Web Site Management: Creating and Managing a Web Site

12 June 2013

Helen Varley Sargan

Web site design has four main aspects:

- Planning the site
- Designing it
- Building it
- Maintaining information and structure

To make a website useful, usable and accessible, the planning and designing stages are vital and should usually run alongside each other. They create the structure and navigation of the site. If the design is already available (for instance if you use the University templates) you will need to familiarise yourself with how you use the template with your planned content.

1. Planning the site

You will need to create a development and management plan for the site so that its long-term future is taken care of. Continuous care will be needed, albeit at a reduced level, after the initial effort of setting it up.

Resources (human and hardware) and skills

Your plan should address these questions:

- Who is going to build and maintain your website, and where are you going to host it?
- If you anticipate contracting out the initial design and build, are you going to contract for maintenance as well, or will that be done in house (if so, by whom?)
- If a current member of staff is to maintain the site, do they have sufficient time and the right skills to do the job? Previous experience for in-house staff is desirable but not essential, but a forward-looking training plan must be in place and staff must have an enthusiasm to learn.
- Do you have any money - if not you will have to find some. You may need to pay for staff time for setting up and maintenance, use of a service, purchase of hardware and software, and training - both for courses and time for self-training; you may also want to pay for some design work.

Hosting of the site and design work required

There are a number of alternatives in the University both for hosting and for the availability of University templates. How you plan to host your website might be determined by what content and services you want to provide and whether your site is to use the University web templates, but the skills of staff available will play a part in the decision.

- If you are planning on running your own web server, the University templates are available on request from Barney Brown (barney.brown.admin.cam.ac.uk)
- If you are planning on using a design company, you may choose to have your website hosted outside the University. University domain names can be used with externally hosted sites (see http://www.ucsf.cam.ac.uk/info-inst/domain-names/external-references)
The Computing Service runs a range of options for web services (managed web server, virtual machine service and hosted server service) - see [http://www.ucs.cam.ac.uk/internet](http://www.ucs.cam.ac.uk/internet) and follow links for details.

The Computing Service runs the Falcon Content Management Service, which provides you with a website templated in the new University web style, with some choice in colours used. Various tools are built in, there is a help site and courses are run every month (see [http://www.ucs.cam.ac.uk/falcon/](http://www.ucs.cam.ac.uk/falcon/)). There is a charge of £100 per year, chargeable after 6 months development or when the site is made live.

In time a templated Drupal instance will be available from MISD.

The process of developing a site

---

### Hold a meeting

If you are creating a website for a group, be it department, college or research group, you need to **include** people in the planning process. If you do this they are much more likely to be supportive of the site and its aims and everyone will have something to contribute. Ensure that the person leading the group has a broad vision of what is required and what resources are available so that the group is reliably informed on what may or may not be possible. They must also be willing to make decisions.

Introduce them to the ideas outlined below - it would be best for you to think it through first so you can give a steer to the meeting (though not necessarily get your own way).
Recommended reading: Yale Web Style Guide (3rd edition) - ‘The Site Development Process’
http://www.webstyleguide.com/wsg3/1-process/7-development-process.html (from which the above diagram is taken)

Purpose of the site
What’s the web site for? The whole structure and emphasis of the site will hang on the answer to this! You must have content and credibility to bring people to your site and return to it. Create an outline of the information that you’d like to present, earmarking any information that is already available in another form - written lists, handbooks, research papers. Do not feel limited to using pre-existing information, often text written for booklets does not work well on the screen and it may be better to start again. Think of the context for your site - it must link to other information that is relevant be it faculty, department, or admissions information.

If you are re-designing a site, consider what has worked well or badly in the existing site. If you have analytics data, look at how the users travelled through the site and where they exited; if you haven’t, ask some users whether they found what they were looking for. Always look for evidence and then you will be able to show if you made an improvement. ‘Freshening up’ something that doesn’t work well is a bit of a waste of time and money - make it work better and not just how your boss would like.

If you want particular functionality for the site, work out what you want and list it up - this could include news system, showing rss feeds on web page, having directories of users, picture carousels.

Intended audience(s)
In many cases you want to appeal to a number of different audiences, which means you need to structure your web site with several different ‘ways in’, each clearly appropriate for a particular audience and giving them what they need. If you have many intended audiences, you will have to decide which are the most important to you and use those target audiences to create your main structure.

Information structure and content
Whoever is going to produce the pages for your website, you will still have to go through the process of planning your structure and content. Although some web designers can help with this (but will charge), by and large you are the expert on your content and how the user needs to access it, so you should be the one to create an outline of the information you want to present. With your purposes in mind, you can now go ahead and make a structure for your site, based on the content that you want your primary user groups to see. This stage will force you to consider the logical arrangement of information and the way users will have to navigate around your pages. Spending a little time making the server extensible in structure at this point is a worthwhile exercise.

How to do this? A flip-chart pad and felt pens, cards to write subjects on that can then be sorted into groups, post-it notes on the wall or on a flip cart, a pin board to use with cards, and a digital camera to record different arrangements. An article at ‘A List Apart’ gives a good view on how to do this - see http://www.alistapart.com/articles/taking-the-guesswork-out-of-design/.

When building a structure, you need to be aware that users will read much less written information on the screen compared with on paper. Reading paper publications is a process of scanning and threading together contiguous passages. When reading from a screen, users need smallish chunks of information, and if not, they wish to be able to print out and read from paper. They will not usually tolerate sitting and reading long passages from a screen. What this means is that handbooks, prospectuses or similar written matter need to be adapted into many chunks in a structure of their own (how to do this is discussed later).

Content of the home page
Now you have established aims, you need to plan out what you want on your home page, in terms of content. Decide what content is needed for each of the groups. This will overlap, but the analysis will allow you to consider which information is universal and which ought to be labelled as appropriate for a particular audience. Make the information as user-centric as possible - users don't want or need to know about the administrative structure of your department, but to find out about courses, research projects, events etc.
The University templates require images for the home page – at least one large picture (more to make a carousel), and smaller images if you want to make more impact with ‘teasers’. Many of the items below are taken care of in the template structure.

