Survey of Testing Methods for Web Applications

Bilal Al-Ahmad[1] and Khaled Al Debei[2]

[1]Computer Information Systems Department, The University of Jordan, Aqaba, Jordan
[2]Information Technology Department, The University of Jordan, Aqaba, Jordan

Corresponding author:
Bilal Al-Ahmad
Email: b.alahmad@ju.edu.jo

Published: 24 December 2020

Abstract
Web applications are extensively used in our life aspects and domains, it plays a basic role in our routine daily operations. There is a critical need to ensure that web application is reliable, usable, and secure. Therefore, a great efforts has to be paid to improve the quality and reliability of such applications. Web testing is one of the main technical areas of software engineering. This study presents a survey about software testing methods for web applications. This survey provides a summary review to a number of several studies and techniques that done in testing of web applications. Also, it includes classification of these studies according to the testing type and their related contribution. In particular, this study mainly captures six different testing types for web applications like functional, security, usability, performance, compatibility, and structural. The contributions are classified as approaches, tools, models, frameworks, architectures, metrics, case studies, questionnaire, and prototypes.

Keywords: Software Engineering, Software Testing, Web Applications Testing, Survey.

INTRODUCTION

A web application is a program stored on a remote server and it can be accessed over the internet using HTTP protocol. Lately, web applications are used widely to support different sectors such as business, health, education, governmental agencies, e-commerce, and etc. Thus, web application has to be tested successfully to ensure that meet the necessary specification as explained in the study by Lakshmi and Mallika (2017).

According to software testing approaches introduced by Naik and Tripathy (2011), there are several essential reasons that made failures to web application such as the errors that related to client or server, configurations, network components, code, environment, and compatibility issues. Web testing involves of executing the application using combinations of input to show failures as in the study by García and Dueñas (2011). These failures are caused by faults in the running environment or in the web application itself. Web applications have huge number of users in all over the world and are difficult to test due to the diversity nature on different environments, technologies, hardware, software, and platforms as pointed by Di Lucca and Fasolino (2006).

With the current quick expansion of the Internet and web applications, the value to have reliable web application become necessary. Testing discover the majority of errors of the software. As a result, web applications should be tested wisely to make sure that the required functionality and features are satisfied by the applications. Thus, all components and features of web application must be tested successfully by applying effective testing methodologies and tools to check functionality, usability, performance, compatibility, security, and structure as in Figure 1. In addition, testing web applications process includes verification of different features related to availability, faults, severs performance, client performance, and the architecture of web application.

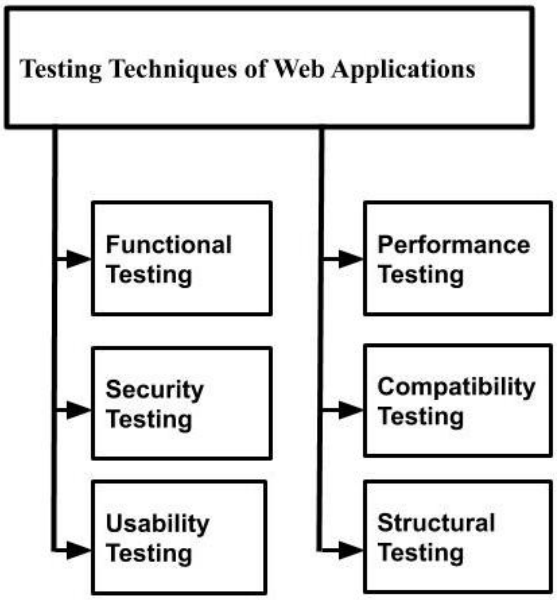

Figure 1. Testing Techniques of Web Applications

The functional testing on web application include checking the essential actions that web application must do as pointed by Di Lucca and Fasolino (2006). Each single function that is provided by the website need to be tested as a black box testing as in Arora and Sinha (2012), functional testing ensures text, link, form, cookies, database, image, hyperlink, script language, and design language as

in Yu (2019). Security testing aims to validate the effectiveness of the overall web system strengths against unwanted access of illegal users, as well as their ability to protect system resources from improper uses and to grant the authorized access to authorized service for legal users. Also, system vulnerabilities affecting the security aspects of web application may be contained in the application code, or in any of the different hardware, software, middle-ware components of the systems. This type of testing checks basic login and storage information such user login, session, log file, encryption, and directory setting as described by Yu (2019).

Usability testing aims to inspect to what degree an application is easy to use. Usability testing mainly focuses on testing the user interface, GUI, and navigation. Usability is a critical issue for a web application. When Web applications usability testing is executed, concerns about the completeness, correctness, and navigation along application are to be considered. Usability testing assesses whether web page design, overall layout, and navigation are user friendly as in Yu (2019). Performance testing objective aims to verify system performances such response time, service availability, load, stress, and connection speed as in Yu (2019). For web applications, system performance is a critical issue since web users expect quick response to their requests and the availability of web services by analyzing data from access log. The performance testing comprises load test, stress test, and link speed test.

Compatibility testing verifies that the applications can be run on different machines that are used by users. various operating systems Compatibility testing will have to discover failures due to the usage of server platforms, client browsers, and various operating systems. As a result, some compatibility failures that related to the application and the running environment are responsible for compatibility failures which affect the reliability of web applications. In addition, test the compatibility of browser and operating system on different platforms. Structural testing of web applications concentrates on data flow testing technology as proposed by Xiaoli Lu and Yunwei Dong (2019). Based on this structure model, the design techniques of testing cases to support the structure test of web application software. This testing type is considered as white-box testing execution. The Structural testing has many models to capture various types of data flow information of a web applications such control flow graph, inter-procedural control flow graph, object control flow graph, and composite control flow graph as proposed by Di Lucca and Fasolino (2006).

This study provides a survey on number of studies which focused on web applications testing. Also, it presents a classification of these studies according to what is the type of testing techniques. In addition, the survey shows the exact contribution for each study.

MATERIALS AND METHODOLOGY

This study presents the method of doing this survey and how it is broken down as well as how we classified the studies and the approaches. First of all, the survey includes studies that have been collected from the years between 1996-2020. This paper surveys the contributions for six main testing types as illustrated in Figure 2. The contributions classified into models, tools, approaches, frameworks, architectures, and metrics. The main purpose of this study is to present a brief survey on the testing techniques of web applications. The survey concisely covers six essential types of testing techniques: functional, security, usability, performance, compatibility, and structural.

