A Review on Web Application Testing and its Current Research Directions

D. Rajya Lakshmi[1], S. Suguna Mallika[2]
[1]Department of Computer Science and Engineering, JNTUK-UCEN, India
[2]Department of Computer Science and Engineering, CVR College of Engineering, India

Article Info

Article history:

Received Jun 6, 2017
Revised Apr 8, 2017
Accepted Apr 22, 2017

Keyword:

Automated testing
Coverage testing
Testing techniques
Testing tools
Web application testing

ABSTRACT

Testing is an important part of every software development process on which companies devote considerable time and effort. The burgeoning web applications and their proliferating economic significance in the society made the area of web application testing an area of acute importance. The web applications generally tend to take faster and quicker release cycles making their testing very challenging. The main issues in testing are cost efficiency and bug detection efficiency. Coverage-based testing is the process of ensuring exercise of specific program elements. Coverage measurement helps determine the "thoroughness" of testing achieved. An avalanche of tools, techniques, frameworks came into existence to ascertain the quality of web applications. A comparative study of some of the prominent tools, techniques and models for web application testing is presented. This work highlights the current research directions of some of the web application testing techniques.

Corresponding Author:

S. Suguna Mallika,
Department of Computer Science and Engineering,
CVR College of Engineering,
Vastunagar, Mangalpally, Ibramhimpatnam, R.R. District, India.
Email: suguna.kishore@gmail.com

1. INTRODUCTION

With the advent of internet revolution and the colossal rise in the development of web applications as well as their corresponding usage, it is becoming mandatory for quality testing of web applications. Web Application Testing is gaining importance given the major stake of economic relevance in the contemporary society. The cost of fixing a bug is directly proportional to the time of its discovery. The longer the time it takes to unearth a bug, the costlier it becomes to fix it as the software would have been distributed or under use by the customers[1].

The mammoth customer base and a global distribution of the customers in the case of web application testing (WAT), software bugs detected late proved pricy for the applications owners in the past and will continue to be in the future [1]. The versatility of web applications is a predominant feature which is making the testing of web applications a tough job [2-4]. However, high quality testing would always contribute in better customer retention and loyalty for web applications, thereby directly contributing to a thriving and sustaining business. The principal feature of web applications that differentiates it from traditional testing on desktop is that web applications are completely heterogeneous in nature at various levels [4], [5]. Features of web applications which discriminate them from customary desktop applications are highlighted in Figure 1.

The enormous customer base, a heterogeneous execution environment, heterogeneous languages used for component development, heterogeneous operating systems, faster maintenance rate, multi-tier architecture, transactional concurrency, dynamic state changes (like for e.g., pressing of back button on browser et. al.) are the principal features which discriminate web applications from standalone applications [2]. Such complexities inherent with web applications make the testing of web applications a challenging job thus establishing a clear need for more sophistication in the WAT. To quote an example, PayPal had to pay huge compensation for its customers for a small service outage resulting from a faulty upgradation of its website [6].

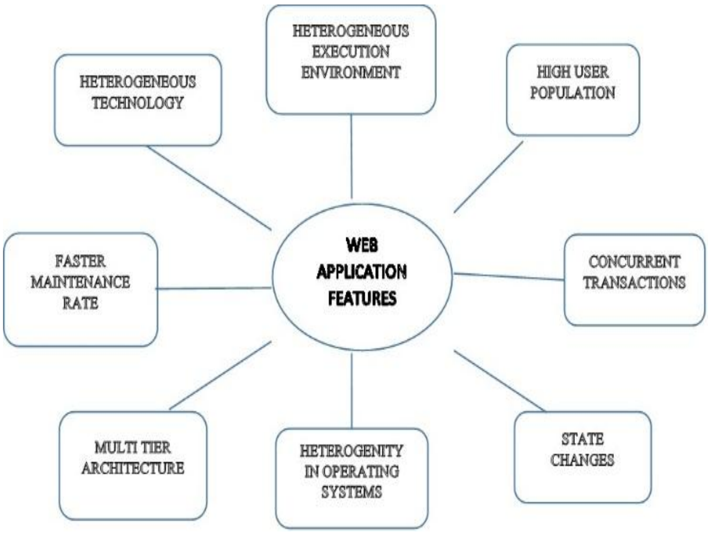

Figure 1. Features of Web Applications

A focus on all the major web application testing challenges is presented in Section 2, different architectures proposed for web application testing earlier are consolidated in section 3, various models available for web application testing are presented in section 4, several methods, techniques for web application testing are presented in Section 5. Some of the top testing tools available for automated testing of web applications and a comparative study of the applications are presented in Section 6.