Content: recommended items

- ‘Corporate identity’ or logo with name of institute, optionally asserting copyright
- Postal address and email contact and/or webmaster address
- Lists of links to information in categories, grouped for particular types of user or types of information or other categories (short descriptions may be necessary). Ensure you do not add too many links otherwise the page will be overloaded and may be difficult to use.
- Option for quick links that don’t require description - may be particularly useful for local users or to access deep content.
- request for feedback (a comment form for users queries or direction to webmaster email)
- a search box - use the University web search
- a metadata description and some metadata keywords to facilitate worldwide searches finding your site and having meaningful search results
- Links to policies about privacy and the use of cookies (and maybe others, such as freedom of information/data protection, information and contacts for disabled visitors, and accessibility of the site).

Beware of icons. It is extremely difficult to design icons that are universally understood or legible to those with poor eyesight.

Wording

After you have decided on what links to have, think carefully about wording and grouping. The comprehensibility of the information should go hand-in-hand with the layout of the web pages. The web is an international medium and the wording for links needs to be clear and unambiguous. If you wish to appeal to different language groups it may be desirable to make your site available in several languages (remember you will have declare the character set(s) used on your web pages as well).

Throughout the site try to think of the content both in terms of your site in isolation and in terms of the communities it belongs to - this may affect the wording you use and extend the reach of the site. Think also about what words you might expect people to use to look for your site - these can be added as metadata (which is not seen on the page but is seen by search engines). The fourth section of this document tells you about how to add it to your pages.

Structuring your content (information architecture)

Classically, you ought to aim for a hierarchical structure with a home page at the top of the tree, and several major submenus within branches of your tree. The major links off the home page are the branches of information. If you are setting up a very large body of information you may end up with several such trees connected to the home page, rather than a smaller number of excessively deep trees (which will force the users into clicking through many links to finally arrive at the information they seek). This gives ‘browse functionality’ by contextualising content within sections. An ideal is for the user to reach their goal (in subject area if not in detail) in three clicks. See http://www.webstyleguide.com/wsg3/3-information-architecture/3-site-structure.html for further help on creating a navigable structure.
Directories or folders
The level down from the home page should be structured into suitably named directories or folders, each with a default page (how you provide a default page will depend on the web server or content management you are using). The name of the folder will appear in the URL, so should be a proper word, or hyphenated proper words - this need not be the same as the title of the page since it’s probably desirable to keep the URL short and succinct.

If you are manually managing your content, placing each obvious section of information in a directory will make it much easier to manage the server and to add new information. If you are using some kind of content management system (cms), putting content in folders will be how your navigation is created. For the server to be used effectively by untutored users, and managed by others, the structure ought to be evident. An example of directory structure mapped onto the page appearance is shown below (from http://www.webstyleguide.com/wsg3/5-site-structure/3-site-file-structure.html).

Ensure you plan a straightforward naming scheme see p 18 for details about naming considerations.

Within folders, further folders can be added to give a more detailed tree structure. Pages can be added to any folders or independently at the top level of the site.
Dynamic or database-generated content

Most CMSs will produce content with navigation aids created by collecting information into folders, which may be visible folders or may just be categories. Some allow you to create short folder and page names (rather than hyphenating the title of the folder or page); if it’s possible to add this feature to sites produced by a CMS (via an optional feature or plug-in) then it will be an advantage as it will produce shorter and more usable URLs.

Example structure of a department

The darker boxes are directories/folders and the lighter are pages.

Index = default page for the folder

Pages do not need to be named at .html - the name will depend on the server technology used.
2. Designing the website

University web page templates

If you want to use University pages, there are currently only templates available on demand, or through using the Falcon Content Management service. Please don’t take these templates and alter them - they are designed to integrate with other sites in the University on the web and any changes will possibly disorientate users and detract from the original design.

If you use the Falcon content management service, you can select from a number of colour schemes and features.

Creating your own templates

Getting a theme for your site

All users prefer to have a fast-loading page with no obstacles or delays. Any pages that have additional bells and whistles should occur after initial entry to the site. Don’t believe that you need to be trendy and employ gimmicks - good looking pages with clear information structure and layout is much more important.

Unify your site by giving the same overall design to the home page, second-level pages and end level content pages. Users need continuity when they are going through links on the same site, and an overall identity will also ‘brand’ your pages as being from the same source.

Choose a neutral background colour and utilise as many conventions as possible to ensure users find your pages straightforward. Users have expectations about where elements of the page will appear (see Where’s the Search? Re-examining User Expectations of Web Objects - http://usabilitynews.org/wheres-the-search-re-examining-user-expectations-of-web-objects/) - make the most of this.

Recommended reading: Web style Guide - see http://www.webstyleguide.com/wsg3/6-page-structure/3-site-design.html

The home page

The home page sits on its own. Although it should have the 'look and feel' of the site it is not usually the same as the other pages - it has a different function and use.

- Ensure it has links and grouping to give the basis for navigation throughout site
- Ensure it has a rapid load time (see webanalysyer, available through the Firefox web developer toolbar Tools > View Speed Report or http://www.websiteoptimization.com/services/analyze/).
- Ensure it has changing information on it, so regular visitors see something new - this could be featuring changing news, events or pictures.

Second level pages

The second level and subsequent pages will need maximum space for content, so the least amount of space should be taken up by branding and titles. You should have a template that includes the standard items of the page, placement for any alternative things that might appear (lists, pictures, tables, all levels of heading) and gives room for at least a global navigation so users can get back to the home page, search and get help. Decide whether you need a standard navigation bar that helps users round second level and even third level pages, and allocate where that will appear on the template.

Some studies of how users read content (see http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/) indicate that shorter pages that use headings are most likely to be read more thoroughly.
Look at other designs you like
When you see other designs you like, look at the source HTML for them and find out how they work. You can find out what works well and what doesn’t by using other pages as a testbed.

Creating a navigation scheme
Navigation is essential to direct users around your site. Navigation devices allow them to enter the site at any point and to get to other areas they might want. The crucial point with any navigation system is consistency - the user must know where on any page to look for assistance. For maximum visibility, placement should be at the top of the page and/or at the left side. Users are likely to miss navigation placed on the right.

