

HTML Checklist

Any exception to W3C HTML v4.0 standards have been approved and documented
HTML code is W3C HTML v4.0 compliant (barring any approved exceptions)
Web page renders correctly when viewed with Opera 5.0, Internet Explorer, and Firefox browsers
Comments and change control logs are not included in the HTML sent to the client

Image Checklist

This image adds value to this Website
This image is stored in the most appropriate format (eg GIF files for buttons and JPG files for photos)
If a GIF file, the image size is a multiple of 8 pixels
The visual size of the image is appropriate for the size of the viewable screen (ie it does not occupy too much or too little screen real estate).
The physical file size of this image is as small as possible without compromising the quality of the image (ie the file was saved using the optimum compression ration)
An appropriate ALT tag is included with this image
The WIDTH and HEIGHT (expressed as page % and not absolute pixel sizes) tags have been specified for this image
Notre Dame has a legal right to use this image on a Web site 9ie it is not trademarked or copyrighted by someone else, unless they have granted Notre Dame explicit permission to use this image)

Image Validation

The total size of all of the images on this web page does not excel 50 Kbytes
There is not more than one animated image on this web page
Photographic images aside, (eg JPG images) there are not more than 256 total colors used on this web page
Any image maps used are client side (as opposed to server side)

Font Validation

The font is proportional
The primary font is Verdana, with Arial and Sans-Serif specified as alternatives
The browser's base font is not altered
Only relative font sizes (small, medium, and large) are used, rather that specific point sizes
No more than three font sizes are used on the web page
Symbol fonts are used only when absolutely necessary
If symbol fonts are used, they are properly mapped to the "private use area" of the developer's Unicode
Browser default colors are not overridden

Printer-Friendly Validation

The test on the Web page is formatted correctly when printed via a 72-dpi printer using Letter and A4 paper sizes

The content of the webpage is clearly readable when printed with a black and white printer

The content of the web page is clearly readable when printed with a color printer

The background color of the web page is white

Only dark colors are used for the next on the web page

Style Sheet Checklist

The style sheet is W3C CSS Level 1 compliant

The style sheet is correctly interpreted by the 4.x generation of web browsers

The style sheet complies with printer friendly standards

The style sheet is defined as an external CSS file

Web pages do not modify the style sheet dynamically

Web pages that use the style sheet provide acceptable rendering when viewed by browsers that do not support CSS or have CSS turned off by the client

Table validation

There are no unwanted spaces or carriage returns in the table

No cell is overpopulated with too much verbiage

Every cell in the table is populated (ie no null values) as some browsers collapse empty cells. Extra scrutiny should be applied if the information is imported from a database dynamically

The WIDTH and HEIGHT tags were specified for all cells (using screen % instead of absolute pixels wherever possible).

Developers placed the majority of the page content inside the table and left paid advertisements (and other information that needs to load first) on the outside

Style Guide and Template Validation

The web page follows (except where documented/approved) the style guidelines documented in the Notre Dame website style guide

The web page was based on the most appropriate web page template

Plug-In Testing checklist

The web site lists the plug-ins (and versions) that are needed to view all of the content on the site

The web site (after requesting the client's permissions) is able to detect whether or not the required plug-ins (and correct version) are installed on the client side

In the event that the web site is unable to accurately determine whether or not a plug-in is installed, the web site contains an area that tells users how to proceed.

Installation checklist

Different versions of the same brand of browser are installed in different instances of an operating system.

Only general-release software is used. No OEM, SP, or Beta versions are used, with the exception of any required Y2K patches that are necessary in order to make the product work post Y2K

All of the installations use the installation defaults for directory names, cache sizes, fonts, plug-ins, etc.

Instability and Serviceability Checklist

Product documentation explains the exact order in which the components should be installed and the configuration settings that are required or recommended

Product documentation explains how to uninstall the product cleanly

Product documentation adequately describes when and how the data files or database should be reorganized

Automatic updates install and operate correctly on all of the supported platforms

Automatic updates install and operate correctly when other applications have been added/removed before and after the update is performed

Redirect Testing Checklist

The default 400, 401, 402, 403, and 404-error pages have been developed and properly configured on the production web servers

If this link is being redirected, it goes to the correct final destination and is not re-redirected

If a link points to a directory (instead of a specific web page), the link ends with a slash

Bookmark testing checklist

Every web page has a bookmark that accurately reflects the content of the page

No bookmark is longer than 32 characters, since browsers typically truncate the display of verbose descriptions

Each bookmark must start with “vEOC - ...”

