The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

A Comparative Evaluation of approaches for Web Application Testing

Pourya Nikfard
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia
Npourya2@live.utm.my

Assoc. Prof. Dr. Suhaimi bin Ibrahim
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia
suhaimiibrahim@utm.my

Mohammad Hossein Abolghasem Zadeh
Advanced Informatics School (AIS)
University Technology Malaysia (UTM)
Kuala Lumpur, Malaysia
Hazmohammad2@live.utm.my

***Abstract*-Testing is one of the five main technical activity areas of software engineering. Software testing is important in quality assurance of web applications, in which test case is crucial. Compared with other software systems, web applications have many differences in the testing process. Web testing is an effective technique to ensure the quality of web applications. A number of approaches have been presented, to tackle this problem. In this paper, we classify these approaches into two classes (Functional Requirement and Non-Functional Requirement). Then, we evaluate these approaches based on some criteria (like testing criteria, technique used, performance, reliability, accuracy, testing method and testing criteria). Exploration of that categorization will help researchers who are working on web application testing to deliver more applicable solutions.**

I. Introduction

In current years, with the quick expansion of the Internet and WWW, Web applications have become
common in the entire world. As more and more businesses rely on Web applications, the value and consistency of the applications become essential. Testing can discover the majority of errors of the software. So, Web applications should be tested carefully to make sure that the requirement specifications are fulfilled by the applications. Web application testing is more complex than usual programs because of their characters of distribution, heterogeneity, platform independence and concurrence. The large dispersion of Internet has created an important enlargement of the order of Web based applications with more and more severe requirements of security, operability, reliability and interoperability. With the quick growth of Internet, Web applications rapidly sweep the world that are applied more widely and deepness in some parts like government, industry, education, and finance. Simultaneously, the quality of Web applications is one of the most important things to success of an organization or a department. However, both complexity and scale of Web applications are rising, and the cycle of development is becoming shorter and shorter, while the Web applications' quality is more and more becoming an concern issue, so that the Requirements that are very high are not only put forward for development of Web application, but also the requirements for testing technology for Web applications are more strict. In order to make sure the consistency and quality of Web applications, the Web applications testing research require being make stronger continually. As a pattern of software with the rising popularity, Web application has been paid a lot of attentions and developed quickly. Web applications have many specifications like distributed, heterogeneous, platform independent and concurrent. These characteristics make the testing for them more complex and hard rather than traditional software. So, to discover an appropriate technology of testing for web applications has become precedence. Software testing is a hard and complex job and web application testing may be even more complicated, due to the peculiarities of these applications. In the previous years, research has been done to concentrate on a lot of problems in the web application testing and a lot of solutions and answers have been suggested. The functionality, quality and reliability of Web applications are significant issues because software can block the whole of businesses and verify strong mortification. These issues have improved the requirements for models, methodologies and tools to develop the testing and designing of Web applications. Significant issues for Web applications are complexity, speed, design maturity and large dimensions. The testing of Web applications is not a simple job.

In this paper, we focus on the web application problem and offer a survey of recent approaches. We then compare them with respect to a set of criteria. By offering this

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

overview and classification of existing proposals for web applications, we hope this research helps researchers to deliver wellbuilt approaches.

II. Classification of the web applications approaches

A. *Different classes of web applications*

There are different classes of Web applications, which are considered for different points and intended at different target group. Dart [Dart, 1999] classified the most frequently encountered Web sites according to the functionalities offered, and classes belonging to Web applications are summed up below:

Deliverable: Information can be downloaded from the servers by End users, e.g. software upgrade.

Customizable: Content can be customized by End users to fit specific preferences, e.g. Email setting system.

Acceptable: Information can be inputted and submitted it to the servers by End users, e.g. subscription to newsletters.

Interactive: Mutual communication between users or sites, e.g. business-to-business.

Transactional: Goods can be bought by End users, e.g. online tickets shop.

Service oriented: Services can be received by End users regularly, e.g. virus scan program per week.

Accessible: Queries can be made into a large database and extracted information by End users, e.g. supplier looks up catalogue of parts.

Data warehouse: Queries can be made into a collection of large databases and extracted information, e.g. Google search engine.