2. WEB APPLICATION TESTING CHALLENGES

With a unique set of characteristics for web applications, the challenges involved in testing them are also multi fold [7]. To unearth a failure, it is necessary to test the web application in a combination of input with state. With its close linkage to the environment in which it is running, web application testing poses critical challenges [1]. However the running environment has a predominant effect on the nonfunctional requirements like availability, performance, compatibility, stability, accessibility, usability, security etc. [1], [7-9]. Specifically, the heterogeneity involved in the various languages used, execution environments, technologies and operating system, make the testing of web applications a critical issue to handle [7]. An efficient test suite should comprise of a those set of test cases which perform coverage testing of all possible combination of parameters [45]. Some of the major testing challenges with web applications are presented in Figure 2.

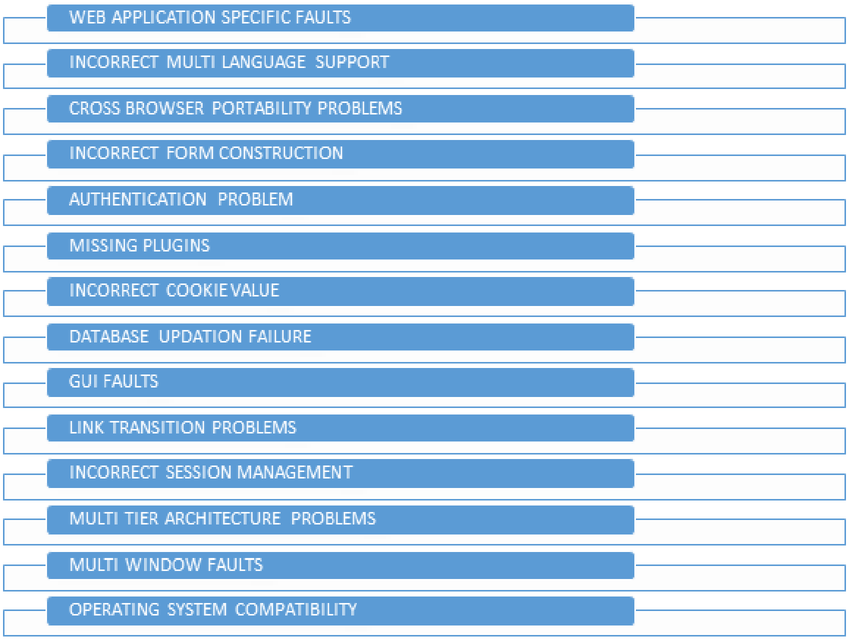
WEB APPLICATION SPECIFIC FAULTS
INCORRECT MULTI LANGUAGE SUPPORT
CROSS BROWSER PORTABILITY PROBLEMS
INCORRECT FORM CONSTRUCTION
AUTHENTICATION PROBLEM
MISSING PLUGINS
INCORRECT COOKIE VALUE
DATABASE UPDATION FAILURE
GUI FAULTS
LINK TRANSITION PROBLEMS
INCORRECT SESSION MANAGEMENT
MULTI TIER ARCHITECTURE PROBLEMS
MULTI WINDOW FAULTS
OPERATING SYSTEM COMPATIBILITY

Figure 2. Web Application Specific Faults

3. WEB BASED APPLICATION ARCHITECTURES

Paydar and Kahani proposed a framework which works on the principle that there should be a formal format for test specification which is understood by the executor module behaving like a web browser and performing HTTP based interactions. An automated test execution framework which is extendible has been developed. The framework is distributed supporting functional, security, load, stress and performance testing [10].

Kung presented a framework on BDI architecture i.e. the Beliefs, the Desires, and the Intentions Architecture. According to this, beliefs, desires and intentions are associated with the agents of the framework. All kinds of testing including the performance, functional, coverage, state and structural etc. are supported by this architecture [11].

Li et.al, proposed a model driven testing framework for testing web applications called MDWATP, which takes as input the model of the web application and generates test cases automatically. This framework is also useful to perform regression testing of the web applications [12].

Yang et. al., proposed an architecture which makes use of six subsystems namely Source Document Analysis Subsystem (SDAS), Test Management Subsystem (TMS), Test Development Subsystem (TDS), Test Execution Subsystem (TES), Test Failure Analysis Subsystem (TFAS), Test Measurement Subsystem (TMES) for performing the complete testing process including the activities extraction of control flow from source code, repository access interfaces, test execution, test validation, test failure analysis, and measurement of test coverage [13].

4. MODEL BASED TESTING OF WEB APPLICATIONS

A model based mutation testing where mutations are introduced into an a priori tested model of the web application using either a state chart diagram or an event sequence graph. Any seepage in the faults unearthed during the regular model based testing are exposed in the MBMT approach [14].

Mining workflow models has been proposed by Schur, Roth, and Zeller where a set of DOM trees are created in abundance keeping in view the incomplete set of execution traces. The additional DOM trees are created by keeping in view the application behavior [15].

