

Web Testing: Tool, Challenges and Methods

Shakti Kundu

School of Computer Applications,

Lovely Professional University, Jalandhar-Delhi GT road, NH-1, Phagwara, Punjab, India.

Abstract

Web testing is the name given to software testing that focuses on web applications. Issues such as the security of the web application, the basic functionality of the site, its accessibility to handicapped users and fully able users, as well as readiness for expected traffic and number of users and the ability to survive a massive spike in user traffic, both of which are related to load testing. In this paper web testing based on tool, challenges and methods have been discussed which will help in handling some challenges during the website development. This paper presents best methods for testing a web application.

Keywords: *E-commerce, Tool, Web application, Web Testing.*

1. INTRODUCTION

Testers are still learning how to best test e-commerce applications, as most of these applications are business critical, and it is still a massive and growing marketplace. Millions of dollars have been spent on websites, and the investors expect success. Unfortunately, e-commerce history is filled with expensive failures. Some of which could have been avoided by better testing before the site was opened to the general public [1].

The repercussions of having a poorly operating website are staggering, and even affect the brick and mortar stores that the websites are enabling online. A survey based on recent study showed that when errors are found on an e-commerce website, 28% of the people stopped shopping at the site, 23% stopped buying from the site, and 6% of the people were so upset, that they stopped buying at brick and mortar store that the site is based on [2]. One can only have an idea or opinion that the customers feel that if the

company cannot provide a quality website, then they

may not be able to sell a quality product from their stores. Lots of user reviews can help in establishing this fact.

Other factors that are very important regarding web applications are security, reliability and recoverability. Uptime requirements for web applications are far more stringent than for off-the-shelf/shrink-wrap software. People expect that websites are secure, and are available twenty-four hours a day, seven days per week. When they are not, the business suffers [3].

The remainder of the paper is organized as follows: Section 2. describes web application performance tool, Section 3. describes the challenges during web testing, Section 4 highlights various web testing methods and finally some conclusions are provided in the Section 5.

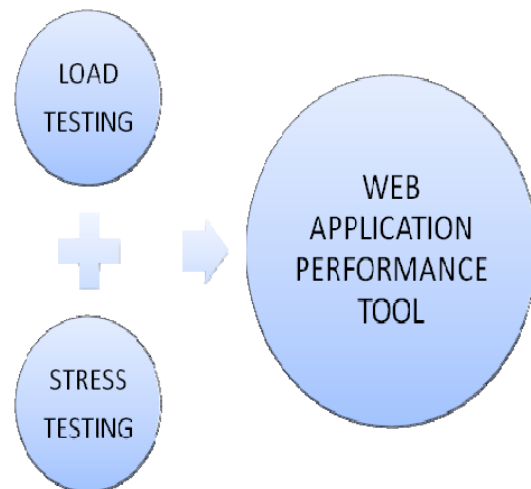


Figure 1. Web Application Performance Tool (WAPT)

2. WEB APPLICATION PERFORMANCE TOOL (WAPT)

A Web Application Performance Tool, also known as (WAPT) is used to test web applications and web related interfaces. These tools are used for load and

stress testing of web applications, web sites, web servers and other web interfaces. WAPT tends to simulate virtual users which will repeat either recorded URLs or specified URL and allows the users to specify number of times or iterations that the virtual users will have to repeat the recorded URLs. By doing so, the tool is useful to check for bottleneck and performance leakage in the website or web application being tested.

WAPT is a web load and stress testing tool from softlogica [5] having following specifications:

- Handles dynamic content and HTTPS/SSL;
- Easy to use;
- Support for redirects and all types of proxies;
- Clear reports and graphs.

3. CHALLENGES DURING WEB TESTING

A WAPT faces various challenges during testing and should be able to conduct tests for:

- Browser compatibility
- Operating System compatibility
- Windows application compatibility where required (especially for backend testing)

WAPT allows a user to specify how virtual users are involved in the testing environment i.e. either increasing users or constant users or periodic users load.

- Increasing user load, step by step is called Ramp [6] where virtual users are increased from 0 to hundreds, where, Ramp test is the test which uses escalating numbers of users over a given time frame to determine the maximum number of users the webserver can accommodate before producing error messages.
- Constant user load maintains specified user load at all time.
- Periodic user load tends to increase and decrease the user load from time to time.