You should consider having a (minimal) general or global navigation bar (including the search box), which appears on every page and allows users to stay orientated. Many sites (such as IBM [http://www.ibm.com/] and [http://www.bbc.co.uk/]) have such a scheme.

Local navigation can be a navigation bar scheme such as in the Undergraduate admissions site ([http://www.study.cam.ac.uk/undergraduate/life/accommodation.html])

or a simpler scheme such as on the Computing Service site, which includes a 'breadcrumb trail':

In any navigation scheme, remember to give the user a cue for their location - as shown above, use a second colour to show the section the user is in.
The ‘breadcrumb trail’ tracks through a useful set of steps that the user may have followed or could follow to widen the categories of information, for instance the scheme that is used on http://www.cam.ac.uk/. This can be very useful, but depends on your content being well classified and not heavily crosslinked, when breadcrumbs get tricky.

Other navigation bar schemes are possible using:

- ‘Collapsed’ navigation, which, when selected opens an area of the page and closes others (not used very often, but see http://tympanus.net/Tutorials/CollapsingSiteNavigation/).
- Pop-down navigation bar items with fly-outs, such as at http://www.lib.cam.ac.uk/

**Use several approaches together**

People may want different sorts of navigation on the page - **global navigation** to let them use the search and help, or to start again; a **navigation bar** to give them sections of the site; **breadcrumbs** to climb out of information to the levels above; and **left-hand navigation** to move around a circumscribed body of information. For example see http://www.ucs.cam.ac.uk/support/winsuptech/activedir/omindesign:

![UCS University Computing Service](image-url)
Mobile navigation

If you want the site navigation to be effective on mobile and tablet devices (responsive design), this needs to be carefully built into the site. A responsive design will usually dispense with the left-hand navigation, so the site navigation bar should provide this in the mobile view. In responsive and tablet views it is also usual to collapse any navigation bars and make them available via a navigation icon (which has become fairly ubiquitous).

Accessible navigation

Some forms of navigation can present accessibility problems. Least accessible are image maps, since the links are not available unless the image can be used - an alternative mode of navigation will be needed if an image map is used, and pull-down menus, the contents of which (if they are javascript driven) may not be readily available at all. Pull-down menus also have the disadvantage that the user has to actively seek the navigation rather than it being there up-front, and they can often miss it. Use of graphics can work fine as long as appropriate alt information is used, sometimes alongside the ‘title’ tag in a link (which some browsers use to give a fuller description on-screen). Text and tables have no problems.

References: Creating navigation aids and menus


Making a grid

Next stage is to make a grid or wireframe of how you want the pages. If you are getting a designer to give you some options, this will be an excellent starting point for them, and if not, it will be a good starting point for you!


On a very basic level, you need to allocate areas of the page for functionality. This is an example of a grid that gives all the options needed, not all of which need to be on every page (diagram from [http://webstyleguide.com/wsg3/4-interface-design/3-interface-design.html](http://webstyleguide.com/wsg3/4-interface-design/3-interface-design.html)):
To create a version of a grid that’s suitable for you, try starting with a flip-chart part and felt tip pens (see second article below for different strategies).

Before you start, it is interesting to see how users look at web pages - http://usabilitynews.org/eye-movement-patterns-on-single-and-dual-column-web-pages/.

Creating templates

With a grid of what’s wanted, the next step is to create a code framework that will deliver that grid. To create this you will need expertise in coding and writing CSS, or you will need to employ someone who has these skills.

The technical recommendations that follow will give you an opportunity to create a more detailed breakdown of your design and highlights issues that you need to consider, many of which are raised as issues in the following section about accessibility. If you are using a CMS (including Falcon) then some of these recommendations will be managed by the system.

Technical recommendations for designs

Dimensions:

- If possible, design the home page to fit into one screen, so the user doesn’t have to scroll vertically, but do not let this overrule the content. If the content doesn’t fit into one screen, try to have some feature that indicates it extends vertically, so users don’t think they’ve read it all when they get to the bottom of their screen.

- Do not tie your design to a particular width of window, make it dynamic

Users have surprisingly low tolerance of window sizes that vary from their ‘usual’. They will get fed up of scrolling, and may not notice there is more text to read or a graphic is cut off.
**HTML/XHTML code:**

- Learn what the HTML/XHTML standards describe and what elements commonly used browsers support, and always have the dtd at the top of your page. For current usage xhtml 1.0 or html4 are the most likely. Html5 is still upcoming but should be used with caution because of backwards compatibility issues.

- Use HTML/XHTML and CSS (http://www.ucs.cam.ac.uk/web-accessibility-guidelines/pages#style) rather than adding fixed mark-up in the content

- Use structural mark up (http://www.ucs.cam.ac.uk/web-accessibility-guidelines/pages#structure) rather than images or JavaScript, and avoid using tables for layout purposes

- Check your HTML/XHTML and CSS with validation tools (http://www.ucs.cam.ac.uk/web-accessibility-guidelines/pages#validation) to make sure they contain no errors and with an accessibility tool to ensure a wide range of users can view it (see http://www.w3.org/WAI/) - the Firefox web developers toolbar gives direct access to do all these tools. Validation of the code (http://validator.w3.org/) and checking accessibility by disabled users (http://www.w3.org/WAI/) are absolute requirements.

**Colours (http://www.cam.ac.uk/site/accessibility-guidelines/pages#colour):**

- If you use a background colour and/or graphic, always set text and all link colours as well

- Use the ‘web-safe’ colour set (216 colours) to assist with cross-platform and cross-browser consistency [not so critical for developed world now, but still important for third world users]

Set the link colours to ensure that they over-ride any user-set link colours. Don’t underestimate the value of the default links colours that users expect.