Automatic: Providing/proposing content that are generated automatically to end users, e.g. mySimon suggestion of software agent according to contents of end users' browsing.

The nine tips summed up above are the classes of Web applications that are classed according to their principal functionalities. However, In spite of the different classes of Web applications, they have frequent individualities that are the actuality of life for researchers and developers.[24]

B. *Testing the Functional Requirements of a Web Application*

Testing the functional requirements of an application endeavors at proving that features of application and operational actions correspond to their specifications. In other words, this kind of testing is accountable for discovery failures of application due to errors in the implementation of functional requirements, rather than failures due to the environment of application's running. To accomplish this plan, any failures due to the running environment should be avoided, or decreased to a minimum. Preliminary statements about the environment of running will have to be completed before design the test and execution.

Most approaches and methods used to test the functional requirements of "traditional" software can be used for Web applications, too. Equally to traditional software testing, functionality testing of Web application has to depend on the following basic aspects:

- *Testing levels*, which identify the diverse range of the tests to be completed, i.e. the collections of components to be tested.
- *Test strategies*, which describe heuristics or algorithms to generate test cases from models of software representation, models of implementation, or models of test.

Test models, which represent the associations between an elements of representation or a implementation of component [25].

Testing processes, which describe the flow of testing activities, and other decisions such as when to start testing, who is to do the testing, how much endeavor should be used, etc.

However, in spite of their similarity to conventional applications, Web applications as well have distinctive features that cause precise problems for each characteristic explained in the preceding list. For instance, the description of *testing levels* for a Web application needs superior concentration than that applied to traditional software. At the *unit testing* level, the range of a unit test cannot be distinct exclusively, because it depends on the existence of diverse kinds of components (e.g. Web pages, script functions, embedded objects) existing in on both the client and server side of an application. In relative to *integration testing*, the frequent diverse mechanisms used to incorporate a heterogeneous of application and dispersed components can produce some coupling levels and data flow among the components, which have to be measured to institute a correct incorporation strategy.

As for the *strategies for test design*, the traditional approaches of black box, white box, or grey box testing may be taken into description for designing test cases, supplied that preliminary reflections are distinct. In common, *black box testing* of Web applications will not be diverse from black box testing of software applications. In both cases, using a programmed treatment condition, a sufficient set of test cases is distinct based upon the particular functionality of the thing to be tested. However, a Web application's definite features can influence test design and execution. For instance, testing of components dynamically created by the running application can be very costly, because of the complexity of identifying and

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

redeveloping the same situations that created each component. So, models of traditional testing used to represent the behavior of an application may have to be modified to this individuality and to the running environment of Web application. *White box testing*, irrespective of a nature of application, is typically based on coverage criteria that take into explanation structural features of the application or its components. Sufficient models representing structures of application or component are used, and coverage criteria and test cases are properly specified. The endeavor of white box testing is to wrap the structural elements measured. Since the architecture and components of a Web application are mostly diverse from those of a traditional application, suitable models representing structural information at diverse levels of granularity and abstraction are needed, and coverage criteria have to be distinct accordingly. For instance, models representing navigation as well as traditional structural features of an application require to be taken into account. Coverage criteria must spotlight both on hyperlinks, which permit user navigation in the application, and on internal items of component of application (e.g. its code statements).

In addition black and white box testing, *grey box* testing can also be measured for Web applications. Grey box testing is a combination of black and white box testing, and believes both the behavior of application, from the end user's perspective (same as black box testing), and the application's internal structure and technology (same as white box testing). According to [26], grey box testing is appropriate for testing Web applications because it issues in high-level design, environment, and interoperability circumstances. It is anticipated that this kind of testing will expose problems that are not simply recognized by black box or white box analysis, in particular problems interrelated to end-to-end information flow and dispersed hardware or software system configuration and compatibility. Context-specific failures related to Web applications are usually uncovered using grey-box testing. Lastly, for the *testing processes*, the traditional approach for testing execution that begins from unit test and proceeds with integration, system testing, and acceptance testing can also be in use into account for Web applications. For each stage, however, distinctions with admiration to testing traditional software have to be perceived and explicit solutions have to be considered. A significant testing process issue is, for example, to set up an *environment* to carry out tests at each stage: driver or stub modules are typically needed to run tests at the unit or integration phase. Results for testing a Web application have to clearly consider the application's dispersed running environment, and to accept the essential communication mechanisms for performing the components being tested. [27, 28]