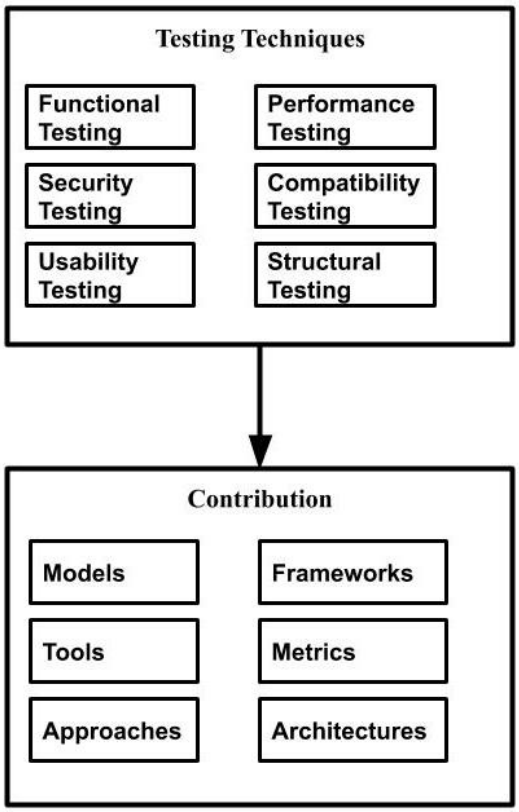

Figure 2. Mapping between Testing Techniques and contributions

According to the literature, many researches have been done on testing of a web application and a lot of models and tools have been proposed for this purpose. The functional testing is one of the testing types that may be performed on web applications. It is responsible for testing a functionality of each feature on a website and discovering failures in required services. The features include user interface, database, links, cookies, and etc.

Table 1: Papers published in the functional testing of web applications with its contribution

Researchers/Years	**Contribution**
Di Lucca et al. (2002)	Model
Elbaum et al. (2003)	Model
Benedikt et al. (2002)	Tool
Marchetto et al. (2008)	Approach
Di Lucca et al. (2003)	Model
Andrews et al. (2005)	Model
Stepien et al. (2008)	Approach
Ricca and Tonella (2001)	Model
Offutt et al. (2004)	Approach
Sinha and Paradkar (2006)	Model
Qian et al. (2007)	Model
Sampath et al. (2007)	Model
Gamma and Beck (2005)	Tool

Gold (2003)	Tool
Ricca and Tonella (2005)	Model
Sneed and Huang (2006)	Tool
Marchetto and Tonella (2009)	Approach
Dallmeier et al. (2012)	Tool
Benedikt et al.(2002)	Tool
Li et al. (2006)	Approach
Peng and Lu (2011)	Approach
Artzi et al. (2011)	Tool
Mesbah and Prasad (2011)	Tool
Liu and Tan (2008)	Tool
Niese et al. (2002)	Tool
De Jesus et al. (2015)	Approach
Wang et al. (2012)	Framework
Nagowah et al. (2012)	Tool
Razak and Fahrurazi (2011)	Tool
Cohen et al. (1997)	Approach
García and Dueñas (2011)	Tool
Xiong and Probert (2003)	Architecture
Tian et al. (2003)	Architecture
Becker (2002)	Model
Stocco et al. (2017)	Tool
Jan et al. (2016)	Tool
De Moura et al. (2017)	Tool
Hillah et al. (2017)	Tool
Nguyen et al. (2019)	Metrics
Jia and Liu (2002)	Tool
Gao et al. (1997)	Approach
Sampath et al. (2004)	Tool

Table 1 presents some researches that have addressed the functional testing of a web application and the contribution of each article. As shown in Table 1, many tools and approaches have been presented for functional testing. Some of these researches, such as (Benedikt et al. (2002), Benedikt et al.(2002), Gold (2003), Sampath et al. (2004), Gamma and Beck (2005), Sneed and Huang (2006), Stocco et al. (2017) and De Moura et al. (2017)) have proposed tools for detecting functional problems and automating the testing activity. The researches (Stepien et al. (2008), Offutt et al. (2004), Marchetto and Tonella (2009), Li et al. (2006), Peng and Lu (2011), De Jesus et al. (2015), Cohen et al. (1997), and Gao et al. (1997)) have presented seveal approaches for state-based, task based testing, and etc. Besides that, the studies by (Di Lucca et al. (2002), Elbaum et al. (2003), Di Lucca et al. (2003), Andrews et al. (2005), Ricca and Tonella (2001), Sinha and Paradkar (2006), Qian et al. (2007), Sampath et al. (2007), Ricca and Tonella (2005), and Becker (2002)) proposed their contribution as testing models of different aspects for the web functionalities. Moreover, few studies conducted by (Xiong and Probert (2003), and Tian et al. (2003)) contributed their efforts as an architecture to improve the functional testing.

Table 2: Papers published in the security testing of web applications with its contribution

Researchers/Years	**Contribution**
Jan et al. (2019)	Approach
Tappenden et al. (2005)	Architecture
Tian-yang et al. (2010)	Tool
Felderer et al. (2016)	Approach
Büchler et al. (2012)	Model
Huang et al. (2005)	Framework
Avancini and Ceccato (2011)	Approach
Curphey and Arawo (2006)	Tool
Armando et al. (2010)	Approach
Haixia and Zhihong (2009)	Model
Wassermann and Su (2004)	Framework
Thomé et al. (2014)	Tool
Salas and Martins (2014)	Tool
Bezemer et al. (2009)	Approach
Fonseca et al. (2009)	Tool
Li et al. (2010)	Tool
Xiong and Peyton (2010)	Framework
Pellegrino and Balzarotti (2014)	Tool
Sudhodanan et al. (2016)	Tool
Petukhov and Kozlov (2008)	Tool
Dukes et al. (2013)	Case Study
Mainka et al. (2012)	Tool
Fonseca et al. (2013)	Tool
Bozic et al.(2015)	Approach
Artzi et al. (2011)	Tool
Marback et al. (2009)	Tool
Zech et al.(2019)	Tool
Bozic and Wotawa (2018)	Approach
Rouaix (1996)	Tool
Liu and Richardson (2000)	Framework
Adams and Armstrong (1998)	Tool

Security testing is another essential type of web application method. The security testing is a very important step that should be accomplished in a website, especially in e-commerce website, which includes transferring money. The main objective of the security testing is to confirm the effectiveness of a web application and protect it from undesirable users. Security testing needs to be examined in all parts of a web application, such as application code, hardware and software.

