

Redeveloping binghamton.edu

Implementing a smart, sustainable solution

Strategy Report: Appendices

mStoner

Table of Contents

Appendix A: Design & usability best practices.....	3
Appendix B: Models for determining necessary resources.....	7
Appendix C: Establishing an internal communications plan.....	10

Appendix A:

Design & usability best practices

Creating a flexible design suite

When designed well, a website allows an institution to express its strengths and points of distinction while guiding visitors toward the information they're looking for. In order to best express all that's great about Binghamton University, mStoner recommends that the University:

- ▶ **DESIGN A HOMEPAGE THAT ESTABLISHES THE BINGHAMTON UNIVERSITY BRAND** and sends visitors down the right paths for key information.
- ▶ **BUILD A FLEXIBLE, VISITOR-CENTERED ARRAY** of top-level pages.
- ▶ **APPLY BEST PRACTICES IN INTERFACE DESIGN THROUGHOUT ALL PAGES OF THE SITE.** These best practices are aimed at ensuring consistency, ease of access, and general usability standards.
- ▶ **CREATE AN INTERFACE THAT REFLECTS KEY ELEMENTS** of the brand.
- ▶ **VALIDATE THE VISUAL APPEAL OF THE SITE THROUGH CONCEPT TESTING AND EVALUATION.** Our process for design includes surveying concept designs in order to identify the visual elements that appeal most to target audiences.

WAI and Section 508 Guidelines

1. GENERAL

- ▶ Include a doc type for every page (XHTML Transitional or Strict)
- ▶ Include a language code on each html tag
- ▶ Write only standards-compliant HTML and validate

2. IMAGES

- ▶ Use alt tags (short and concise that relay the contents of the image)
- ▶ For spacer images, use ""

3. TEXT & LINKS

- ▶ Declare character encoding of all documents (covered by validation)
- ▶ Use structural rather than presentation text elements, including headings
- ▶ When spawning a new page, indicate so in a link
- ▶ Make use of the TITLE attribute throughout

4. NAVIGATION

- ▶ Use skip-navigation links for large navigation regions if not controlled through layout heading structure

5. COLOR & TYPE

- ▶ Don't use confuse-able colors for meaningful objects
- ▶ Specify foreground and background text colors in every document that specified text colors at all
- ▶ Specify a generic font family in each font declaration

- ▶ If pictures of text are used, do so only for brief segments, as in navigation bars
- ▶ Always use an alt that reiterates the actual text
- ▶ Use scalable fonts, not absolute values
- ▶ Include ability to manipulate font size [AAAA]

6. TABLES & FRAMES

- ▶ Use tables for tabular data
- ▶ Tables not to be used for layout purposes
- ▶ Use basic header information for data tables (<th> element)
- ▶ Use <caption> elements and summary attributes whenever possible
- ▶ Use scope attribute for complex data-table headers
- ▶ Use column and row groupings
- ▶ Table-less page layouts used for display of all non-tabular data

7. STYLESHEETS

- ▶ Use stylesheets to control presentation of documents; do not bother with accessibility-specific stylesheet media types
- ▶ Specify a generic font family in each font declaration
- ▶ Include multiple style sheets for "skinning" sites

8. FORMS

- ▶ Group related form options using <fieldset> and <optgroup>
- ▶ Always account for telephone modality (voice vs. TTY) and make allowance for the provision of alternate formats on request
- ▶ Do not code forms so that an action takes place as soon as the visitor makes a selection. Include a GO button (or similar function) so the visitor has control over what happens when using the form
- ▶ Make sure form field is labeled explicitly. If an explicit label cannot be used a title attribute will be added

9. MULTIMEDIA

- ▶ Set up a schedule to provide at least a transcript of the dialogue and meaningful sound effects of any posted online video or audio
- ▶ Use all available accessibility features in Flash as needed to meet requirements
- ▶ Provide captioning and audio description for online video as needed to meet requirements

10. CERTIFICATION AND TRAINING

- ▶ Write and post an accessibility policy. State the pages to which it applies—for example, new pages only or pages retrofitted after a certain date
- ▶ Certify the accessibility of your pages against a known standard, such as the Web Content Accessibility Guidelines priority levels. Note any known or deliberate deviations from those standards

mStoner will use the following validators:

- ▶ W3C/WAI: <http://validator.w3.org>
- ▶ Section 508: <http://cynthia.contentquality.com/>
- ▶ Cascading Stylesheets: <http://jigsaw.w3.org/css-validator/>
- ▶ Use new State of New York-approved accessibility validation tool: <http://nysf.hisoftware.com/downloads/index.htm>