A User Representation Model Graph (URMG) is constructed based on Customer Behavior Model Graph (CBMG) which is in turn constructed from a customer's access logs and automated tests are run on the web applications [16].

An object oriented testing model for testing web applications has been proposed wherein an Object Relationship Diagram(ORD) is devised embedding all the web application components including the relationships Navigation, Request, Response and Redirect using which the testers comprehend the structures and dependencies among the various web application components. The Page Navigation Diagram (PND) and the Object State Diagram (OSD) are used for performing navigation testing and structural testing of web applications [17].

A navigation model of the transition of pages is constructed based on the hyperlinks present in the web pages. Dynamic requests of users are modeled separately [18].

Realistic Usage Model (RUM) has been proposed which is used by a Simple Work Load Model (SLM). SLM relies on simulating the number of users by observing the server logs and studying the user requests [19].

5. WEB APPLICATION TESTING METHODOLOGY

Different types of testing techniques like coverage testing, structural testing, statistical testing, combinatorial interaction testing, penetration testing, Search based software engineering testing, Unique Input/Output method using Genetic Algorithms [20], Web application Slicing [21], [22], Hierarchical testing [23], Bypass testing, Cross Browser compatibility testing [24], Leveraging User session Data Testing have been presented by various researchers in the context of web application testing [37].

a. Structural Testing -Data flow analysis on web applications is performed and model for testing the application is built dynamically.
b. Statistical Testing – Input Sequence is generated to test the interactions with web applications based on the profile use of the web application.
c. Mutation Testing – The technique of introducing faulty code (called mutants) into the source code deliberately at predetermined points and testing the software to uncover any unknown errors. It is one of the effective coverage criterion techniques for testing of web applications [46].
d. Combinatorial Interaction Testing - Using a combination of different techniques by first designing a unique input space matrix for the web application
e. Penetration Testing - Automated tests which are run simulating the active attacks to expose the susceptibilities of the web applications.
f. Search Based Software Engineering Testing - Exploration of solutions within a state space and calculating a fitness function to the solution iteratively until we arrive at a most optimal solution. The technique is employed for branch coverage of web applications.
g. Using UIO and Genetic Algorithms - Path selection is done based on a unique input/output (UIO) algorithm and automatic test case generation using Genetic Algorithms which results in the best test sequences.
h. GUI Interaction Testing - GUI widgets events sequences are performed and the web application tested for correctness by observing the state of the GUI widgets.
i. Web Application Slicing - Reduced web application which behaves completely as the original one with respect to some criterion and performing the testing.
j. Cross Browser Compatibility testing - Subjecting web applications to deployment across different browsers for adherence to expected results.
k. Hierarchical Strategy - High level operational profile is developed enumerating frequency of use of operations and a high level function group to thoroughly test such an operation or related components is done.
l. Bypass Testing - Bypass client side checking by providing invalid inputs to web application to check correctness and security of the web application.
m. Leveraging User Session Data - Test cases are generated by applying strategies to collected user interactions in the form of URL's and name-value pairs.
n. Browser Fuzzing By Scheduled Mutation - Browsers are validated by using the static and dynamic ways, the former based on the input format while the latter randomly executing instructions giving one input at a time.
o. Invariant Based Technique - Testing the web application by crawling the web pages, and formally designing a state flow graph with all the possible user interaction sequences resulting in the possible user interface states.
p. Model Based Testing Technique - Web application is reduced to a state transition graph and navigation through links is tested to ascertain correct behavior of the web application.

A complete code coverage for any application assures thorough testing and higher probability of catching defects but the tester has to leverage upon the cost involved in complete code coverage verses the number of defects unearthed. A plethora of code coverage tools are available both open source and licensed for testing the web applications. A brief summary of all the various methods or techniques is presented in Table 1.

Table 1. Summary of Different Testing Methods/ Techniques available for Web Application Testing

S No	Testing Technique	Central Idea Behind the Technique	References
1	Structural Testing	Data flow analysis	[22], [25]
2	Statistical Testing	Interactions based on a profile use of a web application	[22]
3	Mutation Testing	Fault based testing	[14-31]
4	Combinatorial Interaction Testing	Combination of different techniques.	[32]
5	Penetration Testing	Active attack simulation	[33]
6	Search Based Software Engineering Testing	Branch Coverage of web applications	[34]
7	Using UIO and Genetic Algorithms	Path Selection using GA based algorithm	[20]
8	GUI Interaction Testing	State based testing of GUI widgets	[35]
9	Web Application Slicing	Testing on a slice of web application	[21], [22]
10	Cross Browser Compatibility Testing	Testing on different browsers	[24]
11	Hierarchical Strategy	Development of an operational profile	[23]
12	Bypass Testing	Security testing by avoiding client side validations	[36]
13	Leveraging User Session Data	URL testing	[37], [38]
14	Browser Fuzzing By Scheduled Mutation	Browser testing in static and dynamic ways	[39]
15	Invariant Based Technique	State based testing of interactions	[32]
16	Model Based Testing Technique	State based testing based on link navigation	[40]

A detailed study of each of the above mentioned techniques and their future research directions are consolidated in Table 2. The research directions highlighted in the table are directions in which the presenters of the respective work are heading towards.