Generate test case from functional requirement of web applications, Generating test case from modeling database interaction in web applications, Testing web application based on web browser interaction, Call for testing (CFT), Generating test case from component based web applications, Testing web application based on use cases, Beta testing of web application, Generating test case of web application based on Z specification, Generate automated test code for web application with activity oriented approach, Using probable FSM for testing web application , An approach called on the fly for testing web applications, Surveying some methods of testing of web application and Adding the agent to intelligent web application testing approaches can be assigned as Functional requirement of web applications.

C. *Testing the Non-Functional Requirements of a Web Application*

There are diverse non-functional requirements that a Web application, either clearly or unreservedly, is typically needed to satisfy. For each nonfunctional requirement, testing behaviors with precise goals will have to be considered. An explanation of the confirmation activities that can be executed to test the major non-functional requirements of a Web application are presented below:

Performance Testing: Performance testing is performed to confirm specified performance of the system (e.g. response time, service availability). Frequently, performance testing is carried out by simulating hundreds, or even more, concurrent user accesses above a distinct time period. Information regarding accesses is recorded and then analyzed to approximate the load levels exhausting the resources of the system. In the case of Web applications, performance of the system is a significant issue because Web users do not desire to stay too long for a reply to their requirements; as well, they also anticipate that services will be available forever. Successful performance testing of Web applications is a significant task because it is not feasible to know in advance how many users will really be associated to a real-world running application. So, performance testing should be measured as an endless activity to be performed by analyzing data from access log files, with the intention of tune the system sufficiently. Failures that can be exposed by performance testing are mostly caused by running

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

environment faults (e.g. scarce resources, poorly deployed resources), even if any software component of the application level may contribute to inadequacy, i.e. components executing any business rule by algorithms that are not optimized.

Load Testing: Load testing is frequently used as a synonym for performance testing but it varies from the latter because it needs that system performance be assessed with a predefined load level. It goals to evaluate the time required to perform a number of tasks and functions under predefined situations. These predefined situations comprise the minimum configuration and the maximum activity levels of the running application. Also, in this case, many simultaneous user accesses are simulated. Information is recorded and, when the tasks are not performed inside predefined time limits, failure reports are created. As for the complexities of performing load testing of Web applications, reflections similar to the ones made for performance testing can be taken into account, too. Failures found by load testing are mostly due to faults in the running environment.

Stress Testing: Stress testing is carried out to assess a system or component at or further than the limits of its particular requirements. It is used to assess the response of the system at activity peaks that can exceed limitations of system, and to authenticate if the system crashes or is capable to improve from such situations. Stress testing varies from performance and load testing because the system is performed on or further than its breaking point, while performance and load testing simulate usual user activity. In the case of Web applications, stress testing complexities are similar to those that can be gathered in performance and load testing. Failures found by stress testing are mostly due to errors in the running environment.

Compatibility Testing: Compatibility testing is performed to verify if an application runs as anticipated on a running environment that has various mixtures of hardware, software, and middleware. In the case of Web applications, compatibility testing will have to discover failures because of the usage of diverse platforms of Web server or browsers of client, and corresponding releases or configurations. The large diversity of feasible combinations of all the components involved in the carrying out of a Web application does not make it possible to test them all; thus typically only the most general mixtures are considered. As a result, just a subset of feasible compatibility failures might be exposed. Both the application and the running environment can be responsible for failures of compatibility. A common rule for avoiding compatibility failures is to supply users of Web application with suitable information about the expected configuration of the running environment and with appropriate diagnostic messages to cope with any incompatibilities found.

Usability Testing: Usability testing goals to authenticate to what extent an application is simple to use. Typically, design and implementation of the user interface both influence usability. so, usability testing is mostly centered around testing interface of the user: issues regarding the exact content rendering (e.g. graphics, text editing format) as well as the simplicity of messages, prompts, and commands that are to be measured and confirmed. Usability is a serious issue for a Web application. Indeed, it may verify the success of the application. As a result, front-end of application and the method users interrelate with it often are features that are given larger care and awareness during the development process of application. When Web application usability testing is performed, issues interrelated to an application's navigation fullness, rightness, and succinctness are also measured and confirmed. This kind of testing should be an interminable activity performed to develop the usability of a Web application; techniques of user profiling are typically used to attain this goal. The application is mostly conscientious for usability failures.