Table 2 presents some researches that have addressed the security testing and the contribution of each research. As displayed in Table 2, many approaches, tools, frameworks, and models have been presented for security testing. Some of these researches, such as (Tian-yang et al. (2010), Curphey and Arawo (2006),Thomé et al. (2014), Salas and Martins (2014), Fonseca et al. (2009), Li et al. (2010),

Pellegrino and Balzarotti (2014), Sudhodanan et al. (2016), Petukhov and Kozlov (2008), Mainka et al. (2012), Fonseca et al. (2013), Artzi et al. (2011), Marback et al. (2009), Zech et al.(2019), Rouaix (1996), and Adams and Armstrong (1998)) have proposed tools to conduct security testing. In particular to framework contributions, the research studies (Huang et al. (2005), Liu and Richardson (2000), Wassermann and Su (2004), and Xiong and Peyton (2010)) have developed framework to analyze and assess security testing. Additionally, other studies (Jan et al. (2019), Felderer et al. (2016),Avancini and Ceccato (2011), Armando et al. (2010), Bezemer et al. (2009), Bozic et al.(2015), and, Bozic and Wotawa (2018)) proposed multiple approaches for security testing to ensures the key aspects of quality of a web application to avoid the security failures and vulnerabilities through detecting potential attacks as possible. On the other hand, few studies as Büchler et al. (2012), and Haixia and Zhihong (2009) proposed security models for automating security testing, and modeling database security testing for web applications. Few studies by Tappenden et al. (2005), and Dukes et al. (2013) outlined their efforts as architecture and case study respectively.

Table 3: Papers published in the usability testing of web applications with its contribution

Researchers/Years	**Contribution**
Halfond and Orso (2007)	Approach
Halfond et al. (2009)	Approach
Jensen et al. (2013)	Tool
Al-Zain (2012)	Prototype
Bellettini et al. (2005)	Tool
Spool and Schroeder (2001)	Theory
Hong et al. (2001)	Tool
Atterer et al. (2006)	Approach
Tullis et al. (2002)	Approach
Robal et al. (2017)	Framework
Grigera et al. (2017)	Tool
Paz et al. (2015)	Approach
Roy et al. (2017)	Questionnaire
Harrati et al. (2015)	Approach
Mvungi and Tossy (2015)	Metrics
Menzi-Cetin et al. (2017)	Approach
Gupta and Ahlawat (2016)	Approach
Cemellini et al. (2018)	Approach
Kaur and Sharma (2018)	Approach
Scholtz et al. (1998)	Tool
McCreight et al. (2019)	Approach
Babatunde et al. (2020)	Model
Sukmasetya et al. (2020)	Questionnaire
Oliha (2020)	Framework
Marien et al. (2019)	Prototype
Osada et al. (2020)	Tool

One of the web application testing that concerns with the way users interact with a web application is usability testing. Usability testing is critical for the success of a web application because

it purely measures user habits and expectations of that application. It focuses on testing a user interface, effectiveness and ease of use of the web application. Many researchers have conducted usability testing in their work. Table 3 presents some researches that have worked on usability testing and the contribution of each research. As shown in the above table, some works (Roy et al. (2017) and Sukmasetya et al. (2020)) have presented a questionnaire-based usability testing covering aspects of usefulness, satisfaction and ease of use. Other works by (Jensen et al. (2013), Bellettini et al. (2005), Hong et al. (2001), Grigera et al. (2017), Scholtz et al. (1998), and Osada et al. (2020)) have focused on presenting tools for automating usability testing. Moreover, the studies conducted by (Halfond and Orso (2007), Halfond et al. (2009),Atterer et al. (2006), Tullis et al. (2002), Paz et al. (2015), Harrati et al. (2015), Menzi-Cetin et al. (2017), Gupta and Ahlawat (2016), Gupta and Ahlawat (2016), Cemellini et al. (2018), Kaur and Sharma (2018), and McCreight et al. (2019)) represented their efforts as approaches to improve the usability testing for web applications. Few research introduced usability testing prototype as in Al-Zain (2012) and Marien et al. (2019). Nonetheless, it is not easy to find a tool, approach or a model that can measure usability testing of a whole web application since there is no standardized way of measuring it.

Table 4: Papers published in the performance testing of web applications with its contributions

Researchers/Years	**Contribution**
Križanić et al. (2010)	Tool
Draheim et al. (2006)	Approach
Hamed and Kafri (2009)	Approach
Pradeep and Sharma (2019)	Tool
Menascé (2002)	Framework
Subraya and Subrahmanya (2000)	Approach
Shams et al. (2006)	Model
Wu and Wang (2010)	Framework
Kiran et al. (2015)	Architecture
Mukherjee et al. (2014)	Approach
Khan and Amjad (2016)	Approach
Kao et al. (2013)	Framework
Ahmad et al. (2018)	Model
Draheim et al. (2006)	Approach
Ali and Badr (2015)	Framework
Pradeep and Sharma (2019)	Evaluation
Denaro et al.(2004)	Approach
Zhu et al.(2010)	Approach
Bernardino et al.(2016)	Model
Kalita and Bezboruah (2011)	Approach
Gao et al.(2010)	Framework
Jiang and Jiang (2009)	Model
He et al. (2019)	Approach
Shaw (2000)	Case Study
Hamed and Kafri (2009)	Architecture
Abbors et al. (2012)	Tool

Lutteroth and Weber (2008)	Model
Dhote and Sarate (2012)	Approach
Zhou et al. (2014)	Model
Adamoli et al. (2011)	Tool
Bozdag et al. (2009)	Framework
Barna et al. (2011)	Model
Krishnamurthy et al. (2010)	Model
Wang et al. (2013)	Model
Rodrigues et al. (2015)	Tool
Avritzer and Weyuker (2004)	Model
Proko and Ninka (2013)	Case study
Huerta-Guevara et al. (2019)	Evaluation
Chen et al.(2008)	Framework
Da Silveira et al.(2011)	Model
Schieferdecker et al.(1997)	Approach
Yang et al. (1998)	Tool
Ahmad (2020)	Case study
Shivakumar (2020)	Framework
Silva and Lopes (2020)	Model
Mahmood and Sirshar (2017)	Tool
Wu and Zhu (2017)	Tool

Performance testing focuses on verifying specific web application performance. The performance of a web application can be represented by many features such as a response time, service availability, reliability and stability. The main objective of performance testing is to identify and eliminate the performance deterioration in a web application. Table 4 presents some works that have been done on performance testing. All of these studies use a variety of performance testing tools, depending on its needs and preferences.

