INTERNATIONALJOURNALOFENGINEERING SCIENCES&MANAGEMENT

A MODEL FOR WEB BASED APPLICATION USING MANUAL AND AUTOMATED TESTING TECHNIQUES AND ALSO STUDY FAULTS, THEIR EFFECTS AND TESTING CHALLENGES IN WEB BASED APPLICATIONS

Prateek Rokadiya*1, Saumy Dadhich2, Suyash Singhal3, Akshita Pamecha4, Siddarth Sharma5, Payal Dodeja6

ABSTRACT

Now a day, most of the desktop-based applications are becoming online. The Purpose of creating and developing online application is to provide the facility to the customers at any time. There are variety of applications are available on internet, which serves the specific services to the customers like flipkart, Amazon and many more. In order to provide quality services to the customers/end-client, there may be continue process so that we can upgrade our web application. Therefore, in this proposed research work, a web application-testing architecture and tool is developed and designed. During different phases of development and deployment, the model includes different testing approaches for maintain the quality of the product and services. With the help of this model, we will also provide the result analysis and the future enhancement.

Keywords- *Software Testing, architecture, defect, web application.*

I. INTRODUCTION

The web applications are one of the crucial applications. The concerns of such application in terms of testing and the way they provide the services to the end user must be taking care in the effective manner. In this paper, we have investigated the software testing strategy and techniques. There are two main categories of web applications first static web applications and second dynamic web applications. In static web, applications a set of connected web pages (HTML) forms are available, and using the client request the web page is reflect through the web browser. These applications only contain static data such as text, images and other information. On the other hand, some web applications are able to accept the user input and according to the user input, they change their states or provide the user oriented data after submission of request. Such kind of application is known as dynamic web applications.

Both kinds of applications are easily implementable with the existing technologies such as JSP, ASP, PHP and other similar technologies. Therefore, in web servers (who distribute the web response) the type of application server is also required to support this application page execution. Therefore, a static web application can be work according to the figure 1,

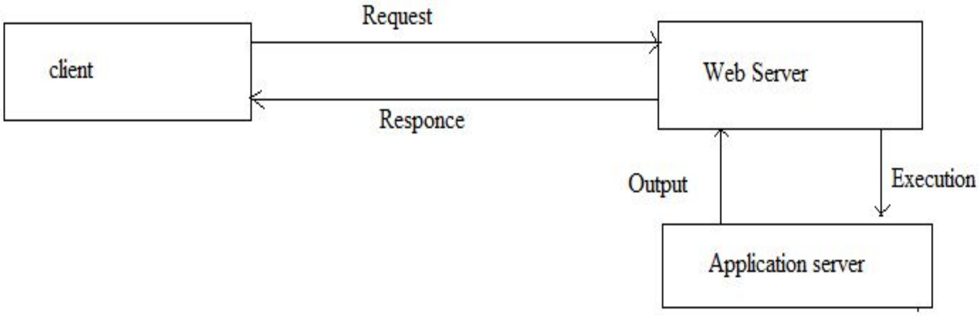

Figure 1: Static Web Applications

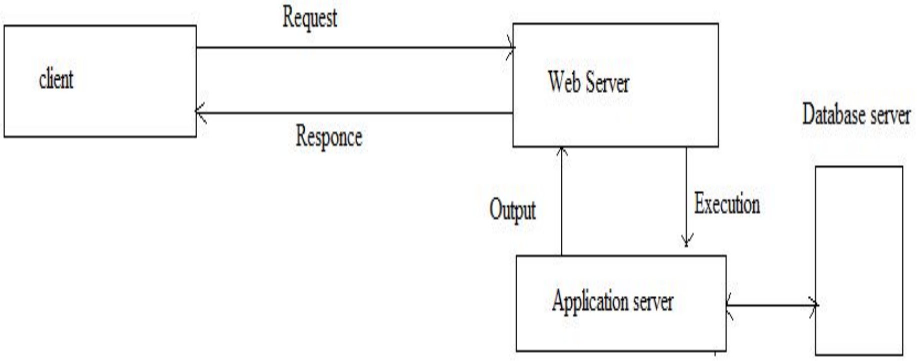

Figure 2: Dynamic Web Applications

Moreover, the dynamic application can work based on figure 2, the basic difference between both the applications are the existence of data base server. Remaining requirements for executing the request and response are same in both the systems.

In addition, of that, the web applications are desire to make more attractive and efficient, therefore various other optimization techniques and scripts are also integrate with the application.