Frameset Testing checklist

Pages using framesets are displayed correctly

Frames are not resizable

Page within the frameset can be bookmarked

The back button recalls the URL of the last frame viewed

The initial frameset is downloaded in an acceptable period of time

Pages using framesets can be printed correctly or an alternative page is available for printing

Nested framesets (if used) have sufficient screen real estate assigned to each frame

All external links land new browser windows (ie third party web sites are not embedded inside the vEOC frameset)

Search engines can find all of the content within the framesets

Site organization testing checklist

“Core” web pages can be located within four clicks.

All web pages on the web site can be found by casually browsing the web site (ie no need to resort to a site map or search engine)

Information on the web site can be found by using the search strategies that a vEOC visitor might consider (eg stock ticker symbols, company names, and vEOC product names)

The web site does not contain any orphaned files (ie files that cannot be reached by following any path from the home page)
the web site uses the same or fewer pages than their competitors in order to accomplish the same tasks

web site map testing checklist

all "core" web pages can be found using the site map
only "core" web pages are listed on the site map
web pages are listed in an appropriate order/hierarchy
links are all functional and go to the correct pages

Internet search engine testing checklist

The first set of results is returned within 5 seconds (excluding internet transmission times)
The result set is sorted appropriately (eg alphabetically or by % likelihood)
The search engine functions correctly when a user enters common words that are likely to generate a huge number of hits such as "the," "a," or "veoc"
The search engine functions correctly when a user enters non-existent words that are unlikely to generate any valid answers such as "sdhjhsf," or "djfks," or null requests
The search engine ignores the source code used to build a web page and only indexes the content of the web page (eg requesting information on "JavaScript" will only return documents that reference javascript, not all of the web pages that use javascript in their source code)
The search engine does not index sensitive words such as "secret," or "fraud," or competitor's names
The search engine functions correctly when you enter a search string with the maximum number of characters and a word with the maximum number of characters plus one
The search engine functions correctly when you enter multiple word requests, with and without the Boolean operators "and," "or," "not," "+," and "-."
The search engine functions correctly when you enter one or more wildcards
If fuzzy logic is enabled, the search engine offers alternate suggestions for zero hit requests based on searches using a spell-checked version of the initial search string.

Tool evaluation checklist

External links can be checked but (optionally) not scanned any further. No need to check the internal links of another organization's web site
When encountering a recursive loop, the tool does not go into a "death-spiral"
The tool does not ignore duplicate links
The tool is able to handle dynamic links
The tool is able to handle framesets.
The tool is able to handle cookies (session and/or persistent)
The tool can handle pages that require user input (eg forms)
The tool facilitates identifying suspiciously small or large pages
The tool facilitates identifying absolute links
The tool facilitates identifying pages that are too deep

User Interaction

Form testing checklist

All data entry fields have the HTML SIZE attribute set correctly (SIZE is used to specify the width of the field).

All data entry fields have the HTML MAXLENGTH attribute set correctly (MAXLENGTH is used to specify the maximum number of characters a user can enter)

If radio controls are used, a default is always selected

All required fields use a visual cue to indicate to the user that the field is mandatory

If a form uses a drop-down data entry field (control), the options are sorted appropriately and the field is wide enough to display all of the options.

Data is not lost when the user clicks the browser's back (and subsequently Forward) buttons midway through a series of forms

Data is not lost when the user clicks the browser's Forward (and subsequently Back) buttons midway through a series of forms

Data is not lost when the user clicks the Go//History buttons to revisit previous forms

Validating Forms on a Web Site

Data is not lost when the user clicks the Bookmark/Favorite buttons midway through a series of forms

Data is not lost when the user clicks the browser's reload button midway through a series of forms

Data is not lost when the user resizes the browser window

Duplicate data is not added to the database when a user presses any combination of Forward, Back, Go/History, Bookmark/Favorite, Reload, or Resize buttons midway through a series of forms

The browser places the cursor on the most appropriate field/control when the form is first viewed

Using the browser's tab key allows the client to tab through the input fields on the form in a top-to-bottom, left-to-right order.