Always have good contrast between text, link and background colours so those with visual problems can see the text. Beware use of red/green colour schemes as they will be indecipherable for those who are colour blind. Suitable testing tools are linked from http://www.cam.ac.uk/site/accessibility-guidelines/pages#colour

**Images and multimedia content (http://www.ucs.cam.ac.uk/web-accessibility-guidelines/pages#images and http://www.ucs.cam.ac.uk/web-accessibility-guidelines/pages#multimedia):**

- Always use alternate image tags. Only include text for alternate tags for images that are important, for decorative images create an empty alternate tag and users who can’t see the screen will not be aware there was an image there.

- Optimize your images so they are only as large as you need on the website, both in dimensions and resolution, and include the dimensions of an image to speed loading the page.

By putting in the dimensions of the graphic, the browser knows how much space to allocate to the graphic when it starts loading, rather than working it out after loading has finished.

**Moving images should be used with caution on-screen (by default) for the following reasons:**

- they can be very irritating and movement may distract the user from reading the page

- video should be offered in several different formats and linked to an alternative - it should not be set to autostart. You should provide a transcript or summary and, if the video is for a wide audience, also captioning.

- the words within images are not indexed in search engines and are not readable to those who cannot see the image, unless you make special efforts to present a transcript.
Javascript and other dynamic technologies (http://www.ucs.cam.ac.uk/web-accessibility-guidelines/pages#scripts):

Very few people use the web with Javascript turned off, but how your pages work without it is still something you need to have thought about. Don’t use Javascript when xhtml/html would work instead. Techniques for providing accessible java and javascript are detailed via links on the above page.

A synopsis of technical requirements:

- make your pages dynamic, not fixed width (If you must use fixed width, use a ‘standard’ width of, say 800 pixels, set with the stylesheet)
- set a required standard for HTML/XHTML, add a DTD to every page, and validate them
- Use stylesheets (CSS) rather than formatting mark-up. Ensure that the page makes sense if the stylesheets are not used. A single multipurpose stylesheet, with additional ‘special use’ stylesheets will be the easiest to manage and maintain.
- if you set a background colour, always set text and all link colours too. Standard navigation colours from the web-safe palette should be used if possible (these ought to be all set in the stylesheet)
- be cautious about the number of images you use. Always optimise graphics and use descriptive alt tags unless they are purely decorative, in which case use a null alt tag (alt=""")
- moving images, back-end databases, javascript, java may all reduce or block access to the information for some - learn the precautions you need to take.
- test against a number of different graphical browsers (and different versions of the same browser) on different platforms
- test for disabled access, which covers a number of needs:
  - those with impaired vision may use a browser that reads the text to them (which cannot manage complex tables easily) or may need to set their screen character size larger than the default (so text that is not dynamically sized will cause problems)
  - choose colours with good contrast for people with poor sight
  - choose colours suitable for those with the commonest forms of colour-blindness

Accessibility requirements

The Equality Act 2010 mandates that web information and services provided by an institution or part of an institution, either as information or services for students or prospective students or teaching materials, must be accessible, and you must anticipate this requirement.

Accessibility applies to:

- web pages, including forms
- web interfaces to databases
- online teaching materials
- catalogues and indexes
- any other online resources

TechDis makes available some useful training materials for staff involved in web development or web management (http://www.jisctechdis.ac.uk/techdis/investinyourself) and details about creating accessible materials (http://www.jisctechdis.ac.uk/techdis/userneeds).

There is an overview of the accessibility of University pages at http://www.cam.ac.uk/about-this-site and a full University Web Accessibility policy, which was updated in April 2010/May 2011, at http://www.cam.ac.uk/about-this-site/accessibility/university-of-cambridge-web-accessibility-policy. The policy sets out why it is required and makes the following policy statement:

All web pages should be assessed by the guidelines published by the Web Accessibility Initiative (WAI) from the World Wide Web Consortium, known as Web Content Accessibility Guidelines (WCAG) 2.0, available at http://www.w3.org/TR/WCAG20/. The University requires that:
• All new web pages should be written to at least conformance level 2 standard (AA) but to conformance level 3 standard (AAA) if possible.
• All existing pages should meet at least conformance level 1 standard (A) of Web Content Accessibility Guidelines (WCAG) 1.0 (http://www.w3.org/TR/WCAG10)/,
• Most pages should meet conformance level 1 standard (A) of the newer guidelines by 1 September 2011. A development plan should be in hand to make all pages conformant to at least A level within as short a time as possible.

Departments, faculties and research groups and other groups that publish information on the web are responsible for being conversant with accessibility issues, auditing their web material and taking reasonable steps to ensure their websites comply with these requirements. Any third-party who is engaged to design web pages for the University, whether hosted within or without cam.ac.uk, will be required to comply with these guidelines. Sites will be checked periodically.

How to make your web pages accessible
There are guidelines for how to make content accessible available at http://www.ucp.cam.ac.uk/web-accessibility-guidelines/here is a synopsis of what they say:

• Make sure your content is properly structured, using headings, paragraphs and lists, and is syntactically correct.
• Web pages must validate against the dtd on them, then you can have a good idea what browsers are going to show to users. Check them against a range of web browsers (the most recent two versions of MS Internet Explorer, Firefox, Chrome and Safari).
• The content should make sense when no stylesheets are used to make it look nice, no images are shown, and no adjunct technologies (such as javascript or flash) are available to the user.

Excellent information about web accessibility may be found at the WebAIM site (see http://www.webaim.org/articles/) and also on the RNIB website (see http://www.rnib.org.uk/professionals/webaccessibility/designbuild/Pages/design_build.aspx.

3. Building and testing the website
How to manage your content
Web page content can be inserted by hand into template files, which can be useful if you have only a few pages, otherwise it is very time consuming and subject to error. Increasingly content is being managed in content management systems, but there are more simple ways to manage content in templates. Responsive templates can be very complex and contain functionality that is impossible to create manually, so use of a CMS makes their use possible and management of content easier.

Building pages without using a database can utilise a set of templates. There are a number of ways they can be used:

• pages produced manually from template files
• pages produced from template files in an application, for instance in dreamweaver
• use of server side includes (ssi) to add parts of the template to pages
• use of a scripting solution to add parts of the template in an intelligent way (for instance HTML::Mason + Poet)

Databases
The use of databases on the web is usually referred to as being at the ‘back-end’, because they hold the information for the page in a raw form that will be finally presented possibly in a number of different ways.

