CS607 Website Design Process

1. Site Planning

A *static site development plan* is intended to expose the need for formal thinking and to bring everyone on a project into sync over the fundamentals.

- Requirements Analysis.
- Draw a site map.
- Assemble and structure content.
- Make a text-only site.
- Graphic design.
- Establish a maintenance plan.

Requirements Analysis (big part of Software Engineering) defines

- Audience.
- Purpose.
- Constraints (e.g. download time limits, browsers to support)

Producing a single page is easy, however multi-page sites must be organised. Web technology is based on *networked file systems* - structuring information is not as easy as in database systems.

- Understand the file structure (naming, permissions).
- Directory hierarchy.
- File versions.

2 Site Map

Areas that are expected to grow should be identified and perhaps given their own directory in the file system. These directory names become permanent once a site is public and can never be changed without breaking links from other sites.
The site map forces you to make decisions about whether the site is temporal or not, e.g. will the be a ``January issue'' section?
Will the site be

- Anchor service site (e.g. www.yahoo.com , Cetus Links - Object-Orientation) lots of functional links, user visits often, knows what the site is about and has a specific task in mind.
- One-time site lots of graphics, user was never here before, won't return (unless they are very brave). Most CS undergrads home pages are good examples of this!

3 Example site map - photo.net

Overall principles:

- This is an anchor site, not a one-time site. The cover page will therefore have links to as many sections and services as possible.
- photo.net will not be temporal. There will be no notion of a "January issue".

Directories (as found in photo.net)

- /doc/ for documentation on the server itself
- /doc/sql -- data model files

- /install -- files needed for installation only

- /global -- files served up by AOLserver when it gets unhappy (e.g., file not found or too many threads), also for privacy statements, etc.
- /graphics for site-wide logos and other images that aren't specific to content sections
- /ads -- banners ads to be served by /adserver scripts (below)
- /tutorial/ for a textbook for learning photographers, with its own index page and links from each chapter back to the index.
- /webtravel/ for travel guides to various photographic destinations. Multi-document guides with custom illustrations, e.g., maps, will have their own subdirectories.

4 Content

- Map nodes to filenames.
- Create files and insert content.
- Automation?

Style

- Sign your work.
- Give its status.
- Make links into context.
- Use context-free document titles.
- Format device-independently.
- Use standard HTML.
- Use style sheets.
- Write for the printed work too.
- Write readable text despite the links.
- Avoid talking about mechanics.

Text-only site?

- Turn content into legal HTML with navigation, ownership signatures, titles etc.
- Gives an overall feel of the site without substantial investment.

Graphic Design

- Bad graphic design is worse than no graphic design.
- Plain content is better than content free graphics.
- Graphics must add something extra.
- Careful thought before changing the way standard elements look (e.g. links underlined).
- Browser specific extensions (worse behaviours).

Maintenance

- Pasting new text into a document.
- Updating graphics.
- Automatically generated pages are ideal.
- HTML editors tend to produce pages that are hard to edit by hand (e.g. to make quick fixes etc, extracting content).
- Allow the users to edit the pages themselves. This leads to problems requiring some form of Source Code Control system.

Usability

- e-commerce site: how easy is it for the user to buy something.
- visits versus sales?
- Company sites: how easy is it to find information, download software updates, purchase stuff?

5 Summary

- Some formal thinking before you build a site will save a lot of grief and broken links later.
- Come up with a set of problems that you want addressed before consulting a graphic designer; "make it look pretty" is not a plan.
- Periodically do a usability test.
- Periodically ask fundamental questions about what you want your site to do and measure your current site against those goals.