

The WebSite Quality Challenge

Edward Miller
Software Research, Inc.
901 Minnesota Street
San Francisco, CA 94107 USA

© Copyright 2000 by Software Research, Inc.

Email comments to miller@soft.com
See also the companion papers:
[WebSite Testing.](#)
[WebSite Loading and Capacity Analysis.](#)

ABSTRACT

Because of its possible instant worldwide audience a WebSite's quality and reliability are crucial. The very special nature of the WWW and WebSites pose unique software testing challenges. Webmasters, WWW applications developers, and WebSite quality assurance managers need tools and methods that can match up to the new needs. Mechanized testing via special purpose WWW testing software offers the potential to meet these challenges.

INTRODUCTION

WebSites are something entirely new in the world of software quality! Within minutes of going live, a WWW application can have many thousands more users than a conventional, non-WWW application. The immediacy of the WWW creates an immediate expectation of quality and rapid application delivery, but the technical complexities of a WebSite and variances in the browser make testing and quality control more difficult, and in some ways, more subtle. Automated testing of WebSites is both an opportunity and a challenge.

DEFINING WEBSITE QUALITY & RELIABILITY

A WebSite is like any piece of software: no single, all-inclusive quality measure applies, and even multiple quality metrics may not apply. Yet, verifying user-critical impressions of "quality" and "reliability" take on new importance.

Dimensions of Quality. There are many dimensions of quality, and each measure will pertain to a particular WebSite in varying degrees. Here are some of them:

- *Time:* WebSites change often and rapidly? How much has a WebSite changed since the last upgrade? How do you highlight the parts that have changed?
- *Structural:* How well do all of the parts of the WebSite hold together. Are all links inside and outside the WebSite working? Do all of the images work? Are there parts of the WebSite that are not connected?
- *Content:* Does the content of critical pages match what is supposed to be there? Do key phrases exist continually in highly-changeable pages? Do critical pages maintain quality content from version to version? What about dynamically generated HTML pages?
- *Accuracy and Consistency:* Are today's copies of the pages downloaded the same as yesterday's? Close enough? Is the data presented accurate enough? How do you know?
- *Response Time and Latency:* Does the WebSite server respond to a browser request within certain parameters? In an E-commerce context, how is the end to end response time after a `SUBMIT`? Are there parts of a site that are so slow the user declines to continue working on it?
- *Performance:* Is the Browser-Web-WebSite-Web-Browser connection quick enough? How does the performance vary by time of day, by load and usage? Is performance adequate for E-commerce applications? Taking 10 minutes to respond to an E-commerce purchase is clearly not acceptable!

Impact of Quality. Quality is in the mind of the user. A poor-quality WebSite, one with many broken pages and faulty images, with Cgi-Bin error messages, etc. may cost in poor customer relations, lost corporate image, and even in lost revenue. Very complex WebSites can sometimes overload the user.

The combination of WebSite complexity and low quality is potentially lethal to an E-commerce operation. Unhappy users will quickly depart for a different site! And they won't leave with any good impressions.

WEBSITE ARCHITECTURE

A WebSite can be complex, and that complexity -- which is what provides the power, of course -- can be an impediment in assuring WebSite Quality. Add in the possibilities of multiple authors, very-rapid updates and changes, and the problem compounds.

Here are the major parts of WebSites as seen from a Quality perspective.

Browser. The browser is the viewer of a WebSite and there are so many different browsers and browser options that a well-done WebSite is probably designed to look good on as many browsers as possible. This imposes a kind of *de facto* standard: the WebSite must use only those constructs that work with the *majority* of browsers. But this still leaves room for a lot of creativity, and a range of technical difficulties.

Display Technologies. What you see in your browser is actually composed from many sources:

- *HTML.* There are various versions of HTML supported, and the WebSite ought to be built in a version of HTML that is compatible. And this should be checkable.
- *Java, JavaScript, ActiveX.* Obviously JavaScript and Java applets will be part of any serious WebSite, so the quality process must be able to support these. On the Windows side, ActiveX controls have to be handled as well.
- *Cgi-Bin Scripts.* This is link from a user action of some kind (typically, from a FORM passage or otherwise directly from the HTML, and possibly also from within a Java applet). All of the different types of Cgi-Bin Scripts (perl, awk, shell-scripts, etc.) need to be handled, and tests need to check "end to end" operation. This kind of a "loop" check is crucial for E-commerce situations.
- *Database Access.* In E-commerce applications either you are building data up or retrieving data from a database. How does that interaction perform in real world use? If you give in "correct" or "specified" input does the result produce what you expect?

Some access to information from the database may be appropriate, depending on the application, but this is typically found by other means.

Navigation. Users move to and from pages, click on links, click on images (thumbnails), etc. Navigation in a WebSite often is complex and has to be quick and error free.

Object Mode. The display you see changes dynamically; the only constants are the "objects" that make up the display. These aren't real objects in the OO sense; but they have to be treated that way. So, the quality test tools have to be able to handle URL links, forms, tables, anchors, buttons of all types in an "object like" manner so that validations are independent of representation.

Server Response. How fast the WebSite host responds influences whether a user (i.e. someone on the browser) moves on or continues. Obviously, InterNet loading affects this too, but this factor is often outside the Webmaster's control at least in terms of how the WebSite is written. Instead, it seems to be more an issue of server hardware capacity and throughput. Yet, if a WebSite becomes very popular -- this can happen overnight! -- loading and tuning are real issues that often are imposed -- perhaps not fairly -- on the WebMaster.