There are two basic ways that databases can be used on the web:

• as simple containers of information that is then put into a template to form the pages that make up
a website - the principal of a content management system

- as a repository of information that can be actively queried and from some or all of which a page is created in response. In addition a database can be set up to acquire information from a user (see later for a note on this use).

Databases will only be effective with suitably structured data.

Simple templating and automating

Pages created from a database into a template framework are no different to standard web pages - the pages are assembled using a script that takes the data and inserts it by some means into a template, which is then stored in a static form until a further update of the content takes place. It is virtually impossible to detect pages that are generated in this way as their final appearance is of a static page.

Virtually any database can be used in a simple way to generate pages by inserting database fields into a template composed of the html Although this has advantages, the downside is that unless the set up is very detailed the components of the template that could be dynamic, such as navigation, do not get changed, so it’s only suitable for simple sites or tranches of content.

Information management

Using and repurposing Wikis and blogs

At a simple level both wikis and blogs are websites. Several technical users have looked at the wiki as an ideal way to have direct input to web pages and have developed tools for this re-purposing.

Blogs can also be used more broadly for publishing a website (see http://manila.userland.com/ but other blogging tools can be used). Since version 2.5, WordPress has had options that allow it to publish a website instead of a blog - see http://codex.wordpress.org/Pages for information.

Frameworks

There are a group of products that put in place a framework for managing information, on top of which can sit a more specialised module for content management. They are often, but not solely, open source, and a number are listed here:

<table>
<thead>
<tr>
<th>Product</th>
<th>Type of product</th>
<th>CMS add-ons or other info</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML::Mason (now Mason + Poet)</td>
<td>Open source (Perl)</td>
<td>The Mason+Poet framework itself can be developed for use (see <a href="http://www.masonhq.com/">http://www.masonhq.com/</a>). Bricolage is a mason cms - <a href="http://bricolagecms.org/">http://bricolagecms.org/</a> (both undergoing development, 2012 and 2013)</td>
</tr>
<tr>
<td>Zope</td>
<td>Open source (Python)</td>
<td>Plone - <a href="http://plone.org/">http://plone.org/</a></td>
</tr>
<tr>
<td>Contensis</td>
<td>Commercial (Asp)</td>
<td>Integrated - <a href="http://www.contentmanagement.co.uk/">http://www.contentmanagement.co.uk/</a></td>
</tr>
<tr>
<td>Terminalfour Site Manager</td>
<td>Commercial</td>
<td>Integrated - <a href="http://www.terminalfour.com/">http://www.terminalfour.com/</a></td>
</tr>
</tbody>
</table>
In 2011 (the last year it was run) the Packt Open Source awards the following all were mentioned: Drupal, Joomal, Plone (see http://www.packtpub.com/open-source-awards-home/).

http://www.cmsmatrix.org/ is a handy CMS comparison tool. Open Source products may require Apache with particular modules running, such as cocoon, Jakarta (for Java), Mod-Perl, WebDAV.

Content management systems

As mentioned above, broadly speaking content management is done through using a database for taking page content and inserting that content into one of a range of templates, usually through the following process:

Content management systems (CMS) may be bespoke, intended for commercial organizations, and cost very large amounts of money; or they may be open source, used by anyone with enough technical skill, and the software may be free-of-charge. No CMS is free. The most important aspects of a CMS are:

- that they suit your requirements and the way your team works (both for building and maintenance of the system and for maintaining the content) and their proficiencies - this is the most usual failure point for CMS installations
- cost - not just for buying the product but for training, keeping content up to date and changing templates or instituting page design changes. Make sure you know what you are getting
- lock in - when you want to change product or the company goes belly-up, can you get your content out again?

See ‘How to Select a CMS’ (http://www.contenthere.net/2007/05/how-to-select-a-cms.html) - an excellent article distilling some of the problems associated with choosing a CMS (see also http://www.contenthere.net/2011/06/cms-selection-checklist.html).

In an institutional setting, what you choose depends a lot on the skills and money you have. If the site you want is fairly simple it could be that WordPress will do the job for you. If you have any Java expertise available, then Hippo may be worth evaluating. Likewise if you have php experience, Drupal might be a good choice. They all required work or one sort or another.

The Falcon content management service (see http://www.ucs.cam.ac.uk/falcon/) gives you a fully featured site (based on Plone) with all management of content and configuration through the web.
Problem areas
Major difficulties arise from several places.

- Development time and effort can be extensive and involve highly skilled staff and/or lots of money. Upgrading will be necessary on a regular basis as new versions of systems, CMS and plug-ins become available.
- Usability and complexity - the system must be usable by those contributing and updating information, else more skilled staff will be used for this process (when the system was meant to reduce their workload). Help will be needed for the preliminary structuring of the site.
- Technically, thought must be given to making friendly and usable URLs, so that the pages can be referred to, indexed and bookmarked.
- Training - contributors will need training even when systems are simple to use. The need will be ongoing.
- A CMS can only solve so many problems. Getting content and having it updated will still be as difficult as it ever was.

Accessibility and testing
Tools to use to view and test web pages
Many of the suggested tests can be easily accomplished by using a good html editor or

- Firefox and the web developer toolbar (which installs into it) - see https://addons.mozilla.org/extensions/?application=firefox for listing of extensions.
- A similar developer toolbar available for Microsoft Internet Explorer (built-in tool in IE8 and IE9) or http://www.visionaustralia.org.au/ais/toolbar/ for accessibility toolbar.