Table 2. Advantages and Disadvantages of Various Testing Techniques

S. no.	Testing Technique	Future Research Directions	Reference No
1	Structural Testing	Moving towards development of new automated testing tools for increased efficiency.	[22], [25]
2	Statistical Testing	Automation to generate profiles of user accesses from the log file.	[22],
3	Mutation Testing	Optimization of test suites and a need for introduction of new mutation operators.	[26-31]
4	Combinatorial Interaction Testing	Expansion towards real time systems and try to automate the process of CIT model development.	[32]
5	Penetration Testing	Automation work for development of various tools	[33]
6	Search Based Software Engineering Testing	The technique needs to expand to cater to the languages supporting dynamic types and automation to simulate client side responses.	[34]
7	Using UIO and Genetic Algorithms	Scope for automation of input values which are manually provided by the tester.	[20]
8	GUI Interaction Testing	Varied test lengths to be augmented and more automation to come up with partition making automated.	[35]
9	Web Application Slicing	Automation to build completely automated regression testing using the slicing method along with an improvisation in the selection of test cases.	[21], [22]
10	Cross Browser Compatibility Testing	Development of a larger catalog of known DOM level differences between various browsers and automation for detecting differences between various browsers not listed at DOM level.	[24]
11	Hierarchical Strategy	Validation of approach by deploying it in the industry.	[23]
12	Bypass Testing	Development of some automated framework to develop automated bypass tests.	[36]
13	Leveraging User Session Data	Combination of traditional testing techniques and user-session data to be pursued further. Techniques for filtering user sessions and clustering algorithms can further be explored for taking the initial set of user sessions. Cost Effectiveness against traditional testing stands to be estimated and compared.	[37], [38]
14	Browser Fuzzing By Scheduled Mutation	Development of a reproducing mechanism for recording crash input as it is difficult to record persistent information for browser fuzzing. New Seeds and methods to achieve more crashes.	[39]
15	Invariant Based Technique	Work needs to progress on how to capture user session data and expanding it to larger applications.	[2]
16	Model Based Testing Technique	Need for the development of multiple tools and techniques to enhance adaptability.	[40]

6. TOOLS FOR WEB APPLICATION TESTING

A review of some of the automated testing tools and the type of testing supported by the tools led the survey to some interesting facts that there are not many tools available for testing the non-functional requirements of the web applications. The tools that have been studied as part of this survey are listed in Table 3. A comparative study of the tools is presented in Table 3.

Table 3. Comparison of Various Testing Tools for Web Application Testing

S. No.	Tool Name	Type of Testing Supported	Browser Support	Language Supported	Open Source/Licensed	References
1	WATIR	Functional Testing	IE, Chrome, Safari, Firefox	All	Open source	[41]
2	Selenium	Functional Testing	IE, Chrome, Safari, Firefox	Java, .NET, Ruby, Perl, PHP	Open source	[41], [42]
3	HP-QTP	Functional Testing	IE, Chrome, Safari, Firefox	VB Script	Licensed	[42]
4	Fitnesse	Acceptance Testing	N/A	Java, Python, C#,	Open source	[42]
5	testComplete	Functional Testing, Unit Testing	IE, Chrome, Safari, Firefox	VBScript, Jscript, Python, Delphi Script, C++ Script, C# Script	Licensed	[42]
6	Load Runner	Load Testing	Chrome, Safari, IE, Firefox	Java, .NET, JavaScript, HTML scripting	Licensed	[42]
7	Test Ng	Integration Testing, Functional Testing, End-End Testing, Unit Testing	N/A	Java	Open source	[42]
8	TOSCA	Functional Testing	IE, Firefox, Chrome	Delphi, .NET including WPF, Java Swing/SWT/AWT, VB	Licensed	[42]
9	SilkTest	Functional Testing	IE, Firefox	.NET, Java, Swing, SWT, DOM	Licensed	[42]
10	WinRunner	Functional Testing	Any Browser	Any web based application	Licensed	[42]
11	ApacheJMeter	Performance Testing, Load Testing	Any Browser	web service	Open source	[43]
12	NeoLoad	Load Testing	IE, Firefox, Chrome	ASP, .Net, J2EE, PHP	Licensed	[43]
13	LoadUI	Load Testing	Any Browser	Any web based application	Licensed	[52]
14	WebLoad	Load Testing	IE, Firefox, Chrome	HTTP/HTTPS (SSL, TLS), WebSocket, PUSH, AJAX, SOAP, HTML5, WebDAV and others.	Licensed	[43]
15	WAPT	Load Testing, Stress Testing	IE, Firefox, Chrome and others	Java Script	Licensed	[26]
16	Rational Performance Tester	Performance Testing	Any Browser	Any Script, XSS, SOAP	Licensed	[47]
17	Testing Anywhere	Functional Testing	IE, Firefox, Chrome	Any Web Based Application	Licensed	
18	Qengine	Functional Testing	IE, Mozilla, Firefox	VBScript, Jscript, Python, Delphi Script, C++ Script, C# Script	Licensed(but End- of Sale) announced	
19	MUTANDIS	Functional Testing	Any browser	Java Script	Open source	[27]
20	ATUSA	Functional Testing	Any browser	Ajax based any	Open source	