Websites offer new challenges to developers, as well as testers. Scalability and performance are two areas that are significant in the e-commerce and web applications space. However, when new technology is used to make web applications perform well and scalable, new testing methodologies have to be created along with those technologies. Performance is critical, and based on a study from the Newport Group, more than half the recently deployed transaction-based web applications did not meet expectations for how many simultaneous users their applications could handle [4].

4. WEB TESTING METHODS

4.1 Functional Testing:

Test for all the links in web pages, database connection, forms used in the web pages for submitting or getting information from user, Cookie testing.

Check all the links:

- Test the outgoing links from all the pages from specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send the email to admin or other users from web pages.
- Test to check if there are any orphan pages.
- Lastly in link checking, check for broken links in all above-mentioned links.

Test forms in all pages: Forms are the integral part of any web site. Forms are used to get information from users and to keep interaction with them. Following are the check list on forms:

- First check all the validations on each field.
- Check for the default values of fields.
- Wrong inputs to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.

Cookies testing: Cookies are small files stored on user machine. These are basically used to maintain login sessions. Test the application by enabling or disabling the cookies in your browser options. Test if the cookies are encrypted before writing to user machine. If you are testing the session cookies (i.e. cookies expire after the session's ends) check for login sessions and user starts after session end. Check effect on application security by deleting the cookies.

Database testing: Data consistency is very important in web application. Check for data integrity and errors while you edit, delete, modify the forms or do any database related functionality. Check if all the database queries are executing correctly, data is retrieved correctly and also updated correctly.

4.2 Usability Testing:

Test for navigation: Navigation means how the user surfs the web pages, different controls like buttons, boxes or how user using the links on the pages to surf different pages.

It includes content checking and other user information for user help.

Content checking: Content should be logical and easy to understand. Check for spelling errors. Use of dark colors annoys users and should not be used in site

theme. You can follow some standards that are used for web page and content building. Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.

Other user information for user help: Like search option, sitemap, help files etc. Sitemap should be present with all the links in web sites with proper tree view of navigation. Check for all links on the sitemap. "Search in the site" option will help users to find content pages they are looking for easily and quickly. These are all optional items and if present should be validated.

For usability, the tests can be subjective, but there are standards and guidelines that have been established throughout the industry and it would be easy for a project team to blindly follow them, and feel that the site will be acceptable since the standards are followed.

A proactive suggestion is that while establishing the design guidelines, to define requirements that can be positively identified and measured [11]. A way to do this is to capture and quantify the meaning of learnability, understandability, and operability in a testable form. These concepts need to be formulated into testable requirements by describing how these concepts will be accomplished [3].

4.3 Interface Testing:

The main interfaces are: Web server and application server interface; Application server and Database server interface.

- Check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users.
- Check what happens if user interrupts any transaction in-between?
- Check what happens if connection to web server is reset in between?

4.4 Compatibility Testing:

Compatibility of your web site is very important testing aspect. It includes:

- Browser compatibility
- Operating system compatibility
- Mobile browsing
- Printing options

Browser compatibility: Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with. Your web site coding should be cross browser platform compatible. If you are using java scripts for User Interface functionality, performing security checks or validations then give more stress on browser compatibility testing of your web application.

Test web application on different browsers like Internet explorer, Firefox, Netscape navigator, AOL, Safari, Opera browsers with different versions.

OS compatibility: Some functionality in your web application is may not be compatible with all operating systems. All new technologies used in web development like graphics designs, interface calls like different API's may not be available in all Operating Systems.

Test your web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors.

Mobile browsing: This is new technology age. So in future Mobile browsing will rock. Test your web pages on mobile browsers. Compatibility issues may be there on mobile.

Printing options: If you are giving page-printing options then make sure fonts, page alignment, page graphics getting printed properly. Pages should be fit to paper size or as per the size mentioned in printing option.

4.5 Performance testing:

Web application should sustain to heavy load. Web performance testing should include: Web Load Testing and Web Stress Testing.

In *Web load testing*, the focus is to test if many users are accessing or requesting the same page. Can system sustain in peak load times? Site should handle many simultaneous user requests, large input data from users, Simultaneous connection to database, heavy load on specific pages etc.