Accessibility testing tools
- Test your templates for accessibility with a validation tool that reports on possible problems for disabled users. Check sample pages against two of:
  - Total Validator (http://www.totalvalidator.com/)
  - Cynthia Says (http://www.contentquality.com/)
  - WAVE (http://wave.webaim.org/) particularly good for seeing the logical structure of pages
  - Web accessibility checker (http://achecker.ca/checker/index.php) - very well formatted results
- If you use a technology that requires a plug-in (such as Acrobat or Flash) give a link to a suitable site to pick up software. Adobe now provide an accessibility tool with Reader and more information about accessibility and providing accessible pdfs (http://www.adobe.com/accessibility/products/acrobat/); they also have some information about Flash, and accessibility solutions for Flash MX and more recent versions (see http://www.adobe.com/accessibility/).
- Test your templates against a number of browsers of different versions and on different platforms, (optimally) also with the graphics turned off (this is as well as checking the HTML). Opera and Firefox with the web developer toolbar installed (http://www.chrispederick.com/work/firefox/webdeveloper/) have tools to allow you to test accessibility. There is also a tool (http://www.browsercam.com/) that you can use to view how a range of browsers see your page. It is free for trial use and after that a small charge per day of use.

If a user has to do anything special or have any specific plug-in to read your pages, you risk losing them if they don’t. There are special instances, such as the Acrobat reader, where a note and a pointer towards a source for the reader if they don’t already have it would suffice. Consider whether your Javascript is cross-browser compatible.
Co-workers and external workers

Local Guidelines

Ensure that you have an adequate set of guidelines or recommendations for your server and/or other server managers on your site. Existing and new information providers should be supplied with the guidelines to help them understand the issues behind accessibility problems and these guidelines must be distributed to any web designers you employ or consider employing.

Contracting out the design of a website

You should set out the structural requirements for the site - how many different versions of a first design you want to see, what the home page should contain, what colours to avoid, how many other templates you’d wish, or, if you wish the complete site produced, how many pages are required. In addition, you should also produce a technical specification for the site, including the following:

- the level of html (DTD) that you wish to used. You should insist it is checked against the W3 validator at http://validator.w3.org/, which is accessible directly from the Firefox web developers toolbar
- it should follow the WCAG2.0 accessibility guidelines -level 2 or AA is a good one to aim for
- it should be cross-browser friendly, not written for a particular group of browsers. Provide a list of supported browsers.
- information should be available as text, not locked away in graphics such as Flash presentations
- if you want to use content management, say what technologies you can support or which open-source product you want. Don’t use web designer’s own CMS unless you want to be permanently wedded to them.
- if you are hosting the site, say what web server software you will use and what server features you want or don’t want to utilize. Would you like workers to be trained in the software and/or the system to be used for updating information?
- If the contractor is hosting the site make sure you can get the content back from them if you part company, and there is adequate agreement for the upkeep of the site both in regularity and pricing.

Often designers will not follow these specifications and you will have to check all aspects at every step.

Converting documents

The useful and simple conversion of pre-existing documents revolves around the structuring of the original document.

The process has three aspects:

1. the difference between how a browser works and looking at the printed word and how this must affect the appearance of the document;
2. the conceptual difference between material to be read online and the printed word, and how this must affect the structure of the document;
3. the strategy of moving the content from one format to another.

Appearance

The appearance of your document. If your audience needs to see your design or print out the document as a whole, you should supply a PostScript or PDF file for viewing, printing and/or downloading. A print stylesheet can be used to give a better quality of print out of a web page. See Further CSS course for more info and guidance (http://web-support.csx.cam.ac.uk/courses/)

Trying to re-create the appearance of the paper document on screen is usually a mistake. The print and screen media are so different that the attempt will usually come to grief. Aim to produce a readable and clear on-line version. Do remember that readers of English and other European languages read from a straight left margin. Text that is centred or ranged to the right is difficult for them to follow. Knowing
the limitations is essential: do not use multiple columns for text or attempt to use fancy fonts that may be unreadable on screen. Remove any pictures that are not necessary - screen space is valuable.

Structure
The structure: The document should be broken down into chunks that create a logical structure and are of digestible size. A document with three chapters should initially break into three, each of which is pointed to from an index file. Any large chapter may have a number of sections, so can be broken down further into a file per section - this chapter now needs an index pointing to each of the separate sections. The whole can then be given a logical navigation to get from one part to another.

If you remove many graphics from the original document you may have to adapt the text to do without them. The number and spread should be kept in check so that only the most pertinent are included. Be sure you always have an adequate description appended to the image tag (the alt tag). Users of speaking browsers depend on these tags. Graphical images add weight to the download times, so be careful when adding them.

The possibilities for cross-referencing (putting in links) are vast, but should not be overdone. Links between files that make up the document, which will allow users to 'flick through', are essential. When adding links, always make sure the link text is appropriate - don't use links in the form of 'Click here for the latest version'.

Strategy for converting paper to web
There is a wide range of converters or filters to change a formatted document to HTML. The effectiveness of the transfer hinges upon the formatting of the documents, particularly whether styles have been used. A Word document marked up on an ad hoc basis will not convert as well as a properly structured document for which styles were used, as the formatting in the styles can be interpreted by many converters whereas manual mark-up cannot.

Conversion from a particular format to HTML or XML can usually be accomplished, but depends on the program used. InDesign, Word and others have some conversion available. Creating a template honed for the purposes of export (with a built-in set of styles, for instance) is usually a good option, in some cases scripting could handle the process. Always investigate new versions of software for conversion features - over time they are gradually becoming better but many will give you far more 'conversion' than you bargained for (or wanted).

Dreamweaver can rationalise HTML from Word, and other HTML editors have built-in version of HTML Tidy to ensure your files conform to the DTD you’re using.

Practical problems: file names and line ends
When you start a web site, create a simple naming scheme for directories and files, keeping names short but as whole words when possible, and using a readily predictable case. If anyone is creating files for you, let them know your naming scheme before they start. Keep in mind that writing HTML is a cross-platform activity and one of the aspects of different platforms is the way files can be named. What is acceptable on all platforms is not immediately obvious. If moving files from one platform to another you need to be aware of the differences otherwise maybe your files may become inaccessible or your links won't work. If you ftp or sFTP HTML files by an automatic method, remember to set the transfer filetype as text, and for other files (graphical, audio) as raw data. Underlines in filenames can be difficult to see if you use the filename in the web page - better to use a hyphen instead.