Interaction & Feedback. For passive, content-only sites the only issue is availability, but for a WebSite that interacts with the user, how fast and how reliable that interaction is can be a big factor.

Concurrent Users. Do multiple users interact on a WebSite? Can they get in each others' way? While WebSites often resemble conventional client/server software structures, with multiple users at multiple locations a WebSite can be much different, and much more complex, than complex applications.

ASSURING WEBSITE QUALITY AUTOMATICALLY

Assuring WebSite quality requires conducting sets of tests, automatically and repeatably, that demonstrate required properties and behaviors. Here are some required elements of tools that aim to do this.

Test Sessions. Typical elements of tests involve these characteristics:

- *Browser Independent.* Tests should be realistic, but *not* be dependent on a particular browser, whose biases and characteristics might mask a WebSite's problems.
- *No Buffering, Caching.* Local caching and buffering -- often a way to improve apparent performance -- should be disabled so that timed experiments are a true measure of the Browser-Web-WebSite-Web-Browser response time.
- *Fonts and Preferences.* Most browsers support a wide range of fonts and presentation preferences, and these should not affect how quality on a WebSite is assessed or assured.
- *Object Mode.* Edit fields, push buttons, radio buttons, check boxes, etc. All should be treatable in object mode, i.e. independent of the fonts and preferences.

Object mode operation is essential to protect an investment in tests and to assure tests' continued operation when WebSite pages change. When buttons and form entries change location -- as they often do -- the tests should still work.

When a button or other object is deleted, that error should be sensed! Adding objects to a page clearly implies re-making the test.

- *Tables and Forms.* Even when the layout of a table or form varies in the browser's view, tests of it should continue independent of these factors.
- *Frames.* Windows with multiple frames ought to be processed simply, i.e. as if they were multiple single-page frames.

Test Context. Tests need to operate from the browser level for two reasons: (1) this is where users see a WebSite, so tests based in browser operation are the most realistic; and (2) tests based in browsers can be run locally or across the Web equally well. Local execution is fine for quality control, but not for performance measurement work, where response time *including* Web-variable delays reflective of real-world usage is essential.

WEBSITE VALIDATION PROCESSES

Confirming validity of what is tested is the key to assuring WebSite quality -- and is the most difficult challenge of all. Here are four key areas where test automation will have a significant impact.

Operational Testing. Individual test steps may involve a variety of checks on individual pages in the WebSite:

- *Page Quality.* Is the entire page identical with a prior version? Are key parts of the text the same or different?
- *Table, Form Quality.* Are all of the parts of a table or form present? Correctly laid out? Can you confirm that selected texts are in the "right place".
- *Page Relationships.* Are all of the links a page mentions the same as before? Are there new or missing links?
- *Performance, Response Times.* Is the response time for a user action the same as it was (within a range)?

Test Suites. Typically you may have dozens or hundreds (or thousands?) of tests, and you may wish to run tests in a variety of modes:

- *Unattended Testing.* Individual and/or groups of tests should be executable singly or in parallel from one or many workstations.
- *Background Testing.* Tests should be executable from multiple browsers running "in the background" [on an appropriately equipped workstation].
- *Distributed Testing.* Independent parts of a test suite should be executable from separate workstations without conflict.
- *Performance Testing.* Timing in performance tests should be resolved to 1 millisecond levels; this gives a strong basis for averaging data.
- *Random Testing.* There should be a capability for randomizing certain parts of tests.
- *Error Recovery.* While browser failure due to user inputs is rare, test suites should have the capability of resynchronizing

after an error.

Content Validation. Apart from how a WebSite responds dynamically, the content should be checkable either exactly or approximately. Here are some ways that should be possible:

- *Structural.* All of the links and anchors match with prior "baseline" data. Images should be characterizable by byte-count and/or file type or other file properties.
- *Checkpoints, Exact Reproduction.* One or more text elements -- or even all text elements -- in a page should be markable as "required to match".
- *Gross Statistics.* Page statistics (e.g. line, word, byte-count, checksum, etc.).
- *Selected Images/Fragments.* The tester should have the option to rubber band sections of an image and require that the selection image match later during a subsequent rendition of it. This ought to be possible for several images or image fragments.

Load Simulation. Load analysis needs to proceed by having a special purpose browser act like a human user. This assures that the performance checking experiment indicates true performance -- not performance on simulated but unrealistic conditions.

Sessions should be recorded live or edited from live recordings to assure faithful timing. There should be adjustable speed up and slow down ratios and intervals.

Load generation should proceed from:

- *Single Browser.* One session played on a browser with one or multiple responses. Timing data should be put in a file for separate analysis.
- *Multiple Independent Browsers.* Multiple sessions played on multiple browsers with one or multiple responses. Timing data should be put in a file for separate analysis. Multivariate statistical methods may be needed for a complex but general performance model.
- *Multiple Coordinated Browsers.* This is the most-complex form -- two or more browsers behaving in a coordinated fashion. Special synchronization and control capabilities have to be available to support this.

SITUATION SUMMARY

All of these needs and requirements impose constraints on the test automation tools used to confirm the quality and reliability of a WebSite. At the same time they present a real opportunity to amplify human tester/analyst capabilities. Better, more reliable WebSites should be the result.