Accessibility Testing: Accessibility testing can be measured a particular kind of usability testing whose goal is to authenticate that the access to content application is permitted even in the presence of reduced hardware and software configurations on the client side (e.g. browser configurations stopping graphical visualization, or scripting execution), or in the presence of users with disabilities, such as visual impairment. In the case of Web applications, accessibility policies such as the one supplied by the Web Content Accessibility Guidelines [29] have been recognized, so that accessibility testing characterizes confirmation the compliance of an application with such rules. The application itself is usually the major reason of accessibility problems, even when accessibility failures may be because of the configuration of the running environment (e.g. browsers where the implementation of scripts is immobilized).

Security Testing: Security testing goals to authenticate the efficiency of the overall defences of Web application against undesired contact of illegal users, its ability to protect system resources from inappropriate use, and granting certified users access to certified services and resources. Application defences have to supply security mechanisms capable to avoid or decrease damage by reason of intrusions, with expenses that should be appreciably less than damages reasoned by a security break. Application vulnerabilities affecting safety may be controlled in the code of application, or in any of the diverse hardware, software, and middleware components. Both the running environment and the application can be

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

conscientious for security breakdowns. In the case of Web applications, heterogeneous implementations and execution technologies, jointly with the very large number of probable users and the opportunity of accessing them from wherever, can make Web applications more defenseless than traditional applications and security testing more complicated to achieve. [27, 28] Testing web application based on use cases and Generating test case from modeling database interaction in web applications approaches can be assigned as Non-Functional requirement of web applications.

III. Comparative evaluation

In this section we evaluate the above web application approaches with respect to the subsequent criteria. We declare that, any approach to web application should satisfy these set of criteria [31].

Performance: This represents how fast a Web Application request can be completed. According to [30], performance can be considered in terms of throughput, latency, execution time, and transaction time. The answer time of a Web application can also be quantify of the performance. High-quality Web application should supply higher throughput, lower latency, lower execution time, faster transaction time and faster answer time. It will be defined by timely manner. If the time of executing the program was less and down, the performance is better.

Reliability: This represents the capability of a Web application to carry out its functions (that is, to preserve its Web application quality). It can be considered by the number of breakdowns of the Web application in a confident time interval. If the testing standard manners are used, the reliability is high.

Accuracy: The accuracy of a quantity is the level of nearness of measurements of a quantity to that actual value of quantity. If the mathematical and logical manners are used, the accuracy is high.

Testing method: it divides in to two parts, white box testing and black box testing. The former that also known glass-box testing and Structural testing find errors that happen in internal structure of program such as database, coding, subassemblies and chips. White-box testing talk about *how* systems do their works. the latter one that also known behavioral testing is used for finding errors in feature's level, high level operations, scenarios of customers and operational profiles. Black-box testing talking about *what* systems should do.

Testing criteria: it divides in to two parts, control flow and data flow. First one talking about the processes like activity diagrams. But second one talking about the parameters and messages that transferred and also talking about the relation between data.

Testing targets: it divides in to two parts, functional and non-functional. In functional testing the input and output of a system will be tested. It explains "what" the capability of the system is. Nonfunctional requirements explain the attributes of behavior of functions or systems. For example, "how" or with what quality the system should do its function.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

TABLE I. COMPARING WEB APPLICATION TESTING APPROACHES

	Testing criteria	Technique used	Performance	Reliability	Accuracy	Testing method		Testing criteria		Testing targets	
						White box	Black box	Control flow	Data flow	functional	Non functional
[1]	Generate test case from functional requirement of web applications	Formal language (object Z modeling)	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No
[2]	Generating test case from modeling database interaction in web applications	GFSM GFSMTT	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
[3]	Testing web application based on web browser interaction	FSM	Not Applicable	Yes	Yes	No	Yes	No	Yes	Yes	No
[4]	Call for testing (CFT)	User acceptance testing	Yes	Yes	Not Applicable	Yes	Yes	Yes	Yes	Yes	No
[5]	Generating test case from component based web applications	finite state machine (FSM) Logical component (LC) Black box testing Automaton Composition of automata	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
[6]	Testing web application based on use cases	Unified Modeling Language (UML) use case diagrams use case transitions models (UCTM) Logical component (LC) constraint directed diagrams (CDD) constraint scenario coverage (CSC)	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