Keep your filenames (or short names) simple, without spaces or slashes. Use real words when possible as they will do better in Google searches. If you are moving files between platforms:

• Unix filenames are case sensitive and should not include characters such as spaces, slashes
• PC filenames and Macintosh filenames are not case sensitive so make sure the case of the filenames used is consistent
If you transfer files between platforms, take care that you use a compatible line-end character. In good HTML editors you can alter which line-end characters your file takes as part of the process of saving the file.

**Usability testing**

Once you have built your home page and second level pages, it is a useful exercise to do some usability testing with some of your more important user groups. To glean results that give any sort of clear picture you need to test at least five individuals or pairs, and record their progress through a set of appropriate tasks. There is a software package called Morae (http://www.techsmith.com/morae.asp) that can be used for recording audio and video of a session, but you will need some training to know how best to perform the tests and analyse the results. You can do iterative usability testing if you have time, so test out whether your changes have made any difference. For full guidance on how to proceed, see two books by Steve Krug - ‘Don’t make me think’ and ‘Rocket surgery made easy’.
4. Maintaining information and structure

Someone should have an encyclopaedic knowledge of your website! There are various pieces of software that are available for maintaining links and keeping track of orphaned files.

- Linkscan is a very useful tool for keeping an eye on dead links, but is commercial and expensive (although they run a free quickcheck service - see http://www.elsop.com/linkscan/quickcheck.html). There are shareware or free alternatives that have less functionality (see http://validator.w3.org/checklink or http://linkchecker.sourceforge.net/). There is also a link scanning extension in Firefox (Link Evaluator) that you can use to scan all links on the page you are looking at - other browsers also have link scanning extensions.

- Google analytics (http://www.google.com/analytics) is an excellent free log analysis tool that measures traffic through your site. It depends on you adding some javascript to every page (which issues a cookie to users) so might not be appropriate unless you have an easy way of doing this. There is a handout from an introductory course on using Google Analytics at http://web-support.csx.cam.ac.uk/courses/.

- Analog was a local product for analysing your server log. It has been recently redeveloped and updated and the download can be found at http://camie.dyndns.org/technical/analog-camie-ed/. There are a number of commercial log and website analysis packages

- Web counters can be useful if you need statistics for particular pages - see Site Meter (http://www.sitemeter.com/default.asp) and Statcounter (http://www.statcounter.com/), which can be made invisible.

Keep a regular eye on your log files as they give important information about traffic, requests, failed requests and will help you spot anomalies and trends. Make sure you keep your privacy policy up-to-date with the tools you use - particularly any third-party tools (not on your server).

Re-organising your website

If you create a new structure for your website, it is up to you to redirect your users to the appropriate place in the new structure. Your server software may allow you to redirect traffic in a way that the user does not notice (but then you run the risk of them never noticing and never changing links). You can add a page that automatically redirects users but also reminds them to change the link, you can add a page that tells them the new location but doesn’t actually send them there, or (sometimes) you can tailor a 404 page to direct users to commonly sought pages. If you do not redirect users they are likely to lose faith in your site and go elsewhere.

I’m including a range of notes about indexing your site, which describe the process and focuses on the local University site-wide search - see http://www.ucs.cam.ac.uk/web-search/ for details.

Indexing your website: Controlling access intentionally

If you don’t want people to read files, then they shouldn’t be on the web server. ‘Spring clean’ files and remove all the information that is no longer pertinent on an annual basis.

There will be some directories that you do not want your indexer to look at and index. When using a spider or robot based indexer, controls over indexing are through a number of means and will be observed by Internet indexers such as Bing and Google, as well as your local indexer. Obviously, if you can take care of all your indexing requirements at once it will save you work in the long run.

These controls are:

- the robots.txt file
- robots metadata tag giving noindex and nofollow information (and combinations) in individual files

All ‘proper’ search engines will observe a robots.txt file and do what it says, and observe the robots metadata tag (see http://www.searchenginewatch.com/webmasters/article.php/2167891).

At another level, access to branches of a web server can be limited by the server software. Combining access control with use of metadata can give information to those within the access domain and some limited information to those outside.
robots.txt
Robots.txt sits at the root level of your web server and give information about what should not be indexed. An example of an instruction to any indexer (shown by the use of *) might look like this:

```
# A comment line just to show what one looks like; it is ignored.
User-agent: *
Disallow: /bin/
Disallow: /cgi/
Disallow: /includes/
Disallow: /tmp/
Disallow: /~
Disallow: /stats/
Disallow: /local.html
```

Another example, including a reference to a named search engine robot, as well as a general instruction, might look like this:

```
User-agent: Ultraseek (webmaster@ucs.cam.ac.uk)  # local search engine
Disallow: /bin/
Disallow: /cgi/
Disallow: /includes/
Disallow: /tmp/
Disallow: /~
Disallow: /stats/
Disallow: /local.html

# tell all others to go away
User-agent: *
Disallow: /
```

Robots meta tag
If the information providers can neither update the robots.txt file nor request changes to it, they can use robots META tag to specify within an HTML page whether indexing robots may index the contents of the document and/or follow links from it to other documents. This is of limited use, since it can only be used in HTML documents, but does not require changes to any robots.txt file. If there is also a robots.txt file, the exclusions there are processed first.

All META tags must be placed within the <HEAD> section of the HTML. The name attribute must be "robots", and the content attribute contains a comma-separated list of directives to control indexing, chosen from

- INDEX or NOINDEX - allow or exclude indexing of the containing HTML page.
- FOLLOW or NOFOLLOW - allow or exclude following links from the containing HTML page.
- ALL - allow all indexing (same as INDEX,FOLLOW)
- NONE - no indexing allowed (same as NOINDEX,NOFOLLOW)

The values of the name and content attributes are case-insensitive. Repeated or contradictory values should be avoided. The defaults are INDEX,FOLLOW, i.e. all indexing is allowed. Note that INDEX and/or FOLLOW cannot override exclusions specified in a robots.txt file, since an excluded document would not be fetched and the tag would not be seen. Also, the NOFOLLOW exclusion applies only to access through links on the page containing the tag - the target documents may still be indexed if the search engine finds links to them elsewhere.