Stress testing: Generally stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checked how system reacts to stress and how system recovers from crashes. Stress is generally given on input fields, login and sign up areas.

In web performance testing web site functionality on different operating systems, different hardware platforms are checked for software, hardware memory leakage errors.

Performance testing is the validation that the system meets performance requirements. This can be as simplistic as ensuring that a web page loads in less than eight seconds, or can be as complex as requiring

the system to handle 10,000 transactions per minute, while still being able to load a web page within eight seconds. For example, even the best server cannot make a difference if the firewall machine, and more importantly the number of firewall rules, are not the same [9].

In the execution of performance tests, Weyuker and Vokolos (2000) [10], who provide a detailed case study on performance testing, outline typical steps to create performance test cases. These are:

- Identify the software processes that directly influence the overall performance of the system.
- For each of the identified processes, identify only the essential input parameters that influence system performance.
- Create usage scenarios by determining realistic values for the parameters based on past use. Include both average and heavy workload scenarios. Determine the window of observation at this time.
- If there is no historical data to base the parameter values on, use estimates based on requirements, an earlier version, or similar systems.
- If there is a parameter where the estimated values form a range, select values that are likely to reveal useful information about the performance of the system. Each value should be made into a separate test case.

Performance testing can be done through the “window” of the browser, or directly on the server. If done on the server, some of the performance time that the browser takes is not accounted for. Scripting GUI orientated transactions to drive the browser interface can be much more complicated and the synchronization between test tool and browser it not always reliable. Therefore, if testers decide to ignore the performance time taken by the browsers, it is important to get “buy in” from project team members and users to understand the compromise. If there are issues, testers should advise management that performance-testing using the GUI will introduce a time-intensive effort that may or may not impact the project timeline [11].

To assist with load and performance testing, testers should use the test scripts that have been created early in the project as a basis for initial load testing. By using the existing scripts, this avoids rework and allows the scripts to be used at different times by different virtual users when validating system performance [12].

4.6 Security Testing:

Following are some test cases for web security testing:

- Test by pasting internal url directly into browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing url options directly. i.e. if you are checking some publisher site statistics with publisher site ID= 123, then changing the url site ID parameter directly to different site ID which is not related to logged in user. Access should deny for this user to view others stats.
- Try some invalid inputs in input fields like login username, password, and input text boxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given download option.
- Test if SSL is used for security measures. If used proper message should get displayed when user switch from non-secure http:// pages to secure https:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on web server.

Security is a critical part of an e-commerce website. The security best practices listed below come from an outstanding article on website security testing by Russ Smith (2001) [8].

Data Collection: Web sites collect data in log files, as well as through forms in which users supply to the website information that is saved on the web server.

- The web server should be setup so that users cannot browse directories and obtain file names. For example, www.example.com/data should not list the files in that folder.
- Data should be secured internally so unauthorized employees do not have access.

Get vs. Post: Technically, there are two ways that information can be collected from a web page when a user submits a web form, one is a GET and one is a POST. A GET shows some information in the Uniform Resource Locator (URL) that could be sensitive, where the POST does not.

- Early in the project, encourage developers to use the POST command wherever possible.
- When testing, check URLs to ensure that

there are no “information leaks” due to sensitive information being placed in the URL while using a GET command.

Cookies: A cookie is a text file that is placed on a website visitor’s system that identifies the user’s “identity.” The cookie is retrieved when the user re-visits the site at a later time. Cookies can expire in a short period of time, such as minutes or hours (session cookie) or can last for months or years (persistent cookie).

- Cookies can be controlled by the user, regarding whether they want to allow them or not. If the user does not accept cookies, will the site still work?

- Are the cookies necessary? Cookies should be used judiciously so that if users have their browsers at the setting to “provide warning message when cookies are used” and multiple cookies are used, the user will be inundated with multiple warning boxes throughout their visit, perhaps discouraging them from re-visiting the site.
- Is sensitive information in the cookie? If multiple people use a workstation, the second person may be able to read the sensitive information saved from the first person’s visit. Information in a cookie should be encoded or encrypted.