S. No.	Tool Name	Type of Testing Supported	Browser Support	Language Supported	Open Source/Licensed	References
				script crawling		
21	Crawljax	Navigation Testing	Any browser	Ajax based any script crawling	Open source	[2]
22	JSART	Regression Testing	Any Browser	Java Script based any web application	Open source	[48]
23	webMate	Regression Layout Testing	IE, Firefox, Chrome and others	VBScript, Jscript, Python, Delphi Script, C++ Script, C# Script	Licensed	[53]
24	reAjax	Functional Testing	Mozilla, Firefox	Ajax based scripts	Open source	[50]
25	WebVizor	Functional Testing	Any browser	Any Language	Open source	[51]
26	Web Portal In Container Testing	Integration Testing	Any browser	Any Script	Open source	
27	Veriweb Tool	Navigation Testing	Any browser	JavaScript	Open source	[49]
28	WebScarab	Security Testing	IE, Firefox, Chrome and others	Any Script, XSS, SOAP	Open source	[44]
29	Acunetix	Security Testing, Penetration Testing	Any browser	Any Script, XSS, SOAP	Licensed	[44]
30	Fortify	Security Testing	Any web browser	C#,.NET, Java, ASP	Licensed	[44]

7. RESULTS AND ANALYSIS

There is a need for the development and devise of a new metric to indicate the health of a web application. A plotting (considering only the tools under study) of the no. of tools vs the type of testing supported by each tool is presented in Figure 3. It is evident that there are not many tools testing the non-functional attributes of web applications like reliability, trustworthiness, and fault tolerance are not readily available. There is still however a dearth for open source tools using mutation analysis techniques which can perform automated test case execution based on mutation analysis while optimizing the test suite.

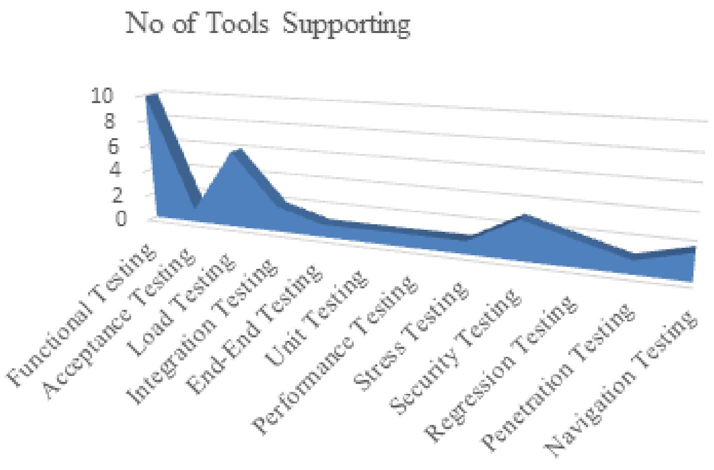

Figure 3. No of Tools Vs Type of Testing

The number of tools vs whether open source/ licensed has been plotted in Figure 4 which depicts that the topmost tools available for web application testing are mostly commercial in nature and there is a need to come up with a full-fledged open source tool for web application testing covering the various aspects of web application testing.