Ignoring the "shorthand" ALL and NONE variants, the following examples show all the possible combinations:

```
<meta name="robots" content="index,follow">
<meta name="robots" content="noindex,follow">
<meta name="robots" content="index,nofollow">
<meta name="robots" content="noindex,nofollow">
```

Stopping access unintentionally
Depending on how your pages are generated, you may be excluding indexers without intending to, or knowing. Problems will arise with indexing of:

- Forgetting to change your robots.txt when you make your site public
- Graphics (including Flash and Shockwave) - use alt text
- Pages requiring passwords, registration, specific location or cookies
- Java applets
- Multimedia files

**Fixing index access problems**

**Graphics:** Alt text will be indexed by search engines, so by adding suitable alt text to all your meaningful graphic based information you will make possible that the info is indexed.

**Protected pages:** the local search engine may be able to index these pages but we would have to test to find what was possible. The internal search engine will index anything accessible within cam, but not content that is only accessible in your local domain.

**Acrobat files:** Give effective title and metadata so that search engine results are usefully labeled (see later)

**Aids to better indexing**

There are several things you can do to ensure the indexing of your files is as good as possible. The Google webmaster guidelines at http://www.google.com/support/webmasters/bin/answer.py?answer=35769 also have valuable advice and gives access to some new tools and a blog for keeping up with new information.

**Title**

Title is the most important piece of information for indexing purposes. There may be a dilution effect (keyword/ 6 words ranked higher than keyword/10 words), so keep the title short and succinct. Remember that this is what goes into bookmark lists and is the heading in search results, so it also needs to adequately reflect the content of the file.

**Structure your information**

Most search engines regard headings marked by h1 tags as being more relevant than other text. Some search engines will have a depth limit on searching a site - most will have anti-spamming measures that ensure a very large number of links on one page will not be indexed. If many such pages are present the site may be black-listed by the indexer.

**Adding metadata**

You can add information to your html files that will be indexed in addition to the content of the rest of the page. The standard model is for description and keywords - the description being used as the summary instead of the start of the file, and the keywords being picked up as search terms. You cannot depend upon the description and keywords metadata tags being.

For instance;

```html
<head>
  <title>Stamp Collecting World</title>
  <meta name="description" content="Everything you wanted to know about stamps, from prices to history." />
  <meta name="keywords" content="stamps, stamp collecting, stamp history, prices, stamps for sale" />
</head>
```

Dublin core is another standard for metadata, which in its simplest form consists of 15 terms (see http://webreference.com/xml/column24/index.html). No major web indexers use Dublin Core but specialist search engines do.

**Metadata and PDFs**
When you generate a pdf, information may be inserted into the description, keywords and title slots from the original file, unless you tell the processor otherwise. You can change these values in the pdf by using the full version of Acrobat (Document properties>Summary), but increasingly users generate their own pdfs and do not know that these values are there, or how to influence them before the pdf is generated.

**PDFs from Word**

By now there are many versions of Word about in the world and this advice does not apply to all of them. In addition to versions of Word being a difficulty, the process is handled differently by various means of producing an pdf file. In the main, the most information is passed from Word if a postscript file is distilled, rather than a pdf file written directly. Word may take the information from some fields in the 'Properties' information, and use it as metadata:

**PDFs from scans**

PDFs created from scans, either as a graphic, where no metadata will be present, or OCR, where the metadata will probably be faulty, need careful checking before being made available on the web.

**Changing metadata**

Edit the metadata using Acrobat and reindex the document (if you are able to) as soon as possible.

**Further info**


**Using the University site-wide search**

There is extensive information on the web about how you can use the University site-wide search engine - see [http://www.cam.ac.uk/cs/web-search/](http://www.cam.ac.uk/cs/web-search/). A synopsis of some important points follows here.

The University-wide search engine consists of two separate indexes - the internal one is what you see if you Raven authenticate.

**Adding a new web server**

If you are staring a new server the best idea is to contact web-support@ucs.cam.ac.uk to let us know - the web server could well be picked up automatically but there are other administrative functions that can be carried out if we are made aware.

**Adding the search facility to your server**

You can add a search facility onto any page of your web server, which could give a search of the whole University index or one or several parts of it, either by giving a configured link or a form. Full information is at [http://www.ucs.cam.ac.uk/web-search/searchforms](http://www.ucs.cam.ac.uk/web-search/searchforms)

**Looking at what is indexed**

The search facility servers give access to a list of what is indexed for your site, as well as being the place you can add urls to the index or reindex your site). Go to [http://int.web-search.cam.ac.uk/](http://int.web-search.cam.ac.uk/) or [http://ext.web-search.cam.ac.uk/](http://ext.web-search.cam.ac.uk/) and select ‘Help” and you will see this window:
You can select topics from the right menu (some of which may not be available, depending on whether you are connected to the CUDN). For instance, if you select View Sites and then select View site-by-site details, you will see this, which gives you raw information about the extent of the indexing (you will see your cam-only content included on the int. view, and not on the ext. view):
If you click the link to your website from the right-hand list, you will see a list of the urls that have been indexed (clicking on an individual page now will give you details of the search results and when the page was last indexed).

Revisiting a site

Back at the help pages, select Revisit site from right hand menu, to see this:
You can select an option out of those listed depending on your requirements.

**To ensure a new file in added to the index**
Back at the help pages, select **Add URL** from right hand menu, and type in the url you need adding.

**Further info**
- ‘Audiences, Outcomes, and Determining User Needs’ - A List Apart
  http://www.alistapart.com/articles/audiences-outcomes-and-determining-user-needs/

**Content management**
- How to evaluate a content management system -
- Demos of (some) open source products - http://opensourcecms.com/index.php

**More general information about search engines and search engine optimization**
- http://www.searchtools.com/
- Tutorial from Words in a row http://www.wordsinarow.com/seo.html
- Search tools chart - http://www.infopeople.org/search/chart
- Google webmaster tools - https://support.google.com/webmasters/ You can associate your site directly with the tools and it will give you indications where there are indexing problems.

**Where to go now**
- Optimal Web Design (designing for usability) - http://www.surl.org/
- For information about the local search engine and how you can use it, see
  http://www.ucs.cam.ac.uk/web-search/