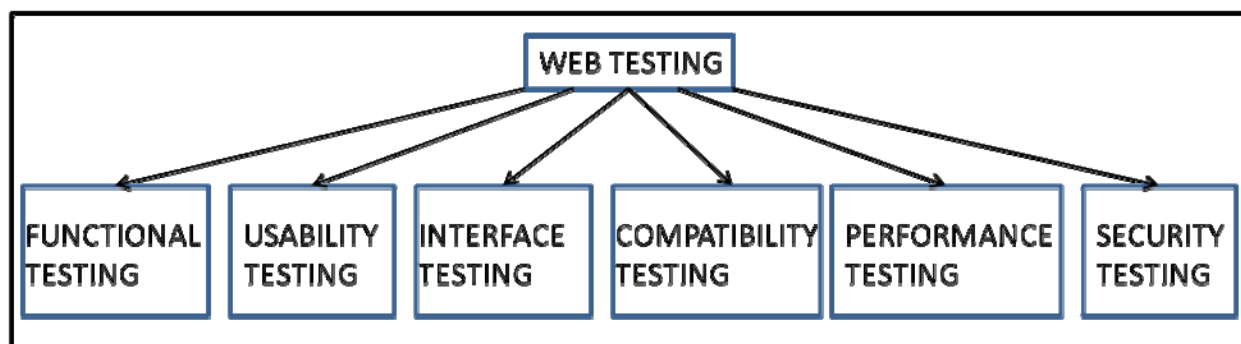


Figure 2. Web Testing Methods.

5. CONCLUSIONS

In this research paper, the importance of website testing have been highlighted which is one of the new breed of testing for the past few years. A survey on web testing methods and challenges described some issues and challenges and ways to avoid same issues.

Web testing is a challenge exercise and by following the best ways and methods described in this paper, some of the challenges can be overcome.

REFERENCES

[1] Samaroo, A., Allott, S., & Hambling, B. (1999). Effective Testing for E-Commerce. Retrieved from the World Wide Web: http://www.stickyminds.com/docs_index/XML0471.doc

[2] Gerrard, P. (2000a). Risk-Based E-Business Testing: Part 1 – Risks and Test Strategy. Retrieved from the World Wide Web: <http://www.evolutif.co.uk/articles/EBTestingPart1.pdf>

[3] MacIntosh, A. & Strigel, W. (2000). “The Living Creature” – Testing Web Applications. Retrieved from

the World Wide Web: http://www.stickyminds.com/docs_index/XUS11472file1.PDF

[4] Shea, B. (2000, May/June). Avoiding Scalability Shock. Software Testing & Quality Engineering Magazine, 42-46.

[5] Retrieved from the World Wide Web: <http://www.softlogica.com/index.shtml> and <http://www.loadtestingtool.com/>

[6] Retrieved from the World Wide Web: <http://www.paessler.com/webstress/features>

[7] Han, J. and Kamber, M. (2004), “The concept and technology of data mining”. Beijing: China Machine Press, 2004.

[8] Smith, R. (2001, Jan/Feb). Behind Closed Doors: What every tester should know about web privacy. Software Testing & Quality Engineering Magazine, 43-48.

[9] Hagen, M. (2000). Performance Testing E-Commerce Web Systems Presentation Paper. Retrieved from the World Wide Web: http://www.stickyminds.com/docs_index/XDD2070filelistfilename1.doc

[10] Weyuker, E & Vokolos, F. (2000). Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study. IEEE Transactions on Software Engineering, 25 (12), 1147-1156.

[11] Gerrard, P. (2000b). Risk-Based E-Business Testing:

Part 2 – Test Techniques and Tools. Retrieved from the World Wide Web:
<http://www.evolutif.co.uk/articles/EBTestingPart2.pdf>
[12] Courtney, P. (1999, July). Testing e-commerce. Applications Development Trends, 24-34.

Shakti Kundu received his MCA from Kurukshetra University Kurukshetra, India in 2006, MPhil in Computer Science from Chaudhary Devi Lal University, Sirsa, Haryana, India in 2008, MTech in Computer Science & Engineering from Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India in 2010. Currently he is pursuing his Ph.D in Computer Science from Manav Bharti University, Solan, Himachal Pradesh, India. The author current research interests are web mining and web testing. He is individual member of Computer Society of India and life member of Indian Society for Technical Education (I.S.T.E), New Delhi.