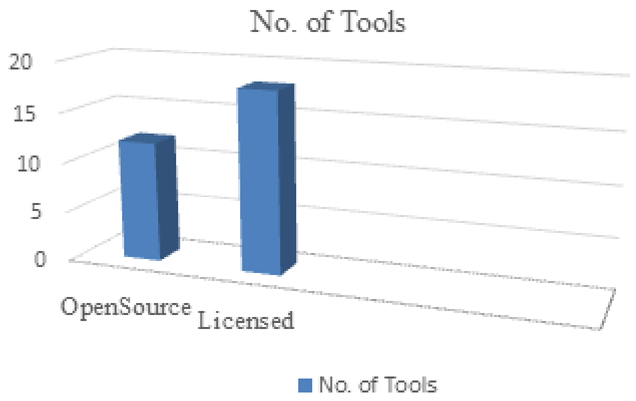

Figure 4. No. of Open Source Vs Licensed Tools

8. CONCLUSIONS

Tools for assessing the non-functional requirements in general are less where non-functional requirements actually play a key role in customer retention and popularity of the web application. Specifically non-functional requirements like usability, compatibility are absolutely in oblivion when it comes to testing. It is observed that with the web making a paradigm shift to the web services like SOAP, REST, and WSDL et.al. Compatibility testing of web applications is quite challenging and an area of growing concern where not much progress is seen in the earlier works. There is a dire need for a testing strategy for testing such non-functional requirements. Proposal of a metric to test the adequacy of test suite for exhaustive testing of a web application is the need of the hour. A metric to indicate the overall health of a web application needs to be proposed.

There is a need for development of a framework to support automated testing of a web application for the proposed strategy. With the entire web moving towards cloud based services, at the next level it is important to explore the testing of web applications on cloud. A panoramic view of testing web applications shows a need for a testing tool that can be deployed online in order to observe the dynamic behavior of web applications, as it is always difficult to catch such bugs as the web applications are often tested on standalone tools. The potential of testing techniques like mutation testing has not been completely tapped with testing web applications. There is a lot of scope to expand the horizons of mutation testing technique with web applications and unleashing its real potential. There is scope for the proposal and validation of some additional mutation operators pertaining to challenges like session management, cookie management where different languages like jsp, python can be explored to present more operators for mutating the web applications. Scope for optimization of regression test suites using appropriate techniques still exists.

ACKNOWLEDGEMENTS

The authors would like to thank the pioneers of research in testing of web applications who proposed several novel techniques and inspiring them to perform a survey and further pursue a direction for effective testing of web applications.

REFERENCES

[1] Shakti Kundu. “Web Testing: Tool, Challenges and Methods”, *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 2, No 3, March 2012. ISSN (Online): 1694-0814.

[2] Arora A., Sinha M. “Web Application Testing: A Review on Techniques, Tools and State of Art”, *International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February-2012 ISSN 2229-5518.*

[3] Qianxiang WANG, Lining QUAN, Fuchen YING, “Online Testing of Web-Based Applications”, *0730-3157/04 2004 IEEE.*

[4] Yu Qi, David Kung and Eric Wong, “Leveraging User-Session Data to Support Web Application Testing”, *IEEE Transactions on Software Engineering, vol. 31, no. 3, March 2005.*

[5] Moheb R. Girgis, Tarek M. Mahmoud, Bahgat A. Abdullatif, Alaa M. Zaki, “An Automated Web Application Testing System”, *International Journal of Computer Applications (0975 – 8887) Volume 99– No.7, August 2014.*

[6] M. Hicks. “*Paypal says sorry by waiving fees for a day”,* October 2004. Online:http://www.eweek.com/c/a/Web-Services-Web-20-and-SOA/PayPal-Says-Sorry-by-Waiving-Fees-for-a-Day. Posted 2004-10-27.

[7] Shivangi Kaushal, Jagpuneet Kaur Bajwa. “Analytical Review of User Perceived Testing Techniques”. *International Journal of Advanced Research in Computer Science and Software Engineering.* Volume 2, Issue 10, October 2012 ISSN: 2277 128X.

[8] Giuseppe Antonio Di Lucca, Anna Rita Fasolino, Francesco Faralli, Ugo De Carlini, *“Testing Web Applications”*, Proceedings of the International Conference on Software Maintenance (ICSM 02) 0-7695-1819-2/02 2002 IEEE.

[9] Giuseppe A. Di Lucca, *“Testing Web-Based Applications: the State of the Art and Future Trends”*, Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC’05) 0730-3157/05 2005 IEEE.

[10] Samad Paydar, Mohsen Kahani, “An Agent-Based Framework for Automated Testing of Web-Based Systems”, *Journal of Software Engineering and Applications, 2011, 4, 86-94*

[11] David Kung, *“An Agent-Based Framework for TestingWeb Applications”*, Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC’04) 0730-3157/04 2004 IEEE.

[12] Nuo Li, Qin-qin Ma, Ji Wu, Mao-zhong Jin, Chao Liu, *“A Framework of Model-Driven Web Application Testing”*, Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06) 0-7695-2655-1/06 2006.

[13] Ji-Tzay Yang, Jiun-Long Huang, Feng-Jian Wang, William. C. Chu, “An Object-Oriented Architecture Supporting Web Application Testing”, *0-7695-0368-3/99. 1999 IEEE.*

[14] Fevzi Bellia, Christof J. Budnikb, Axel Hollmannc, Tugkan Tuglulard, W. Eric Wonge, “Model Based Mutation Testing - Approach and Case Studies”, *Sci. Comput. Program. (2016),*

[15] Matthias Schur, Andreas Roth, and Andreas Zeller, “Mining Workflow Models from Web Applications”, 1184 *IEEE Transactions on Software Engineering,* vol. 41, no. 12, December 2015.