Appendix B:

Models for determining necessary resources

Once a site is launched, its pages must be maintained. Shane Diffily, in his article “Manpower on a Web Team” (<http://www.alistapart.com/articles/manpoweronawebteam>), argues that you can assess staffing needs according to three models:

- ▶ **MODEL 1.** Size: the number of pages on a site
- ▶ **MODEL 2.** Complexity: the level of interactivity that a site offers
- ▶ **MODEL 3.** Level of Activity: the number of visitors who use your website

MODEL 1. Calculating hours by the number of pages on a site

A **TYPICAL*** web page has a shelf life of two to three years, and it needs to be updated roughly every six months. (*This doesn’t include content like news, events, and announcements that are posted more frequently and that have shorter shelf lives.) Each page requires approximately:

- ▶ Three hours of time to create initially
- ▶ One hour of time each year (assuming six-month review cycles) to update

Applying these calculations, Binghamton University should expect, in the course of a given year, the following investment:

- ▶ 150 hours for every 50 pages of new content created for the site (general pages, news stories, etc.)
- ▶ 50 hours for every 50 pages of existing content that must be maintained on the current site

General subpages can be maintained by web editors or administrative staff, but the University should realize that any given department with 100 pages will require 100 man-hours a year just to maintain existing content.

MODEL 2. Calculating hours by complexity

The first calculation serves general pages well, but websites are becoming more and more complex, and visitors expect ever-increasing levels of personalization and application development. The time required to develop custom applications varies with the type of application being developed. Some examples of programming hours necessary to program custom applications:

- ▶ University of Buffalo Site Search (www.buffalo.edu): 175 hours to build and test, one-half FTE to maintain
- ▶ University of Connecticut Virtual Tour (<http://admissions.uconn.edu/virtualtour/index.php>): 600 hours to build (and roughly \$30,000 in filming)
- ▶ Columbia College Student Portfolio (<http://tinyurl.com/jhdu6>): 20 hours
- ▶ University of Redlands Video Feed (<http://www.redlands.edu/x18.xml>): 20 hours
- ▶ Printer-friendly versions of pages: eight to 35 hours
- ▶ Incorporation of existing RSS Feeds: eight hours

Note, please, that these hours don't include HTML coding, design, content development, or maintenance hours.

MODEL 3. Calculating hours by level of activity: the number of visitors who use your website

This measurement assumes that busy sites inevitably have to deal with mountains of feedback, customer problems, and general issues of upkeep. This measurement is the hardest to apply concretely to staffing; and because universities routinely work in a centralized environment, tracking web-generated response and maintenance tasks should be done over time.

Perhaps a better application of this metric is "hours by level of internal support." Let's assume that a centralized team is responsible for providing training, support, and statistics analysis on a regular basis. Time estimates for those tasks would include:

- ▶ CMS training: 3.5 hours (up to 10 participants per training session)
- ▶ Editorial and accessibility training: 3.5 hours (up to 10 participants per training session)
- ▶ CMS technical support: an average of 20 minutes per incident (or three calls an hour)
- ▶ Statistics analysis, review, and recommendations: 4 hours per “client” per meeting

Appendix C:

Establishing an internal communications plan

Managing expectations and creating a sense of inclusion are essential to the success of this web project and all of the other initiatives that cascade from it. We recommend that the Binghamton Web working group keep the University community informed by:

- ▶ **CREATING A PROJECT SITE ON WHICH TIMELINES, UPDATES, AND PUBLIC DOCUMENTS MAY BE POSTED.** This site should be internal, accessible only to current students, faculty, and staff.
- ▶ **DEVELOPING A SCHEDULE—TIED INTO DEADLINES FOR PROJECT DELIVERABLES—FOR HOLDING TOWN-HALL MEETINGS.** Logical milestones would include the presentation of the approved project plan, the conclusion of concept testing, the start of content migration, the completion of beta-testing for the site, and site launch celebration.
- ▶ **DEVELOPING A SCHEDULE FOR ROLLOUT THAT ALLOWS UNITS WITHIN THE UNIVERSITY TO KNOW WHEN THEY'LL BE ABLE TO MOVE INTO THE CMS.** Those who can't move into the CMS immediately need to know how they should continue to update their content and who to turn to for support.
- ▶ **CREATING A TOOLKIT FOR UNITS THAT WANT A HEADSTART.** That toolkit should contain best-practice documents, templates to guide information architecture development, and training resources (either online or in person).