

Designing Web Applications

Overall Design Issues

The first step to designing a Web application is to clearly define the goals: Why are you developing this application, what are the application's objectives, and how will they be measured? It is important to bring together the right set of folks to help create a common vision for the application. It is also important to establish the available budget for attaining your application's goals.

The next logical phase is to determine your customers' requirements. What's driving your customers, and what are their needs and goals? What are they looking for? What's the users' level of expertise? How will they use the application? What are their technical requirements? It is imperative to define usage scenarios; provide examples and carefully describe how, where, when, and why a particular user will use the application. Be sure to prioritize these scenarios so that your most important users come first.

You must also establish usability requirements. How will you determine user satisfaction for the scenarios created? Be sure to capture data that relates to understandability, knowing what will happen when you push a button, and transparency - being able to quickly access the information or functions they need. You should also capture data that relates to your application's visual appeal.

And now you must consider the likely possibility that someone with a disability may be using your application.

User Centered Design References

- Universal Web Design by Crystal Waters
- Designing Web Usability: The Practice of Simplicity by Jakob Nielsen, New Riders Publishing, Indianapolis, 2000. ISBN 1-56205-810-X.
- Cost Effective User-Centred Design <http://www.usability.serco.com/trump/methods/webdesign.htm>
- Handbook of User Centred Design <http://www.ejeisa.com/nectar/inuse/6.2/contents.htm>
- Principles of Universal Design http://www.design.ncsu.edu:8120/cud/univ_design/principles/udprinciples.htm

Design Principles

A primary design principle should be “Function over Flash”. Design decisions should always favor functionality and usability.

Make things obvious to users. Studies have shown that people do not recognize many images and icons as links to other information or functions. So make these navigation options obvious. The more the links look like buttons the better. A user should not have to move the cursor to hunt for links.

Be consistent when designing:

- Placement of buttons
- Wording in messages
- Color scheme
- Click actions

REFERENCES:

Human Factors International <http://www.humanfactors.com/>

Web Usability Illustrated: <http://www.humanfactors.com/library/ecusability.asp>

A systematic, scientific approach to e-commerce design uses human factors or ergonomic principles to minimize the visual, intellectual, mental, and physical "effort" users exert.

Technology Guidelines

- Sun Java applets [Java Look and Feel Guidelines](#)
- Microsoft Active X controls [Microsoft Windows User Experience](#)

Design Choices

Color

Color and Highlighting <http://www.humanfactors.com/color/default.asp>

- Use sparingly
- Use to aid scanning
- Color can speed search of the screen.
- Use no more than six different color codes on a screen.
- Use as a code
- Do not rely solely on color to convey information (such as required fields or errors).
- Follow the population stereotypes. Color can have meaning; in fact, color already has meaning you can use.
- Use to draw attention
- Use to show relationships
- Avoid contrast deficiency
You need a 90% contrast difference between foreground and background in order to read text clearly.
- Avoid chromatic aberration
Do not create text in pure red or blue. Due to chromatic aberration, red and blue do not focus well on the retina (they look fuzzy).
- Avoid chromostereopsis
Do not show pure red and blue together.
- Accommodate colorblind users
 - 9% of men are color weak.
 - 2% of women are color weak.
 - Do not make users rely solely on color to use the system.
 - Everyone is colorblind in low light.
 - You can't depend on the user's ability to see colors. If red vs. green is the only way to tell which fields are in error, about 6% of all users will have trouble telling the difference.

Wording

- Use full words and sentences, as opposed to abbreviations such as “dt” for date or “addr” for address
- Capitalize labels consistently
- Avoid redundant labels (e.g. “Employee Name” when the user knows they are on an employee information screen)
- Word error messages positively to imply that the user is in control and provide insight into how to use the application correctly (“Your entry is wrong” vs. “A phone number should be 10 digits in length”)
- Word error or confirmation messages consistently and display messages in a consistent place
- Omit needless words
- Use simple words
- Use short sentences to increase reading speed and comprehension
- Be specific and concise
- Avoid jargon
- Use words consistently (e.g. Always “Cancel” or always “Escape”)
- Avoid acronyms unless all users will be familiar with the term without ambiguity
- Avoid computer jargon
- Write in an active voice
- Make sure tone is correct for user/task/environment (kids vs adults)
- Avoid jokes
- Beware of anthropomorphism: making the computer pretend to be a person.

Layout

- Match eye movement: scan from top to bottom, left to right
- Left justify
 - Left-justified text, labels, and data are most common, and physically easier to scan than right-justified. The one exception is numbers that are being compared. These should be right-justified on decimal. Avoid right-justification of labels, or other data fields.
- Use sufficiently large icons and buttons to speed traversing and selecting
- Avoid clutter (difficult to find fields; intimidates novices) unless window proliferation is too costly
- Group fields by function
- Break tables lines into groups by leaving every 5th or 6th line blank
- Choose the right sequence:
 - Alphabetical
 - Frequency of use
 - Most Important
 - Logic or task flow
- Do not put text in all Capital Letters
 - Short titles and acronyms are OK. But using all capital letters slows reading speed by 14-20%
- Avoid horizontal scrolling of entry fields (unless scrolling capability is clearly marked)

Article that includes examples to improve field layout design:

Designing Web-Based User Interfaces by Scott W. Ambler

Software Development March 2001 <http://www.sdmagazine.com/articles/2001/0103/0103n/0103n.htm>

Widgets

- Chose an appropriate widget (radio buttons vs drop-down list box)
- Follow standards for your organization

Navigation

- Match flow of pages to flow of work
- Allow for different approaches
- Follow cultural reading patterns (e.g. top to bottom, left to right)

Provide Ways to Recover from Mistakes

Accessibility

“See” the difference between a web page designed for accessibility by blind users versus one that is not well designed: <http://www.humanfactors.com/accessibility/chocolateaudio.asp> is an audio comparison of inaccessible and accessible Web pages.