[16] Xiaolin Xu, Hai Jin, Song Wu, Lixiang Tang, and Yihong Wang, “URMG: Enhanced CBMG-Based Method for Automatically Testing Web Applications in the Cloud”, *Tsinghua Science and Technology ISSN 007-0214-07/10 pp65-75 Volume 19, Number 1, February 2014*.

[17] David C. Kung Chien-Hung Liu Pei Hsia, “An Object-Oriented Web Test Model for Testing Web Applications”, *0-7695-0793/00 2000 IEEE*.

[18] Filippo Ricca and Paolo Tonella, “Analysis and Testing of Web Applications”, *0-7695-1050-7/0 2001 IEEE.*

[19] Xingen Wang, Bo Zhou, Wei Li, “Model Based Load Testing of Web Applications”, *International Symposium on Parallel and Distributed Processing with Applications. 978-0-7695-4190-7/10 2010 IEEE.*

[20] Poonam Soni, Dr. Sanjay Tyagi, “Testing Web Applications Using UIO with GA”, *The International Journal of Soft Computing and Software Engineering* Volume 3, Issue 5, May 2013 ISSN: 2277 128X.

[21] Lei Xu, Baowen Xu, Zhenqiang Chen, Jixiang Jiang, Huowang Chen, *“Regression Testing for Web Applications Based on Slicing”*, Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC’03) 0730-3157/03 2003 IEEE.

[22] Filippo Ricca, *“Analysis, Testing and Re-structuring of Web Applications”*, Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM’04) 1063-6773/04 2004 IEEE.

[23] Jeff Tian, Li Ma, Zhao Li and A. G¨unes¸ Koru, *“A Hierarchical Strategy for Testing Web-Based Applications and Ensuring Their Reliability”*, Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC’03) 0730-3157/03 2003 IEEE.

[24] Ali Mesbah, Mukul R. Prasad, “Automated Cross-Browser Compatibility Testing”, ICSE ’11, May 21–28, 2011, Waikiki, Honolulu, HI, USA.

[25] Marcelo Medeiros Eler, Marcio Eduardo Delamaro, Jose Carlos Maldonado, Paulo Cesar Masiero, *“Built-in Structural Testing of web services”*, 2010 Brazilian Symposium on Software Engineering. 978-0-7695-4273-7/10 2010 IEEE.

[26] Rodolfo Adamshuk Silva, Simone do Rocio Senger de Souza,Paulo Sergio Lopes de Souza, “A Systematic Review on Search Based Mutation Testing”, *Information and Software Technology* (2016)

[27] Shabnam Mirshokraie, Ali Mesbah, and Karthik Pattabiraman, “Guided Mutation Testing for JavaScript Web Applications”. *IEEE Transactions on Software Engineering,* vol. 41, no. 5, May 2015.

[28] Juan P. Galeotti, Carlo A. Furia, Eva May, Gordon Fraser, Andreas Zeller, “Inferring Loop Invariants by Mutation, Dynamic Analysis, and Static Checking”, *IEEE Transactions on Software Engineering, vol. 41, no. 10, October 2015.*

[29] Fabiano Cutigi Ferrari, Awais Rashid, José Carlos Maldonado, “Towards the Practical Mutation Testing of Aspectj Programs”, *Science of Computer Programming 78 (2013) 1639–1662*.

[30] J.J. Domínguez-Jiménez, A. Estero-Botaro, A. García-Domínguez, I. Medina-Bulo, “Evolutionary Mutation Testing”, *Information and Software Technology 53 (2011) 1108–1123*.

[31] Sergio Segura, Robert M. Hierons, David Benavides, Antonio Ruiz-Cortés, “Mutation Testing on an Object-Oriented Framework: An Experience Report”, *Information and Software Technology 53 (2011) 1124–1136*.

[32] Cemal Yilmaz, Sandro Fouché, Myra B. Cohen, Adam Porter, Gulsen Demiroz, Ugur Koc, “Moving Forward with Combinatorial Interaction Testing”, *0018-9162/14© 2014 IEEE.*

[33] Nuno Antunes and Marco Vieira, “Penetration Testing for Web Services”, *0018-9162/14© 2014 IEEE*.

[34] Yuan-FangLi, Paramjit K.Das, DavidL.Dowe, “Two decades of Web application testing—A survey of recent advances”, *Information Systems, Elsevier 43 (2014) 20–54.0306-4379 & 2014*.

[35] Xun Yuan, Myra B. Cohen, and Atif M. Memon, “GUI Interaction Testing: Incorporating Event Context”, *IEEE Transactions on Software Engineering, Vol. 37, No. 4, July/August 2011*.