The works done by (Križanić et al. (2010), Pradeep and Sharma (2019), Abbors et al. (2012), Adamoli et al. (2011),Rodrigues et al. (2015),Yang et al. (1998), Mahmood and Sirshar (2017), and, Wu and Zhu (2017)) have proposed tools for performance testing that can get accurate performance results. Some other works (Menascé (2002), Wu and Wang (2010), Kao et al. (2013), Ali and Badr (2015),Gao et al.(2010),Bozdag et al. (2009),Chen et al.(2008), and Shivakumar (2020)) provide their contributions as frameworks of performance testing for different web applications for meeting the more concurrent users. Furthermore, many other studies (Draheim et al. (2006), Subraya and Subrahmanya (2000),Mukherjee et al. (2014), Khan and Amjad (2016),Draheim et al. (2006),Denaro et al.(2004), Zhu et al.(2010), Kalita and Bezboruah (2011), He et al. (2019), Dhote and Sarate (2012), and Schieferdecker et al.(1997)) have developed approaches for performance testing to capture the essential perspectives of load, stress, connection speed, response time, and service availability. Few studies (Kiran et al. (2015), and, Hamed and Kafri (2009)) introduced performance architectures to. Similarly, the studies by Pradeep and Sharma (2019), and Huerta-Guevara et al. (2019) showed an evaluation to measure and assess the performance criteria's for testing of web applications. In terms of case study contribution, the studies by Shaw (2000), Proko and Ninka (2013), and Ahmad (2020) introduced some case studies to highlight the main issues related to performance testing. Also, several

performance models conducted by the studies (Shams et al. (2006), Ahmad et al. (2018), Bernardino et al.(2016), Jiang and Jiang (2009), Lutteroth and Weber (2008), Zhou et al. (2014), Barna et al. (2011), Krishnamurthy et al. (2010), Wang et al. (2013), Avritzer and Weyuker (2004), Da Silveira et al.(2011), and Silva and Lopes (2020)) to put more efforts on this critical type of testing techniques.

Table 5: Papers published in the Compatibility testing of web applications with its contributions

Researchers/Years	**Contribution**
Mesbah and Prasad (2011)	Tool
Xu et al. (2003)	Tool
Motwani et al. (2015)	Framework
Chen et al. (2018)	Tool
Li and Zeng (2014)	Model
Tanaka (2019)	Tool
Shi and Zeng (2015)	Model
Minamide and Mori (2012)	Model
Saqib and Shahzad (2018)	Model
Poberežnik (2013)	Approach
Kaalra and Gowthaman (2014)	Framework
Li and Chou (2015)	Approach
Gaardboe et al. (2017)	Model
Cheng et al. (2017)	Approach
Balasundaram et al. (2012)	Approach
Chen et al. (2018)	Approach
Gao and Miao (2013)	Model
Kumar and Arya (2016)	Approach
Manhas and Sarwar (2011)	Evaluation
Agbozo and Spassov (2018)	Approach

Compatibility testing ensures that an application is compatible with different computing environments, web application requires to be tested for compatibility on several browsers, browsers versions, and operating system versions. Table 5 presents some researches that have done on compatibility testing and the contribution of each research. As shown in Table 5, some researches have focused on developing tools for compatibility testing. For example, the work of (Mesbah and Prasad (2011), Xu et al. (2003), Chen et al. (2018), and Tanaka (2019)) have presented an automated solution for cross-browser compatibility testing of modern web applications. Some other researches have concentrated on presenting models, approaches or frameworks. For example, the studies by (Liu et al. (2000), Liu et al. (2001), Liu (2006), Liu et al. (2008), Endo et al. (2010), Qi et al. (2005), Elbaum et al. (2003), Wang et al. (2008), and Ricca and Tonella (2001)) proposed several approaches to develop efficient methods for compatibility verification of web applications. Moreover, the researchers (Li and Zeng (2014), Shi and Zeng (2015), Minamide and Mori (2012), Saqib and Shahzad (2018), Gaardboe et al. (2017), and Gao and Miao (2013)) presented testing models to capture configuration and browser compatibility issues which help to get high quality we applications. In perspective to framework contributions, the studies (Motwani et al. (2015), and, Kaalra and Gowthaman (2014)) introduced framework for browser compatibility testing of web applications.

Nevertheless, due to the large diversity of possible combinations of all the components involved in the execution of a web application, it is not practical to test all of them. Therefore, it is usually that only the most common combinations are taken into consideration and tested. This means that an only part of possible compatibility failures might be exposed.

Table 6: Papers published in the structural testing of web applications with its contributions

Researchers/Years	**Contribution**
Liu et al. (2000)	Approach
Liu et al. (2000)	Model
Kung et al. (2000)	Methodology
Liu et al. (2001)	Approach
Liu (2006)	Approach
Yang et al. (1998)	Tool
Endo et al. (2008)	Tool
Liu et al. (2008)	Approach
Eler et al. (2010)	Tool
Zheng and Chen (2007)	Model
Praphamontripong and Offutt (2010)	Tool
Artzi et al. (2011)	Framework
Ricca (2004)	Model
Endo et al. (2010)	Approach
Qian et al. (2007)	Model
Karam et al. (2006)	Model
Bai et al. (2006)	Framework
Qi et al. (2005)	Approach
Carniello et al. (2005)	Tool
Elbaum et al. (2003)	Approach
Dubey et al. (2012)	Tool
Sampath and Bryce (2012)	metrics
Wang et al. (2008)	Approach
Romli et al. (2020)	Tool
Olugbara and Letseka (2020)	Model
Ricca and Tonella (2001)	Approach

Structural testing concerns with testing the structure of code, It is also sort of white box testing. The structural testing is based on the knowledge about the system being tested. It focuses on how system does rather than the functionality of the system. Table 6 presents works that have addressed the structural testing. As shown in Table 6, many tools, model, approaches and frameworks have been developed to handle structural testing. For example, Ricca and Tonella (2001) have proposed a first approach for structural testing of primarily static web applications by focusing on HTML pages. Also, several approaches presented by the studies by (Liu et al. (2000), Liu et al. (2001), Liu (2006), Liu et al. (2008), Endo et al.(2010), Qi et al. (2005), Elbaum et al. (2003), and Wang et al. (2008)) to capture different aspects of the design and code of web applications such as data flow and control flow. Too, Several studies as in (Yang et al. (1998), Endo et al. (2008), Eler et al. (2010), Praphamontripong and

Offutt (2010), Carniello et al. (2005), Dubey et al. (2012), and Romli et al. (2020)) proposed various tools to automate the white box testing in order to get high quality web structure. Furthermore, some models provided by the studies (Liu et al. (2000), Zheng and Chen (2007), Ricca (2004), Qian et al. (2007), Karam et al. (2006), and Olugbara and Letseka (2020)) to improve the overall web structure. The work done by Artzi et al. (2011) and Bai et al. (2006) outlines structural framework for web applications by using Javascript language and multi agents respectively. The study by Sampath and Bryce (2012) contributes some structural metrics for web applications.