[7]	Beta testing of web application	Beta testing	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No
[8]	Generating test case of web application based on Z specification	Formal language (Z specification)	Yes	Yes	Not Applicable	Yes	No	Yes	Yes	Yes	No
[9]	Generate automated test code for web application with activity oriented approach	Activity oriented approach Black box testing	Yes	Yes	Not Applicable	No	Yes	Yes	No	Yes	No
[10]	Using probable FSM for testing web application	FSM Usage model of portable FSM	Yes	Yes	Yes	No	Yes	Yes	No	Yes	No
[11]	Adding the agent to intelligent web application testing	Performance testing Regression testing	Yes	Yes	Not Applicable	Yes	Yes	Yes	No	Yes	No
[12]	Surveying some methods of testing of web application	UML ReWeb/TestWeb State based testing	Yes	Yes	Not Applicable	Yes	No	No	Yes	Yes	No
[13]	An approach called on the fly for testing web applications	UML FSM	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

IV. Conclusion

This paper has aimed to provide an overview and compare recent progress in web application testing. We classify these approaches to two categories, functional requirement of web applications and non-functional requirement of web applications. We study and survey several approaches. But we cannot claim that these approaches are comprehensive and exhaustive. Finally, we have comparison table with different columns that compares all the approaches. The main problems with most of these approaches to testing are the non-functional testing of testing target and accuracy of testing. It should be noted that although the above comparison is conducted based on

some prominent approaches, the outcome of this research is not restricted to such approaches. In other words, my considered criteria either can be served as features to be included in a newly developing system or may be applied to help generally evaluating or selecting software testing approaches. However the results of my comparison show that there is no single superior software testing approach in all cases. Therefore, deciding which approach to use in a certain scenario should be done based on its specifications and combine and present the new approach that is more comprehensive and mature is my future work.

V. Acknowledgement

This research project is sponsored by the Universiti Teknologi Malaysia (UTM) under the UTM-RUG, Vol No. 00H68. Thanks to UTM and individuals who are directly or indirectly involved in this project.

REFERENCES

[1] Zhu, B., et al. Generating Test Case from Functional Requirement of Web Applications. 2nd International Symposium on Electronic Commerce and Security, ISECS 2009, v 2, p 465-468, ISECS 2009.

[2] Song, B., H. Miao, and Z. Chen. Modeling Database Interactions in Web Applications and Generating Test Cases.

[3] Zhu, B., H. Miao, and L. Cai. Testing a Web Application Involving Web Browser Interaction. 10th ACIS Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2009, In conjunction with IWEA 2009 and WEACR 2009, p 589-594, 2009.

[4] Yu, L., et al. Towards Call for Testing: An Application to User Acceptance Testing of Web Applications. Proceedings - 33rd Annual IEEE International Computer Software and Applications Conference, v 1, p 166-171, Proceedings - COMPSAC 2009.

[5] Miao, H., et al. An Approach to Generating Test Cases for Testing Component-based Web Applications. Proceedings - Workshop on Intelligent Information Technology Application, IITA 2007, p 264-269, 2007.

[6] Li, L. and H. Miao. An approach to modeling and testing web applications based on use cases. 2008 International Symposium on Information Science and Engineering, ISISE 2008, v 1, p 506-510ISISE 2008.

[7] Zemin, Z. Study on beta testing of web application. 2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010, v 1, p 423-426, 2010.

[8] Youxin, M. and W. Dafa. Research on Framework of Test Case Generation of Web Applications Based on Z Specification. Proceedings - 2009 International Forum on Information Technology and Applications, IFITA 2009, v 2, p 555-558, 2009 Proceedings - 2009 International Forum on Information Technology and Applications, IFITA 2009, v 2, p 555-558, 2009.

