

Site Development Testing Suggestions

Basics to Testing Web Interfaces

Pierre Berryer

Original Date: April 2002
Version 1.01

Site Development Testing Suggestions

1. Cross Browser Testing
 - a. Recommended Test Browsers
2. Tracking Browser Test Results and Repairs
 - a. Focused Approach Method
 - b. Implementing Changes & Work Distribution
 - c. Identifying Irreparable Bugs
3. HTML Interface Development Tips
 - a. General HTML Suggestions
 - b. Netscape Problems

Cross Browser Testing

An organized system for testing needs to be tailored to the project. Identify key points in the contract that will be used as markers for adherence to cross browser compliancy. This means deciding on how close the application running in one browser must resemble and perform to an instance in another browser.

If the site is an Intranet with control over the user's browser, testing is much simpler. Still, the methods discussed here will help in testing as well as maintaining coding standards.

Obviously, if the application fails to display necessary information in Netscape, that is a critical issue. But what if a client side timeout screen doesn't work? Is a server side timeout adequate? Or what if the navigation icons have white space between them or are unevenly displayed? During testing, all types bugs should can be identified. This may include many that may be deemed insignificant, such as pixel differences in the alignment of icons from one browser to another. However, if the client requires icons to be displayed in the same location on each application screen, then larger alignment differences are a valid problem.

Visiting an example statistics page such as those on Hitbox can help determine browser usage statistics, however, they should be compared to other sources. Here is an example statistics link from 2001:

<http://vwr1.hitbox.com/HitboxUI?p=1&acct=WQ500208NHSW&pd=4&y=2001&m=12&t=420&cat=2&hp=1&dt=2001>

The following is a recommended list of browsers to test on:

1. Internet Explorer 6.0 - 5.5 - 5.0
 - a. These represent the major evolutionary steps in IE. With each new version, many users and corporations stayed with the previous one, waiting for the latest technology to "settle" before making the often huge download. This leaves the three browsers split roughly into thirds. IE 6.0 comes standard with Windows XP and will, over time become the leader.

Unfortunately, the 5.x series has some subtle and strange bugs that warrant testing both 5.0 and 5.5. For instance, some quirks with JavaScript can result in a completely blank page in IE 5.0.

2. Netscape 6.0/6.1/6.2.2
 - a. This series is the Netscape browser for the future. Version 6.0 has many “Irreparable” bugs, which make focusing on it a burden. It still claims a small number of users, partly due to its major bugs when originally released.
3. Netscape 4.7 – 4.0
 - a. The Netscape 4.5 and above series made some major advances. Netscape 4.7x is the most widely used of the 4.x series with Netscape 4.0 & 6.0 holding similarly sized audiences. Netscape 4.0 is the browser that offered decent support for CSS and plug-ins, so it is still around, but definitely old technology.
4. AOL 7.0 (or older versions if available)
 - a. AOL is a different browser from IE even though they share a common code base. AOL also suffers from strange bugs, such as blank pages or even browser crashes. Older versions can be tested if available.

Tracking Browser Test Results & Repairs

For applications with many pages and/or identified bugs, a spreadsheet can help keep track of the issues in the absence of bug tracking software such as Track Record. The following is an example of one browser’s trouble sheet. For a more detailed sample from another project, see the file *Site Example - Browser Compatibility Status 1.2.xls*.

Netscape 6.0 Problems

Tested by Anil Sharma

Page Name	Priority Rate 1 to 4	Problem
Common =	1 2	Does not allows user to navigate by using the SpaceBar Header - Alignment problem
Index.htm	1 2	Does not allows user to navigate by using the SpaceBar Header - Alignment problem
1stQuestion.jsp	2	Enter key allows user to go to next page
2ndQuestion.jsp	2	Enter key doesn't allows user to go to next page
4thQuestion.jsp	1	Something wrong with the TextBox and RadioButton option
Help2_English.jsp	3	The header seems to be of wrong size
Help4_English.jsp	3	The header seems to be of wrong size
ScreeningResults-english.jsp	2	Header - Alignment problem between Print and Help icons
ScreeningResults-english.jsp	?	+ ' is inactive
Survey_Final_english1.jsp		2nd set of Question - Did you have to ask someone to help you use STARS?

Focused Approach Method

The compatibility problems found among the multiple browsers should be listed and prioritized according to their impact on the site. They should be rated along multiple levels. Here is an example, illustrating the difference between four levels.

Triage Key

Level 1 consists of immediate functional issues. These can be identified as preventing a page or page element from working, as well as usability problems with the interface severe enough to warrant immediate repair.

Level 2 consists of obvious cosmetic problems. Issues such as misalignment of page elements fall under this category.

Level 3 consists of cosmetic errors that result in a less attractive interface. The primary reason to deal with these is to maintain a professional web site appearance. Issues such as inconsistent line spacing and improperly sized icons (aliased, choppy or “stair-stepped” edges) fall under this category.

Level 4 consists of purely cosmetic observations that may not be worth the effort of repairing them. Some of them may fall under the category of general recommendations for the design and not browser compatibility repairs.

Implementing Changes & Work Distribution

In addressing repairs, the most critical issues are given priority. Of these, the issues common to multiple pages are scheduled for repair first. In a team environment, a first round of repair tasks is distributed among the team after discussion of each level 1 item's feasibility and determining which members are most knowledgeable about any particular repairs.

The workflow can be divided multiple ways. One method goes along the lines of bug fixes. Each team member is assigned a class of problems. Under a tight schedule, this can help alleviate the need to teach each member how to repair multiple types of problems, letting them focus on one type of problem and become comfortable with repairing it quickly and consistently. This method requires the team lead to track the progress and sign off on the pages when each team member's changes have been made.

Using another method, repairs can be divided among the team by page groups. This would allow members to work without needing to wait for another member to check in a common file, as well as allow the individual team members to sign off on “their” pages as complete. In light of a tight deadline, however, this added benefit may not counterbalance the increased time for knowledge transfer that may be required.

Identifying Irreparable Bugs

With the state of software as it is, there is always the chance of encountering bugs that are not repairable and sometimes cannot be worked around. When something particularly strange comes up, research needs to be done to determine if it falls into this category. For example, some versions of AOL browsers do not correctly identify themselves to browser detector methods. Another example is that Netscape 6.0 has problems tabbing through a page's links. Issues that cannot be repaired may need to be then documented with references to reputable sources such as the browser developer or established Web Developer forums.