RESULTS AND DISCUSSION

Web applications have many challenges involved failures introduced by Kaushal and Bajwa (2012) as well as failures related to web application and running environment as in Kundu (2012). The running environment has a main effect on the nonfunctional requirements like availability, performance, compatibility, stability, accessibility, usability, security as in Kaushal and Bajwa (2012), Kundu (2012), Di Lucca (2002), Di Lucca and Fasolino (2006). Also, the difference involved in the various languages used, execution environments, technologies and operating system, make the testing of web applications a critical issue to handle as in Kaushal and Bajwa (2012). As a result, an efficient test suite should contain of a those set of test cases which perform coverage testing of all possible combination of parameters as proposed by Wang et al. (2013) .

From our point of view the most common testing techniques of web applications are functional, security, usability, performance, compatibility, and structural. Testing of web applications consists of six contributions type indicated as models, tools, approaches, frameworks, metrics, and architectures. By analyzing and studying above testing types , we conclude that testing of functionalities, security, performance, and usability are the most important techniques to ensure high quality web applications.

This survey classifies six main contributions in each testing type like models, tools, approaches, frameworks, metrics, and architectures. The most contributions related to functional testing are models and tools. Furthermore, the main contributions created in security testing are tools and approaches. In particular to usability testing, the key contribution is usability approaches. Moreover, the essential contributions for performance testing are approaches and tools. With regarding to compatibility testing, the principal contribution is model. Moreover, according to reviews in our survey, the main contributions in structural testing are approaches and models. On the other hand, the least contributions are metrics, case studies, questionnaires, and prototypes across all the testing types for web applications.

CONCLUSION

This study presents a brief survey of testing techniques for web applications. The survey is built based on a various of previous studies that have done in this particular field. This survey classifies these studies according to the testing types and their related contribution. Our study mainly captures six different testing types for web applications: functional, security, usability, performance, compatibility, and structural. It also concludes the main contribution for all testing types are approaches, tools, and models.

REFERENCES

[1] García, B., & Dueñas, J. C. (2011). Automated functional testing based on the navigation of web applications. arXiv preprint arXiv:1108.2357.

[2] Di Lucca, G. A., & Fasolino, A. R. (2006). Testing Web-based applications: The state of the art and future trends. Information and Software Technology, 48(12), 1172-1186.

[3] Lakshmi, D. R., & Mallika, S. S. (2017). A review on web application testing and its current research directions. *International Journal of Electrical and Computer Engineering*, *7*(4), 2132.

[4 Kundu, S. (2012). Web testing: tool, challenges and methods. *IJCSI International Journal of Computer Science Issues*, *9*(2), 1694-0814.

[5] Naik, K., & Tripathy, P. (2011). *Software testing and quality assurance: theory and practice*. John Wiley & Sons.

[6] Arora, A., & Sinha, M. (2012). Web application testing: A review on techniques, tools and state of art. *International Journal of Scientific & Engineering Research*, *3*(2), 1.

[7] Di Lucca, G. A., & Fasolino, A. R. (2006). Web application testing. In *Web Engineering* (pp. 219-260). Springer, Berlin, Heidelberg.

[8] Xu, H., Su, C., Wu, S., & Tang, D. (2019). Survey of Testing Methods of O2O Catering Platform. In Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference (pp. 220-224).

[9] Yu, J. (2019). Exploration on Web Testing of Website. In Journal of Physics: Conference Series (Vol. 1176, No. 2, p. 022042). IOP Publishing.

[10] Kaushal, S., & Bajwa, J. K. (2012). Analytical review of user perceived testing techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 2(10).

[11] Di Lucca, G. A., Fasolino, A. R., Faralli, F., & De Carlini, U. (2002). Testing web applications. In International Conference on Software Maintenance, 2002. Proceedings. (pp. 310-319). IEEE.

[12] Wang, Y., Wu, H., & Sheng, Z. (2013). A prioritized test generation method for pair-wise testing. vol, 11, 136-143.

[13] Jia, X., & Liu, H. (2002). Rigorous and automatic testing of web applications. In *Proceedings of the 6th IASTED International Conference on Software Engineering and Applications (SEA 2002)* (pp. 280-285).

[14] Gao, J.Z., Chen, C., Toyoshima, Y. and Leung, D.K., (1997). Developing an integrated testing environment using the world wide web technology. In *Proceedings Twenty-First Annual International Computer Software and Applications Conference (COMPSAC'97)* (pp. 594-601). IEEE.

[15] Sampath, S., Mihaylov, V., Souter, A. and Pollock, L., (2004). A scalable approach to user-session based testing of web applications through concept analysis. In *Proceedings of the 19th IEEE international conference on Automated software engineering* (pp. 132-141). IEEE Computer Society.

[16] Andrews, A. A., Offutt, J., & Alexander, R. T. (2005). Testing web applications by modeling with FSMs. Software & Systems Modeling, 4(3), 326-345.

[17] Artzi, S., Dolby, J., Jensen, S. H., Møller, A., & Tip, F. (2011). A framework for automated testing of javascript web applications. In Proceedings of the 33rd International Conference on Software Engineering (pp. 571-580).

[18] Dallmeier, V., Burger, M., Orth, T., & Zeller, A. (2012). Webmate: a tool for testing web 2.0 applications. In Proceedings of the Workshop on JavaScript Tools (pp. 11-15).

[19] de Jesus, F. R., de Vasconcelos, L. G., & Baldochi, L. A. (2015). Leveraging task-based data to support functional testing of web applications. In Proceedings of the 30th Annual ACM Symposium on Applied Computing (pp. 783-790).

[20] de Moura, J. L., Charão, A. S., Lima, J. C. D., & de Oliveira Stein, B. (2017). Test case generation from BPMN models for automated testing of Web-based BPM applications. In 2017 17th International Conference on Computational Science and Its Applications (ICCSA) (pp. 1-7). IEEE.

[21] Di Lucca, G. A., & Di Penta, M. (2003). Considering browser interaction in web application testing. In Fifth IEEE International Workshop on Web Site Evolution, 2003. Theme: Architecture. Proceedings. (pp. 74-81). IEEE.

[22] Elbaum, S., Karre, S., & Rothermel, G. (2003). Improving web application testing with user session data. In 25th International Conference on Software Engineering, 2003. Proceedings. (pp. 49-59). IEEE.

[23] Hillah, L. M., Maesano, A. P., De Rosa, F., Kordon, F., Wuillemin, P. H., Fontanelli, R., ... & Maesano, L. (2017). Automation and intelligent scheduling of distributed system functional testing. International Journal on Software Tools for Technology Transfer, 19(3), 281-308.

[24] Jan, S., Nguyen, C. D., & Briand, L. C. (2016). Automated and effective testing of web services for XML injection attacks. In Proceedings of the 25th International Symposium on Software Testing and Analysis (pp. 12-23).