If the form's data is sent back to the web server using the HTTP GET command, the data is not truncated

Form validation checklist

All data entry fields are checked for invalid data. An appropriate error message is displayed if the data is found to be invalid

All validations are performed (and error messages displayed) in a top-down, left-to-right fashion

All required fields are checked on the client-side

Wherever possible, all field co-dependencies are checked on the client-side. If, for example, the user places a trade that is a "limit" order, the system checks that a value has been entered for the price. Otherwise, this field should be blank.

All basic data formatting checks are performed on the client-side

All client-side checks are re-checked on the server-side

DHTML testing checklist

DHTML is appropriate for this web site because the majority of visitors use MS-IE 4.0_ or Netscape 4.0+.

All of the DHTML code used in the web site conforms to the W3C DHTML standard
Pages that contain DHTML code are displayed correctly when viewed through MS-IE 4.0 or higher

Pages that contain DHTML code are displayed correctly when viewed through Netscape 4.0 or higher

Popup window testing checklist

The web site is able to detect a browser that has disabled (or does not support) javascript/java/activex and provides the user with an appropriate message

The pop-up follows the rest of the web site's GUI standard

The pop-up is not too large for the parent window and its initial screen positioning is appropriate.

Streaming content checklist

Quotes are up-to-date and are no more than 1 minute old, 99% of the time

The streaming quote server and network is able to handle the expected demand for this service

Clients are able to suspend/restart this service without needing to unsubscribe/re-subscribe

Clients are able to adjust the frequency of the updates to cater to different client-side bandwidths

CGI script testing checklist

The CGI script is able to parse input parameters containing quotation marks, carriage returns, ampersand symbols, hash symbols, dollar signs, question marks, and other control characters

The CGI script is robust enough to handle missing and/or out-of-range input parameters

CGI script validation

The CGI script is robust enough to handle null values being returned from the database

The CGI script is robust enough to handle a "no record found" code being returned by the database

The CGI script is robust enough to handle a "duplicate record inserted" code being returned by the database

The CGI script is robust enough to handle multiple records being returned by the database

The CGI script is robust enough to handle a database timeout code being returned by the database

The web server has sufficient resources to handle the expected number of CGI scripts that are likely to be initiated.

Data integrity validation

A new record is accurately inserted into the database

The record can be accurately read from the database

The record is accurately updated in the database

The record id completely deleted from the database

Server Side Includes Testing checklist

All SSI and XSSSI selection criteria are accurately documented, and each "include" file contains a "start of file" and "end of file" comment

No JSSI files are used

The appropriate content is displayed and formatted correctly for each of the possible selection criteria

No "include" file references another "include" file. While technically possible, this programming style can be difficult to debug and can also impact performance.

Dynamic Server page checklist

The dynamically generated web page is not a candidate for being replaced by one or more static pages

Developers used a single language for all scripts within each dynamically generated web page

No "template" file references another "template" file. While technically possible, this programming style can be difficult to debug and can also impact performance

All DSP templates have been inspected by at least one senior developer who was not the author of the template

All high-frequency web pages have been generated and manually tested

All high-risk web pages have been generated and manually tested

Shopping cart validation

If a user adds more than the maximum number of items that the cart can hold, an appropriate message is displayed

If a user tries to check out an empty cart, an appropriate message is displayed

If a user removes an item from the cart, the web application correctly removes the item and displays an appropriate message

If a user completes a purchase (thereby emptying the cart), but then uses a Back/Go/History button to return to a pre-checkout page, the web application ignores any attempt to alter the contents of the check-out cart and displays an appropriate message

If a user abandons the cart and returns to the web site at a later date (within a specified period), the web application recalls the user's selections, resets the elapsed time, and allows the user to continue shopping.

If a user abandons the cart and doesn't return to the web site within a specified period of time, the web application releases and recycles system resources.

If a user launches two instances of the same browser and adds items to the cart using both browsers, the web application correctly updates the contents of the shopping cart(s)

Credit card testing checklist

If a user enters an invalid credit card number or expiration date, an appropriate message is displayed

If a user enters an invalid contact information, an appropriate message is displayed

Cancelled or old credit cards are rejected

Customers with insufficient funds are informed of the situation via an appropriate message.