[36] Jeff Offutt, Ye Wu, Xiaochen Du and Hong Huang, "Web Application Bypass Testing", *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04) 0730-3157/04 2004 IEEE*.

[37] Sebastian Elbaum, Member, Gregg Rothermel, Srikanth Karre, Marc Fisher II. "Leveraging User-Session Data to Support Web Application Testing", *IEEE Transactions on Software Engineering, Vol. 31, No. 3, March 2005*.

[38] Sreedevi Sampath, Renée C. Bryce, "Improving the effectiveness of test suite reduction for user-session-based testing of web applications". *Information and Software Technology 54 (2012) 724–738.*

[39] Ying-Dar Lin, Feng-Ze Liao, Shih-Kun Huang, Yuan-Cheng Lai. "Browser Fuzzing by Scheduled Mutation and Generation of Document Object Models". The *49th Annual IEEE International Carnahan Conference on Security Technology*.

[40] Hani Achkar, "*Model Based Testing of Web Applications",* August 2010, STANZ.

[41] Nisha Gogna. "Study of Browser Based Automated Test Tools WATIR and Selenium". *International Journal of Information and Education Technology, Vol. 4, No. 4, August 2014*.

[42] Monika Sharma, Rigzin Angmo, "Web based Automation Testing and Tools". (IJCSIT) *International Journal of Computer Science and Information Technologies, Vol. 5 (1), 2014, 908-912. ISSN : 0975-9646*.

[43] J. Križani, A. Grguri, M. Mošmondor, P. Lazarevski. "Load testing and performance monitoring tools in use with AJAX based web applications", *MIPRO 2010, May 24-28, 2010, Opatija, Croatia*.

[44] LaShanda Dukes, Xiaohong Yuan, Francis Akowuah. "A Case Study on Web Application Security Testing with Tools and Manual Testing". *978-1-4799-0053-4/13 2013 IEEE*.

[45] Yu Wang, Hao Wu, Zhenyu Sheng, "A Prioritized Test Generation Method For Pair-wise Testing", *Indonesian Journal of Electrical Engineering, TELKOMNIKA*, Vol 11 No 1, 2013 pages 136-143

[46] Hanh Le Thi My, Nguyen Thanh, Khuat Thanh, "Survey on Mutation-based Test Data Generation", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol 5, No 5, 2015.

[47] Gary Denner, Jeremy Flynn, Maria Mitina, "Using IBM Rational Performance Tester V8.2 to load test IBM Lotus Notes Standard Client in a Citrix XenApp environment", Technical White Paper in IBM, July 2012.

[48] Shabnam Mirshokraie, Ali Mesbah *"JSART: JavaScript assertion-based regression testing"*, Proceedings of the 12th international conference on Web Engineering, July 2012.

[49] http://www.cs.utah.edu/~juliana/pub/veriweb-www2002.pdf as on 8/4/17.

[50] http://selab.fbk.eu/marchetto/tools/ajax/reAJAX as on 8/4/17.

[51] Sara Sprenkle, Holly Esquivel, Barbara Hazelwood, Lori Pollock, "WEBVIZOR: A Visualization Tool for Applying Automated Oracles and Analyzing Test Results of Web Applications",

[52] P.Saraswathi, M.MuraliKrishna, "A Web-based Approach for Automated Testing Applications", *International Journal of Emerging Technologies in Engineering Research (IJETER)*, Volume 4, Issue 11, November (2016).

[53] Valentin Dallmeier, Bernd Pohl, Martin Burger, Michael Mirold, Andreas Zeller, *"WebMate: Web Application Test Generation in the Real World"*, ICSTW '14 Proceedings of the 2014 IEEE International Conference on Software Testing, Verification, and Validation Workshops, Pages 413-418 March 31 - April 04, 2014.

BIOGRAPHIES OF AUTHORS

D. Rajya Lakshmi obtained her B.E in Electronics . and M.Tech in Computer Science and Engineering from Andhra University, India and Ph.D from JNTU Hyderabad, India. She is currently a Professor in the department of CSE at Jawaharlal Nehru Technological University Kakinada's constituent college, University college of Engineering , Narsaraopet. Her research interests include Computer Vision, Image Processing, Software Engineering, Data Mining, Network Security and Soft computing.

S. Suguna Mallika obtained her B.Tech in Computer science and Engineering from Nagarjuna University, India, M.Tech in Computer Science from JNTU Hyderabad, India, and currently pursuing her Ph.D in Computer Science and Engineering from JNTU Kakinada, India. She is currently working as an Associate Professor in the department of Computer Science and Engineering at CVR College of Engineering, Hyderabad, India. Her research interests are currently vested in the area of Software Engineering.