[25] Jia, X., & Liu, H. (2002). Rigorous and automatic testing of web applications. In Proceedings of the 6th IASTED International Conference on Software Engineering and Applications (SEA 2002) (pp. 280-285).

[26] Jun, W., & Meng, F. (2011). Software testing based on cloud computing. In 2011 International Conference on Internet Computing and Information Services (pp. 176-178). IEEE.

[27] Li, N., Ma, Q. Q., Wu, J., Jin, M. Z., & Liu, C. (2006). A framework of model-driven web application testing. In 30th Annual International Computer Software and Applications Conference (COMPSAC'06) (Vol. 2, pp. 157-162). IEEE.

[28] Marchetto, A., & Tonella, P. (2009). Search-based testing of Ajax web applications. In 2009 1st international symposium on search based software engineering (pp. 3-12). IEEE.

[29] Marchetto, A., Tonella, P., & Ricca, F. (2008). State-based testing of Ajax web applications. In 2008 1st International Conference on Software Testing, Verification, and Validation (pp. 121-130). IEEE.

[30] Nguyen, H. V., Phan, H. D., Kästner, C., & Nguyen, T. N. (2019). Exploring output-based coverage for testing PHP web applications. Automated Software Engineering, 26(1), 59-85.

[31] Niese, O., Margaria, T., & Steffen, B. (2002). Automated functional testing of web-based applications. in 5th Intern. Quality Week Europe.

[32] Offutt, J., Wu, Y., Du, X., & Huang, H. (2004). Bypass testing of web applications. In 15th International Symposium on Software Reliability Engineering (pp. 187-197). IEEE.

[33] Peng, X., & Lu, L. (2011). A new approach for session-based test case generation by GA. In 2011 IEEE 3rd International Conference on Communication Software and Networks (pp. 91-96). IEEE.

[34] Qian, Z., Miao, H., & Zeng, H. (2007). A practical web testing model for web application testing. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 434-441). IEEE.

[35] Razak, R. A., & Fahrurazi, F. R. (2011). Agile testing with Selenium. In 2011 Malaysian Conference in Software Engineering (pp. 217-219). IEEE.

[36] Ricca, F., & Tonella, P. (2005). Web testing: a roadmap for the empirical research. In Seventh IEEE International Symposium on Web Site Evolution (pp. 63-70). IEEE.

[37] Ricca, F., & Tonella, P. (2001). Analysis and testing of web applications. In Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001 (pp. 25-34). IEEE.

[38] Sampath, S., Mihaylov, V., Souter, A., & Pollock, L. (2004). A scalable approach to user-session based testing of web applications through concept analysis. In Proceedings. 19th International Conference on Automated Software Engineering, 2004. (pp. 132-141). IEEE.

[39] Sampath, S., Sprenkle, S., Gibson, E., Pollock, L., & Greenwald, A. S. (2007). Applying concept analysis to user-session-based testing of web applications. IEEE Transactions on Software Engineering, 33(10), 643-658.

[40] Sharma, M., & Angmo, R. (2014). Web based automation testing and tools. International Journal of Computer Science and Information Technologies, 5(1), 908-912.

[41] Sinha, A., & Paradkar, A. (2006). Model-based functional conformance testing of web services operating on persistent data. In Proceedings of the 2006 workshop on Testing, analysis, and verification of web services and applications (pp. 17-22).

[42] Sneed, H. M., & Huang, S. (2006). WSDLTest-a tool for testing web services. In 2006 Eighth IEEE International Symposium on Web Site Evolution (WSE'06) (pp. 14-21). IEEE.

[43] Stepien, B., Peyton, L., & Xiong, P. (2008). Framework testing of web applications using TTCN-3. International Journal on Software Tools for Technology Transfer, 10(4), 371-381.

[44] Stocco, A., Leotta, M., Ricca, F., & Tonella, P. (2017). APOGEN: automatic page object generator for web testing. Software Quality Journal, 25(3), 1007-1039.

[45] Tian, J., Ma, L., Li, Z., & Koru, A. G. (2003). A hierarchical strategy for testing web-based applications and ensuring their reliability. In Proceedings 27th Annual International Computer Software and Applications Conference. COMPAC 2003 (pp. 702-707). IEEE.

[46] Xu, L., Xu, B., & Jiang, J. (2005). Testing web applications focusing on their specialties. ACM SIGSOFT Software Engineering Notes, 30(1), 10.

[47] Jan, S., Panichella, A., Arcuri, A. and Briand, L., (2019). Search-based multi-vulnerability testing of XML injections in web applications. Empirical Software Engineering, pp.1-34.

[48] Tappenden, A., Beatty, P., Miller, J., Geras, A. and Smith, M., (2005). Agile security testing of web-based systems via httpunit. In Agile Development Conference (ADC'05) (pp. 29-38). IEEE.

[49] Tian-yang, G., Yin-Sheng, S. and You-yuan, F., (2010) . Research on software security testing. World Academy of science, engineering and Technology, 70, pp.647-651.

[50] Felderer, M., Büchler, M., Johns, M., Brucker, A.D., Breu, R. and Pretschner, A., (2016). Security testing: A survey. In Advances in Computers (Vol. 101, pp. 1-51). Elsevier.

[51] Büchler, M., Oudinet, J., & Pretschner, A. (2012). Semi-automatic security testing of web applications from a secure model. In *2012 IEEE Sixth International Conference on Software Security and Reliability* (pp. 253-262). IEEE.

[52] Huang, Y. W., Tsai, C. H., Lin, T. P., Huang, S. K., Lee, D. T., & Kuo, S. Y. (2005). A testing framework for Web application security assessment. *Computer Networks*, *48*(5), 739-761.

[53] Avancini, A., & Ceccato, M. (2011). Security testing of web applications: A search-based approach for cross-site scripting vulnerabilities. In *2011 IEEE 11th international working conference on source code analysis and manipulation* (pp. 85-94). IEEE.

[54] Curphey, M., & Arawo, R. (2006). Web application security assessment tools. *IEEE Security & Privacy*, *4*(4), 32-41.

[55] Armando, A., Carbone, R., Compagna, L., Li, K., & Pellegrino, G. (2010). Model-checking driven security testing of web-based applications. In *2010 Third International Conference on Software Testing, Verification, and Validation Workshops* (pp. 361-370). IEEE.

[56] Haixia, Y., & Zhihong, N. (2009). A database security testing scheme of web application. In *2009 4th International Conference on Computer Science & Education* (pp. 953-955). IEEE.