[9] Turner, D., et al., An Automated Test Code Generation Method for Web Applications using Activity Oriented Approach. ASE 2008 - 23rd IEEE/ACM International Conference on Automated Software Engineering, Proceedings , p 411-414, 2008.

[10] Zhongsheng, Q. An Approach to Testing Web Applications Based on Probable FSM. Proceedings - 2009 International Forum on Information Technology and Applications, IFITA 2009, v 1, p 519-522, 2009.

[11] X, L. and B. X, Applying Agent into Intelligent Web Application Testing. 2007 International Conference on Cyberworlds, CW'07, p 61-65, 2007.

[12] Kam, B. and T.R. Dean, Lessons learned from a survey of Web Applications Testing. ITNG 2009 - 6th International Conference on Information Technology: New Generations, p 125-130, 2009.

[13] Li, L., Z. Qian, and T. He. An Approach to Testing Web Applications On-The-Fly. 2009 International Conference on Management of e-Commerce and e-Government, ICMeCG 2009, p 428-431, 2009.

[14] Elbaum, S., S. Karre, and G. Rothermel. Improving web application testing with user session data. Proceedings - International Conference on Software Engineering, p 49-59, 2003.

[15] Ricca, F. Analysis, testing and re-structuring of Web applications. IEEE International Conference on Software Maintenance, ICSM, p 474-478, 2004.

[16] Ricca, F. and P. Tonella. Analysis and testing of web applications. Proceedings - International Conference on Software Engineering, p 25-34, 2001.

[17] Ricca, F. and P. Tonella. Web application slicing. 2001: IEEE International Conference on Software Maintenance, ICSM, p 148-157, 2001.

[18] Ricca, F. and P. Tonella. Construction of the system dependence graph for web application slicing. Automated Software Engineering, v 12, n 2, p 259-288, April 2005.

[19] Ricca, F. and P. Tonella, Understanding and restructuring Web sites with ReWeb. IEEE Multimedia, v 8, n 2, p 40-51, April/June 2001.

[20] Ricca, F. and P. Tonella. Web site analysis: Structure and evolution. Conference on Software Maintenance, p 76-86, 2000.

[21] Di Lucca, G. and A. Fasolino, Testing Web-based applications: The state of the art and future trends. Proceedings – 29th annual International Computer Software

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.50 e-ISSN: 2251-7545

and Applications Conference, v 2, p 65, 2005COMPSAC 2005.

[22] Sprenkle, S., et al. Automated replay and failure detection for web applications. 20th IEEE/ACM International Conference on Automated Software Engineering, ASE 2005, p 253-262, 2005.

[23] Sprenkle, S., et al. An empirical comparison of test suite reduction techniques for user-session-based testing of web applications. Proceedings of the 21st IEEE International Conference on Software Maintenance, ICSM, v 2005, p 587-600, 2005

[24] Li Ye, Model based approach for web applications, master thesis, June 2007.

[25] Binder RV (1999) Testing Object-Oriented Systems. Models, Patterns, and Tools. Addison-Wesley: Reading, MA.

[26] Liu C, Kung DC, Hsia P, Hsu C (2000) Object-based Data Flow Testing of Web Applications. In: Proceedings of First Asia-Pacific Conference on Quality Software. IEEE Computer Society Press, Los Alamitos, CA, pp 7–16.

[27] Giuseppe A. Di Lucca, Anna Rita Fasolino, Web Application Testing.

[28] Giuseppe A. Di Lucca and Anna Rita Fasolino, Testing Web-based applications: The state of the art and future trends, 0950-5849/$ - see front matter © 2006 Elsevier B.V. All rights reserved. doi:10.1016/j.infsof.2006.06.006.

[29] Web Content Accessibility Guidelines 2.0 (2005), http://www.w3.org/TR/WCAG20 (accessed 5 June 2005)

[30] Rajesh, S. and D. Arulazi, Quality of service for Web services – demystification, limitations, and best practices.

[31] P. Nikfard, H. Selamat and N. Mahrin, Functional Testing on Web Applications, Postgraduate Annual Research on Informatics Seminar, Advanced Informatics School, University Technology Malaysia (UTM), 2012.