Multiple Columns <http://www.humanfactors.com/accessibility/columns.asp>

Any Web page using tables to position text and other elements may be subject to information being presented in an unexpected, confusing sequence. Therefore it is important to make sure that the data in a table linearizes correctly if the table structure is not rendered by the user's browser. Any text or other elements, presented in a table that is formatted with multiple columns and has more than one row, may lead to sequence problems.

Font Size <http://www.humanfactors.com/accessibility/fontsize.asp>

For maximum accessibility, font sizes should be coded in relative sizes rather than fixed or absolute sizes so the user can adjust browser preferences and get a page to display with a larger font. This may cause some appearance problems with Web pages designed to a fixed grid, but the user with vision problems may not be able to read the page without some kind of text enlargement. Even normal youthful users who require optical correction for near-sightedness may have trouble because such correction makes objects appear smaller than with uncorrected vision. Some browsers, such as Opera, allow a page to be enlarged up to ten times without changing font settings. Other browsers, such as Netscape and Internet Explorer have quick settings that can bump up the font a couple of sizes, leaving the rest of the elements on the page, such as images, unchanged.

Use HTML Markup Correctly <http://www.humanfactors.com/accessibility/markup.asp>

HTML 4.01 has a number of features related directly to accessibility, but the most important item is to use HTML markup correctly. For example, HTML has six levels of headings - think of them as outline headings - that can delineate the structure of a long document. Some browsers allow the user to generate a table of contents from those headings to give an outline of the content of the page. If the Web page developer (or the software the developer employs) makes headings based on appearance alone, such as by using paragraphs with the HTML font element to produce visible structures that look like headings, then the outline of the page content displays nothing.

Jakob Nielsen's Alertbox for October 1996:

Accessible Design for Users With Disabilities <http://www.useit.com/alertbox/9610.html>

- Textual pages are reasonably easy to access for blind users since the text can be fed to a screen reader. Long pages are problematic since it is harder for a blind user to scan for interesting parts than it is for a sighted user. In order to facilitate scanning it is recommended to emphasize the structure of the page by proper HTML markup: use <H1> for the highest level heading, <H2> for the main parts of the information within the <H1>s, and <H3> and lower levels for even finer divisions of the information. By doing so, the blind user can get an overview of the structure of a page by having the <H1>s and <H2>s read aloud and can quickly skip an uninteresting section by instructing the screen reader to jump to the next lower-level heading.
- Use ALT to provide utility descriptions that convey the meaning or role of images.
- All imagemaps should be client-side and should use ALT attributes for each of the link options so that a user who cannot see the image can have descriptions of the destination read as he or she moves the cursor around.
- Never encode information with absolute font sizes but use relative sizes instead. For example, when using Style Sheets, do not set the font-size attribute to a number of points or pixels but set it to a percentage of the default font size.
- It is recommended to test pages with the default font set to 10, 12, and 14 point to ensure that the design is optimal for these common font sizes and then to make additional checks with default fonts of 18 and 24 points to make sure that the design still works at these accessibility-enhancing sizes.
- Transcripts should be made available of spoken audio clips and videos should be made available in versions with subtitles (which will also benefit users who are not native speakers of the language used in the video).
- Do not design imagemaps that require extremely precise mouse positioning. Client-side imagemaps will work even for users who cannot use a mouse at all: the browser should be able to move through the links under keyboard control.
- People vary in their spatial reasoning skills and in their short-term memory capacity. Programmers and graphic designers tend to get uncommonly high scores on tests of spatial reasoning skills and are therefore good at visualizing the structure of a Web site. It is safe to assume that most users will have significantly greater difficulty navigating a Web site than its designers have. Simplified navigation helps all users, but is a required enabler for users at the opposite extreme of the scales. People who have difficulty visualizing the structure of information can be helped if the site designers have produced such a visualization for them in the form of a sitemap; they would be further aided if the browser updated the display of the sitemap with the path of the navigation and the location of the current page.
- Users with dyslexia may have problems reading long pages and will be helped if the design facilitates scanability by proper use of headings. Selecting words with high information content as hypertext anchors will help these users, as well as blind users, scan for interesting links (no "click here", please).

- Most search user interfaces require the user to type in keywords as search terms. Users with spelling disabilities (and foreign-language users) will obviously often fail to find what they need as long as perfect spellings are required. A first suggestion is to for search engines to include a spelling checker; other ideas from advanced information retrieval like query-by-example and similarity search can also help these users (and benefit everybody else at the same time).

REFERENCES

American Federation of the Blind: Designing an Accessible Web Site
http://www.afb.org/info_document_view.asp?documentid=1449

References for Web Accessibility for Mainstream Developers <http://www.w3.org/WAI/References/#mainstreamdev>

How to Build Accessible Web Pages in Dreamweaver <http://spingree.cals.wisc.edu/design/accessibility.htm>

- Useful information for any web page developer

Accessibility info for software and the Web <http://www-3.ibm.com/able/guidelines.htm>

IBM provides guidelines and solutions that can help you in developing an accessible IT infrastructure, including e-mail systems, business applications, and Web sites. Check out the newly added techniques for the IBM accessibility guidelines.

IBM's checklist for Web Application Accessibility <http://www-3.ibm.com/able/accessweb.html>

Each checkpoint includes rationale, techniques and testing strategies.

IBM's checklist for Java 2 applications <http://www-3.ibm.com/able/accessjava.html>

Each checkpoint includes rationale, techniques and testing strategies.

Microsoft Accessibility Technology for All <http://www.microsoft.com/enable/>

Sun Microsystem's Accessibility Program <http://www.sun.com/access/index.html>