[57] Wassermann, G., & Su, Z. (2004). An analysis framework for security in web applications. In *Proceedings of the FSE Workshop on Specification and Verification of component-Based Systems (SAVCBS 2004)* (pp. 70-78).

[58] Thomé, J., Gorla, A., & Zeller, A. (2014). Search-based security testing of web applications. In *Proceedings of the 7th International Workshop on Search-Based Software Testing* (pp. 5-14).

[59] Salas, M. I. P., & Martins, E. (2014). Security testing methodology for vulnerabilities detection of xss in web services and ws-security. *Electronic Notes in Theoretical Computer Science*, *302*, 133-154.

[60] Bezemer, C. P., Mesbah, A., & van Deursen, A. (2009). Automated security testing of web widget interactions. In *Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (pp. 81-90).

[61] Fonseca, J., Vieira, M., & Madeira, H. (2009, June). Vulnerability & attack injection for web applications. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks* (pp. 93-102). IEEE.

[62] Li, N., Xie, T., Jin, M., & Liu, C. (2010). Perturbation-based user-input-validation testing of web applications. *Journal of Systems and Software*, *83*(11), 2263-2274.

[63] Xiong, P., & Peyton, L. (2010). A model-driven penetration test framework for Web applications. In *2010 Eighth International Conference on Privacy, Security and Trust* (pp. 173-180). IEEE.

[64] Pellegrino, G., & Balzarotti, D. (2014). Toward Black-Box Detection of Logic Flaws in Web Applications. In *NDSS*.

[65] Sudhodanan, A., Armando, A., Carbone, R., & Compagna, L. (2016). Attack Patterns for Black-Box Security Testing of Multi-Party Web Applications. In *NDSS*.

[66] Petukhov, A., & Kozlov, D. (2008). Detecting security vulnerabilities in web applications using dynamic analysis with penetration testing. *Computing Systems Lab, Department of Computer Science, Moscow State University*, 1-120.

[67] Dukes, L., Yuan, X., & Akowuah, F. (2013). A case study on web application security testing with tools and manual testing. In *2013 Proceedings of IEEE Southeastcon* (pp. 1-6). IEEE.

[68] Mainka, C., Somorovsky, J., & Schwenk, J. (2012). Penetration testing tool for web services security. In *2012 IEEE Eighth World Congress on Services* (pp. 163-170). IEEE.

[69] Fonseca, J., Vieira, M., & Madeira, H. (2013). Evaluation of web security mechanisms using vulnerability & attack injection. *IEEE Transactions on dependable and secure computing*, *11*(5), 440-453.

[70] Bozic, J., Garn, B., Kapsalis, I., Simos, D., Winkler, S., & Wotawa, F. (2015). Attack pattern-based combinatorial testing with constraints for web security testing. In *2015 IEEE International Conference on Software Quality, Reliability and Security* (pp. 207-212). IEEE.

[71] Artzi, S., Dolby, J., Jensen, S. H., Møller, A., & Tip, F. (2011). A framework for automated testing of javascript web applications. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 571-580).

[72] Marback, A., Do, H., He, K., Kondamarri, S., & Xu, D. (2009). Security test generation using threat trees. In *2009 ICSE Workshop on automation of software test* (pp. 62-69). IEEE.

[73] Zech, P., Felderer, M., & Breu, R. (2019). Knowledge-based security testing of web applications by logic programming. *International Journal on Software Tools for Technology Transfer*, *21*(2), 221-246.

[74] Bozic, J., & Wotawa, F. (2018). Planning-based security testing of web applications. In *2018 IEEE/ACM 13th International Workshop on Automation of Software Test (AST)* (pp. 20-26). IEEE.

[75] Mesbah, A., & Prasad, M. R. (2011). Automated cross-browser compatibility testing. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 561-570).

[76] Xu, L., Xu, B., Nie, C., Chen, H., & Yang, H. (2003). A browser compatibility testing method based on combinatorial testing. In *International Conference on Web Engineering* (pp. 310-313). Springer, Berlin, Heidelberg.

[77] Motwani, A., Agrawal, A., Singh, N., & Shrivastava, A. (2015). Novel Framework for Browser Compatibility Testing of a Web Application using Selenium. *International Journal of Computer Science and Information Technologies*, *6*(6), 5159-5162.

[78] Chen, Z., Peng, T., Gong, S., Li, L., Yan, L., & Han, H. (2018). Intelligence Compatibility Test of Web Services-based Power Business. In 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 2477-2481). IEEE.

[79] Li, X., & Zeng, H. (2014). Modeling web application for cross-browser compatibility testing. In *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 1-5). IEEE.

[80] Tanaka, H. (2019). X-BROT: Prototyping of Compatibility Testing Tool for Web Application Based on Document Analysis Technology. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)* (Vol. 7, pp. 18-21). IEEE.

[81] Shi, H., & Zeng, H. (2015). Cross-browser compatibility testing based on model comparison. In 2015 International Conference on Computer Application Technologies (pp. 103-107). IEEE.

[82] Minamide, Y., & Mori, S. (2012). Reachability analysis of the HTML5 parser specification and its application to compatibility testing. In International Symposium on Formal Methods (pp. 293-307). Springer, Berlin, Heidelberg.

[83] Saqib, N., & Shahzad, S. (2018). Functionality, performance, and compatibility testing: A model based approach. In 2018 International Conference on Frontiers of Information Technology (FIT) (pp. 170-175). IEEE.

[84] Poberežnik, Ł. (2013). A method for selecting environments for software compatibility testing. In 2013 Federated Conference on Computer Science and Information Systems (pp. 1355-1360). IEEE.

[85] Kaalra, B., & Gowthaman, K. (2014). Browser Compatibility Testing Using Manual Testing Methods and Test Tools. International Journal of Advanced Studies in Computers, Science and Engineering, 3(10), 1.

[86] Li, L., & Chou, W. (2015). Compatibility Modeling and Testing of REST API based on REST Chart. In WEBIST (pp. 194-202).

[87] Gaardboe, R., Sandalgaard, N., & Sudzina, F. (2017). The importance of task compatibility for web-enabled Business Intelligence success in e-government. In Proceedings of the 19th international conference on information integration and web-based applications & services (pp. 353-357).

[88] CHENG, Y., QIN, D., & YANG, G. (2017). Mutation test method for browser compatibility of JavaScript. Journal of Computer Applications, (4), 41.

[89] Balasundaram, P., Ramadoss, A., & Muthaiya, M. (2012). IE browser compatibility for web applications. ACM SIGWEB Newsletter, (Autumn), 1-15.

[90] Chen, Z., Peng, T., Gong, S., Li, L., Yan, L., & Han, H. (2018). Intelligence Compatibility Test of Web Services-based Power Business. In 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 2477-2481). IEEE.