II. WEB APPLICATION TESTING CHALLENGES

Since the Web's inception the goals and functionality offered by Web applications, as well as the technologies used to implement them, have changed considerably. Early Web applications comprised a simple set of static HTML pages. However, applications that are more recent offer the users a variety of functions for manipulating data, accessing databases, and carrying out a number of productive processes. These functions are usually perform by means of software component implemented by different technologies such as Java Server Pages (JSP), Java Servlets, PHP, CGI, XML, ODBC, JDBC, or proprietary technologies such as Microsoft's Active Server Pages (ASP). These components exploit complex, heterogeneous execution environment including hardware, software and middle ware components. The remainder of this chapter uses the term Web application (or simply application) to indicate the set of software components implementing the functionality and services the application provides to its users, while the term running environment will indicate the whole infrastructure (composed of hardware, software and middle ware components)needed to execute a Web application. The main goal of testing a Web application is to run the application using combinations of input and state to discover failures. A failure is the manifested inability of a system or component to perform a required function within specified performance requirements. Failures can be attributes to faults in the application's implementation. Generally, there will be failures due mainly to faults in the application itself and failures that will be mainly causes by faults in the running environment or in the interface between the application and the environment on which it runs. Since a Web application is strictly inter woven to its running environment, it is not possible to test it separately to find out exactly what component is responsible for each exhibited failure. Therefore, different types of testing have to be executes to uncover these diverse types of failures. The running environment mainly affects the non-functional requirements of a Web application (e.g. performance, stability, compatibility, usability), while the web application is responsible for the functional requirements. Thus, Web application testing will be judge from two distinct perspectives. One perspective identifies the different types of testing that need to be executing to verify the conformance of a Web application with specified non-functional requirements. The other perspective considers the problem of testing the functional requirements of an application. It is necessary that an application be testing from both perspectives, since they are complementary and not mutually exclusive.

III. PROBLEM DOMAIN AND PROPOSED ARCHITECTURE

The proposed work intended to find a best testing strategy for web based application testing. Now in these days the traditional desktop applications are becomes online-based systems. Therefore, web application development is in on high demand. On the other hand, these applications are using to store sensitive and important data therefore the security and other issues are also involve in this domain. In addition of that the application development and their internal architecture is different from the traditional desktop-based application development.

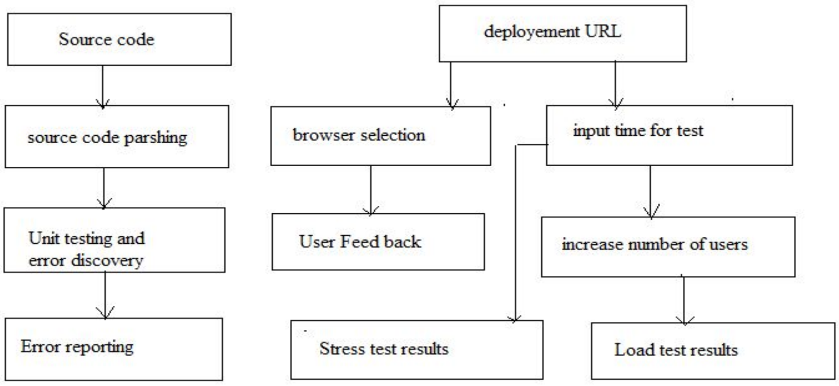

Figure 3: Proposed Testing Model

IV. RESULT ANALYSIS

Time Complexity

Time complexity also known as time consumption that estimated using the time difference between initialization of code processing and the finalizing of testing. In this context, the total time required to perform the test over web application's code files is termed as time complexity.

