that have not been completed in a timely manner are desirable.

It is desirable that the WCMS support parallel workflows.

It is desirable that templates could be created with a simple text editor and do not require additional software to create/maintain

It is desirable that local templates can be created and managed locally.

**Technology**

Does the WCMS use OS/platform specific releases? Does the WCMS have installers for different OS/platforms, or does it need to be compiled/built from source? If the latter, are specific compilers needed?

Please describe whether the WCMS compatible/interoperable with any specific application development frameworks? How are customizations and extensions/plugins/modules best developed? Is there a specific dev framework that is needed?

It is desirable that a developer toolkit, training program, knowledge-base, user groups, and forum are available.

It is desirable that the code base is exposed so that developers can understand system function.

It is desirable that the database schema is available to facilitate reporting with business intelligence tools.

It is desirable that the application can redirect SELECT queries to a replicated (read only) server to separate reading and writing in case one of the two monopolizes system resources.

It is desirable that the application server can be load balanced across multiple instances. If clustering of the application servers is possible, please describe how user sessions are managed.

Is SAAS available for this product and is this recommended for a moderate to large university enterprise system?

Please list ports and protocols that must be accessible through a firewall.

It is desirable that the WCMS provide some measure of antivirus capability for content uploaded via the application interface.

It is desirable that the application can be certified to operate in a VMWare environment.

Does the application require a dedicated environment/server. If not, is this recommended?

**Support**

Under what circumstances would the product cease to be supported?
Please describe any tech library available that includes documentation for both the WCMS as well as bolt-ons/add-ons to their products.

Please describe your release schedule and process for patches and upgrades. Is there a release library and does it contain multiple alpha vs. beta vs. released versions? Does this library include base code as well as modules?

Please describe how clients submit bugs and feature requests. Is there a site where outstanding bugs and requests and their anticipated resolution schedule available for viewing.

<table>
<thead>
<tr>
<th>User Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the proposed software provide context-sensitive help, i.e., help specific to individual data entry fields or related collections of fields?</td>
</tr>
<tr>
<td>Is an online Help Menu offering an Index of Topics and searchable by keyword, similar to that found in Windows, available in the proposed software? Can the online Help be modified by a user with proper clearance?</td>
</tr>
<tr>
<td>If the proposed software provides for user-defined fields, can online help related to each individual field be created?</td>
</tr>
</tbody>
</table>
### 6.4 Low Requirements

<table>
<thead>
<tr>
<th>Features and Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access privileges to images or groups of images by role or group may be beneficial.</td>
</tr>
<tr>
<td>The WCMS may provide more advanced video management.</td>
</tr>
<tr>
<td>The WCMS may have the ability to convert a MS Word document into web content.</td>
</tr>
<tr>
<td>The WCMS may allow image cropping, text overlay, and color modification.</td>
</tr>
<tr>
<td>The WCMS may include a link checker that scans the web site for broken links - internal and external.</td>
</tr>
<tr>
<td>WCMS may allow creation of custom public facing forms for public authoring of structured content.</td>
</tr>
<tr>
<td>The WCMS may allow users to paste content from a MS Word document into the WYSIWYG editor without having to clean up extraneous data.</td>
</tr>
<tr>
<td>The application may provide language translation.</td>
</tr>
<tr>
<td>The application may publish content into a format usable by desktop publishing software.</td>
</tr>
<tr>
<td>The application may publish the same content to multiple distinct websites.</td>
</tr>
<tr>
<td>The application may provide real-time chat capability.</td>
</tr>
<tr>
<td>The application may provide a survey function.</td>
</tr>
<tr>
<td>The application may provide for email newsletter subscription and sending.</td>
</tr>
<tr>
<td>The application may provide a wiki function.</td>
</tr>
<tr>
<td>The application may provide a forum function.</td>
</tr>
<tr>
<td>The application may provide a blog function for web authors.</td>
</tr>
<tr>
<td>Web pages may be personalized to display information to distinct audiences (faculty, staff, students, public).</td>
</tr>
<tr>
<td>The tool or integrated solution may provide full text searching.</td>
</tr>
<tr>
<td>The WCMS may provide a site-wide broken link report.</td>
</tr>
<tr>
<td>The WCMS may provide a site-wide report of orphaned content or content owned by deactivated users.</td>
</tr>
<tr>
<td>The WCMS may provide a local programmer role.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application may be interoperable with Squid for caching.</td>
</tr>
<tr>
<td>The application may be interoperable with Varnish for http accelerating.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please describe any optional development/customization services that may be available above and beyond the proposed product/service either at the time of implementation or on-going.</td>
</tr>
<tr>
<td>Please describe any additional optional design consulting services that may be available above and beyond the proposed product/service.</td>
</tr>
<tr>
<td>Please describe any optional information architecture services that may be available above and beyond the proposed product/service.</td>
</tr>
</tbody>
</table>
Please describe any additional optional services provided for business continuity and disaster recovery that may be available above and beyond the proposed product/service.

Please describe any optional on-going training services that may be available above and beyond the proposed product/service.