[91] Gao, H., & Miao, H. (2013). Research on the dynamic reconfiguration of Web application using two-phase compatibility verification. International Journal of Computer Mathematics, 90(11), 2265-2278.

[92] Kumar, M., & Arya, S. (2016). A Novel Approach to Extend Selenium DB for Better Compatibility with the Web Based Application Testing. International Journal of Latest Research in Engineering and Technology, 2(7), 12-16.

[93] Manhas, J., & Sarwar A. (2011). Cross Browsers Compatibility of E-Government Websites. Proceedings of the 5 th National Conference; INDIACom-2011.

[94] Ardito, L., Maurizio, M., & Shijie, H. (2019). Test Fragility: An exploratory assessment study on an Open-Source Web Application.

[95] Praphamontripong, U., & Offutt, J. (2010). Applying mutation testing to web applications. In 2010 Third International Conference on Software Testing, Verification, and Validation Workshops (pp. 132-141). IEEE.

[96] Liu, C. H., Kung, D. C., Hsia, P., & Hsu, C. T. (2000). Structural testing of web applications. In *Proceedings 11th International Symposium on Software Reliability Engineering. ISSRE 2000* (pp. 84-96). IEEE.

[97] Liu, C. H., Kung, D. C., & Hsia, P. (2000). Object-based data flow testing of web applications. In *Proceedings First Asia-Pacific Conference on Quality Software* (pp. 7-16). IEEE.

[98] Kung, D. C., Liu, C. H., & Hsia, P. (2000). An object-oriented web test model for testing web applications. In *Proceedings First Asia-Pacific Conference on Quality Software* (pp. 111-120). IEEE.

[99] Liu, C. H., Kung, D. C., Hsia, P., & Hsu, C. T. (2001). An object-based data flow testing approach for web applications. *International Journal of Software Engineering and Knowledge Engineering*, *11*(02), 157-179.

[100] Liu, C. H. (2006). Data flow analysis and testing of JSP-based Web applications. *Information and Software Technology*, *48*(12), 1137-1147.

[101] Yang, J. T., Huang, J. L., & Wang, F. J. (1998). A tool set to support web application testing. In *Proceedings of the International Computer Symposium (ICS98)*.

[102] Endo, A. T., da Silva Simao, A., de Souza, S. D. R. S., & de Souza, P. S. L. (2008). Web services composition testing: a strategy based on structural testing of parallel programs. In Testing: Academic & Industrial Conference-Practice and Research Techniques (taic part 2008) (pp. 3-12). IEEE.

[103] Liu, C. H., Chen, S. L., & Li, X. Y. (2008). A WS-BPEL based structural testing approach for web service compositions. In 2008 IEEE International Symposium on Service-Oriented System Engineering (pp. 135-141). IEEE.

[104] Eler, M. M., Delamaro, M. E., Maldonado, J. C., & Masiero, P. C. (2010). Built-in structural testing of web services. In 2010 Brazilian Symposium on Software Engineering (pp. 70-79). IEEE.

[105] Zheng, X., & Chen, M. (2005). Specification-driven Structural testing of multi-tier web applications. http://www.cs.albany.edu/-mhc/WebMTA/paper2.

[106] Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques-a literature review. International Journal of Embedded Systems and Applications (IJESA), 2(2), 29-50.

[107] Artzi, S., Dolby, J., Jensen, S. H., Møller, A., & Tip, F. (2011). A framework for automated testing of javascript web applications. In Proceedings of the 33rd International Conference on Software Engineering (pp. 571-580).

[108] Ricca, F. (2004). Analysis, testing and re-structuring of web applications. In 20th IEEE International Conference on Software Maintenance, 2004. Proceedings. (pp. 474-478). IEEE.

[109] Endo, A. T., Linschulte, M., da Silva Simão, A., & de Souza, S. D. R. S. (2010). Event-and coverage-based testing of web services. In 2010 Fourth International Conference on Secure Software Integration and Reliability Improvement Companion (pp. 62-69). IEEE.

[110] Qian, Z., Miao, H., & Zeng, H. (2007). A practical web testing model for web application testing. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 434-441). IEEE.

[111] Karam, M., Keirouz, W., & Hage, R. (2006). An abstract model for testing mvc and workflow based web applications. In Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06) (pp. 206-206). IEEE.

[112] Bai, X., Dai, G., Xu, D., & Tsai, W. T. (2006). A multi-agent based framework for collaborative testing on web services. In The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06) (pp. 6-pp). IEEE.

[113] Qi, Y., Kung, D., & Wong, E. (2005). An agent-based testing approach for Web applications. In 29th Annual International Computer Software and Applications Conference (COMPSAC'05) (Vol. 2, pp. 45-50). IEEE.

[114] Carniello, A., Jino, M., & Chaim, M. L. (2005). Structural testing with use cases. Journal of Computer Science & Technology, 5.

[115] Elbaum, S., Karre, S., & Rothermel, G. (2003). Improving web application testing with user session data. In 25th International Conference on Software Engineering, 2003. Proceedings. (pp. 49-59). IEEE.

[116] Dubey, H. K., Kumar, P., Singh, R., Yadav, S. K., & Yadav, R. S. (2012). Automated data flow testing. In 2012 Students Conference on Engineering and Systems (pp. 1-5). IEEE.

[117] Sampath, S., & Bryce, R. C. (2012). Improving the effectiveness of test suite reduction for user-session-based testing of web applications. Information and Software Technology, 54(7), 724-738.

[118] Wang, W., Sampath, S., Lei, Y., & Kacker, R. (2008). An interaction-based test sequence generation approach for testing web applications. In 2008 11th IEEE High Assurance Systems Engineering Symposium (pp. 209-218). IEEE.

[119] Romli, R., Sarker, S., Omar, M., & Mahmod, M. Automated Test Cases and Test Data Generation for Dynamic Structural Testing in Automatic Programming Assessment Using. International Journal on Advanced Science, Engineering and Information Technology, 10(1), 120-127.

[120] Olugbara, C. T., & Letseka, M. (2020). Factors Predicting Integration of E-Learning by Preservice Science Teachers: Structural Model Development and Testing. Electronic Journal of e-Learning, 18(5).

[121] Kiran, S., Mohapatra, A., & Swamy, R. (2015). Experiences in performance testing of web applications with Unified Authentication platform using Jmeter. In *2015 international symposium on technology management and emerging technologies (ISTMET)* (pp. 74-78). IEEE.

[122] Kaalra, B., & Gowthaman, K. (2014). Cross Browser Testing Using Automated Test Tools. International Journal of Advanced Studies in Computers, Science and Engineering, 3(10), 7.