Table 5.1: Unit testing results

S. No.	Number of code files	Time (Sec)
1	5	3.6
2	10	5.11
3	15	8.29
4	20	10.93
5	25	12.74
6	30	15.03
7	35	18.97

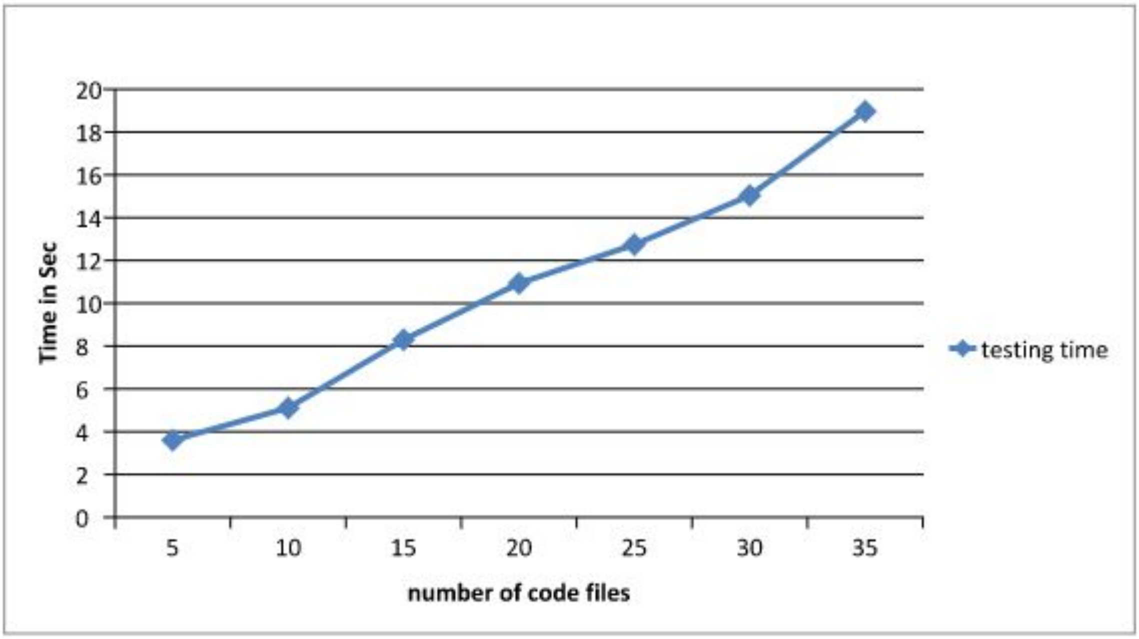

Figure 5.1 Unit testing results

Figure 5.1 shows the unit testing performance of proposed web application testing tool. In this diagram, the X-axis shows the amount of code files for testing and Y-axis shows the time consumed during testing. The web application consists of various other scripts and supporting files are included. Thus, these files are not included during the testing results demonstration. According to results, the amount of time required to test the code files. There are two facts found:

1. more number of code files leads more time consumption
2. If number of code lines in a code is larger than the amount of time required to test the application is also increases in the same manner.

Space Complexity

The amount of main memory required to test an application is termed here as time complexity of the system. Time complexities of the testing tools are considering in two different parts. First memory consumption during the load testing that is demonstrated using figure 5.2 and memory consumption during code testing which is demonstrated using figure 5.3.

Table 5.2: Space complexities during load testing

S. No.	Number of experiments	Memory consumption (KB)
1	1	27193
2	2	28291
3	3	29981
4	4	30258
5	5	31938
6	6	32948
7	7	34992

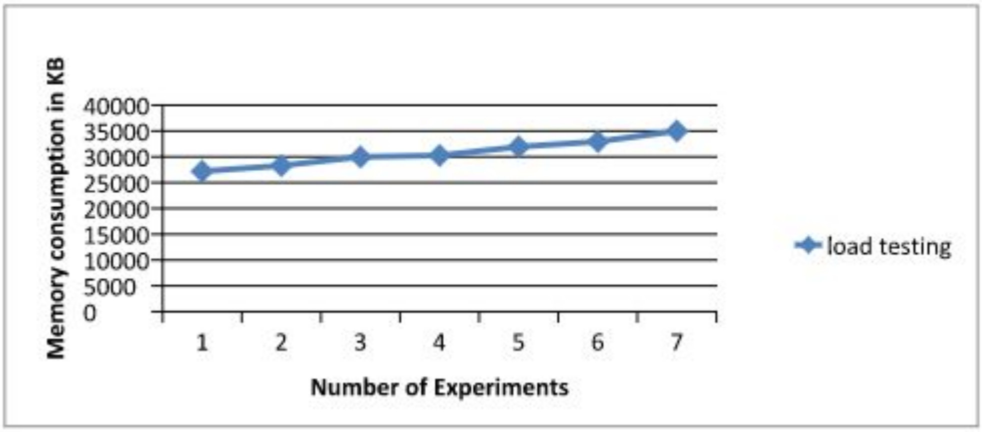

Figure 5.2: Space complexities during load testing

Figure 5.2 shows the space complexity during the load testing, during load, testing the client machine main memory consumption estimated and during each experiment 5 numbers of users are increased. During this testing environment, the estimated results are given using figure 5.2. According to the evaluated results as the number of users increase, the memory consumption of the testing tool is increases in the similar manner. In this diagram, the amount of main memory given using Y-axis and the X-axis shows the number of experiments.

V. CONCLUSION & FUTURE WORK

Now a day, most of the desktop-based applications is becoming online. There are various applications are such as shopping web, banking and other customer management system applications are available online. In order to provide continuous service to the end client, quality applications are required to develop.