Credit card numbers (and other confidential information) are stored in an encrypted format

Credit card information is stored in a separate database/file on a different physical server than other parts of a customer's profile.

This reduces the risk and likelihood that a hacker (internal or external) would be able to utilize this information

Payments are properly received for completed transactions

Tax collection processing is in place to calculate and collect all appropriate taxes (eg Federal, State, County, and other municipalities)

Fraud detection and credit card address verification services are functioning as specified

Legacy system testing checklist

The security of the legacy system is not compromised by its link to the outside world via the web server(s). That is, the web servers are granted the minimum required security levels

The legacy systems are able to handle the increase workload caused by the addition of the web site. This should be confirmed during a combination of peak web site activity and peak legacy system activity, which may follow a different business cycle such as end-of-month and/or quarterly processing

Message between the legacy systems and the web site are not corrupted or lost (ie adequate buffers exist)

Taking the legacy system off-line does not hinder the web site/application

Taking the web site off-line does not hinder the legacy system

Cookie testing checklist

Cookie validation

When cookies are:

Disabled before accessing the site, either one of two things happens:

The site works correctly (albeit more slowly)

The site issues a warning message telling the visitor that the site can only be accessed with cookies turned on

Disabled midway through a transaction, the site is able to detect the situation and handle it gracefully

Deleted midway through a transaction, the site is able to detect the situation and handle it gracefully

When the cookie is edited and some parameters are:

Added, the site detects the situation and handles it gracefully

Deleted, the site detects the situation and handles it gracefully

Swapped, the site detects the situation and handles it gracefully

Set to null, the site detects the situation and handles it gracefully

Some parameters are edited and set to invalid values, the site detects the situation and handles it gracefully

Other validation tests include the following:

When the client PC's memory and/or disk cache is cleared midway through a transaction, the site is able to detect the situation and handle it gracefully. Session cookies are stored

in memory and typically don't get saved to the hard disk. Persistent cookies may need to be deleted manually
when control characters or special operating system commands are added to a cookie, the site detects the situation and handles it gracefully
when multiple entries for the web site are added to Netscape's cookies.txt file, the site detects the situation and handles it gracefully
when the user identification field in the cookie is changed midway through a transaction, the site detects the situation and handles it gracefully. Consider replacing the regular user-id account with values such as Admin, Test, Superuser, or Guest

Session Testing Checklist

Maintaining a session

The web application is capable of maintaining a single session through multiple browsers running on the same client

The web application is capable of simultaneously accessing the same account through multiple clients

Adequate database locking strategies have been documented in the specification and have been properly implemented

The web application time/date stamps transaction using the clock on the web server, not the clock on the client

The web application is able to handle a user disabling cookies (session and/or persistent) midway through a session

The web application is able to handle a user clearing the cache (disk or memory) midway through a session

The web application is able to handle a user disabling javascript and/or VBScript midway through a session

The web application is able to handle a user disabling java applets

The web application is able to handle a user deleting the query portion of the web site's URL midway through a session

The load balancer (if used) is able to maintain a session

Concurrent user's testing checklist

Server memory is freed when a user completes a session or a transaction

Network connections are properly closed when a user completes a session or a transaction

Software licenses are freed and made available to other users when a user completes a session or a transaction

State transition diagram testing checklist

All possible wait states and normal events have been included in the state-transition diagram

All possible unexpected events have been included in the state-transition diagram (eg the client utilizes bookmarked or cached pages that would normally occur later in the transaction, premature termination such as when the server waits for a trade confirmation that never arrives, etc.)

If non-occurrence of a particular event is significant in the completion of another event, the state-transition diagram indicates the proper error routine. If, for example, a trade is not executed on the stock market, final confirmation should not be sent.

No other transactions affect the interaction between any two consecutive wait states shown on the diagram. If, for example, the NYSE suspends trading of a stock midway through a sell order, is the sell order still processed?

There are no redundant wait states or wait states that don't have an exit

All of the wait states can be reached (ie every wait state has at least one entry event).

Transactions do not contain sub-transactions that would be better represented on their own diagrams.

P183 Usability testing

Security

Good documentation

MVC model is used

Audio used has quality

All licenses on all software is documented