The implementation of the proposed testing tool performed using visual studio environment and their performance evaluated with different experimental test beds. The obtained performance of the designed system is summarized using table1

Table1: Performance Summary

S. No.	Parameters	Remark
1	Time complexity	Low
2	Space complexity	Low
3	Response time	Low
4	Error detection rate	High

The above given facts of performance can be discussing as:

I. Time complexity of the system is low, but the time to compile, time taken to test the functions may also increase as the number of code file increases.
II. The above table show that the space complexity of the system is low and it does not depend upon the code written in different files. It is directly proportional to the lines of code actually necessary to compile.
III. As the load on the server gets increases the response time of the system will also get increase. Therefore more the number of concurrent users cause the similar amount of load. On the other hand, if the amount of response time increases than, it will not affect much on the server load.
IV. Error detection rate is not depends upon the files or the code lines. It is directly depends on the deployment environment and incorrectly defined code blocks.

REFERENCES

1 *J. Conallen, "Building Web Applications with UML: Web Application Basics", available at http://www.informit.com/articles/article.aspx?p=30610 , Jan 17, 2003*

2 *Available at http://www.pearsonhighered.com/samplechapter/0201730383.pdf*

3 *M. Buchler, J. Oudinet and A. Pretschner, "SPaCiTE - Web Application Testing Engine", IEEE Fifth International Conference on Software Testing, Verification and Validation (ICST), 2012.*

4 *Y. Liu, P. Li, S. Liu and J. Wu, "Research on the Implementation Technology of Generic Codec for Web Application Testing", International Conference on Software Engineering and Computer Science (ICSECS), 2013.*

5 *Y. Li, P. K. Das and D. L. Dowe, "Two Decades of Web Application Testing—A Survey of Recent Advances", Preprint submitted to Elsevier February 20, 2014.*

6 *J. Bau, E. Bursztein, D. Gupta and J. Mitchell, "State of the Art: Automated Black-Box Web Application Vulnerability Testing", IEEE Symposium on Security and Privacy (SP), 2010.*

7 *S. R. Choudhary, M. R. Prasad and A. Orso, "X-PERT: A Web Application Testing Tool for Cross-Browser Inconsistency Detection", ISSTA'14, San Jose, CA, USA, ACM 978-1-4503-2645, July 21–25, 2014.*

8 *N. Alshahwan and M. Harman, "Automated Web Application Testing Using SearchBased Software Engineering", 26th IEEE/ACM International Conference onAutomated Software Engineering (ASE), 2011.*

9 *F. E. Eassa, M. Zaki, A. M. Eassa and T. Aljehani, "IMATT: An Integrated Multi-Agent Testing Tool for the Security of Agent-Based Web Applications", World Journal of Computer Application and Technology vol 1, no 2, pp. 19-28, 2013.*

10 *A. R. Arunachalam, "Innovative Approach of Testing in Event Driven Software (EDS)",Indian Journal of Science and Technology, vol 7, no s5, pp. 37–40, June 2014.*

11 *A. Mesbah, A. Deursen and D. Roes, "Invariant-Based Automatic Testing of Modern Web Applications", IEEE Transactions on software engineering, vol. X, no. Y, 2011.*

12 *S. Elbaum, G. Rothermel, S. Karre, and M. Fisher, "Leveraging User-Session Data to Support Web Application Testing", IEEE Transactions on software engineering, vol. 31, no. 3, March 2005.*

13 *J. T. Yang, J. L. Huang, F. J. Wang, William. and C. Chu, "An Object Oriented Architecture Supporting Web Application Testing", 0-7695-0368-3/99 $10.00 0 1999 EEE.*

14 *A. Arora and M. Sinha, "Web Application Testing: A Review on Techniques, Tools and State of Art", International Journal of Scientific & Engineering Research, ISSN 2229-5518, vol. 3, no. 2, February-2012.*

15 *V. Dallmeier, M. Burger, T. Orth and A. Zeller, "Web Mate: A Tool for Testing Web 2.0 Applications", Beijing, China, ACM 978-1-4503-1274-5/12/06 JSTools'12, June 13, 2012.*

16 *Dr. R. Kumar P. and K. Bhargav, "A Survey on Performance Testing Approaches of Web Application and Importance of WAN Simulation in Performance Testing", International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 05, May 2012.*

17 *J. Gao, X. Bai and W. T. Tsai, "Cloud Testing- Issues, Challenges, Needs and Practice", Software Engineering: An International Journal (SEIJ), vol. 1, no. 1, September 2011.*