



# ANALYSIS AND STRATEGY OF TESTING TECHNIQUES ON AUTOMATED WEB

**Mamta**

*Deptt. of Computer Science.*

## I. INTRODUCTION

### 1.1 Automation

*The term automation, derived by the earlier word automatic (coming from automaton), was not widely used before 1947, which were introduced in the 1930s. Automation is an automatic control. Automatic control is the use of various control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching on telephone networks, steering and stabilization of ships, aircraft and other applications with minimal or reduced human intervention [1]. It is also used to save energy and materials and to improve quality, accuracy and precision. Automation has been achieved by various means including mechanical, hydraulic, pneumatic, electrical, electronic devices and computers, usually in combination. Complicated systems, such as modern factories, airplanes and ships typically use all these combined techniques.*

### 1.2 The types of automation [2] are

- **Discrete control** is *on-off* control.
- **Continuous control** is feedback control, which is usually *continuous* and involves taking measurements using a sensor.
- **Control loop** is the measurement single variable called a *control loop*. Control that uses a measured signal, feeds the signal back and compares it to a set point, calculates and sends a return signal to make a correction, is called *closed loop* control. If the controller does not incorporate feedback to make a correction then it is *open loop*.
- **Computer control** can perform both sequential control and feedback control, and typically a single computer will do both in an industrial application. Control of an automated teller machine (ATM) is an example of an interactive process in which a computer will perform a logic derived response to a user selection based on information retrieved from a networked database [3].

### 1.3 Automated Web

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization [3].

#### 1.3.1 Testing

Software testing is an investigation conducted to provide information about the quality of the product or service under test. [4] Software testing can also provide an objective, independent view of the software to allow the



business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects). Software testing involves the execution of a software component or system. In general, these properties indicate the extent to which the component or system under test which meets the requirements that guided its design and development, responds correctly to all kinds of inputs, performs its functions within an acceptable time. The job of testing is an iterative process as when one bug is fixed, it can illuminate other. Software testing can provide objective, independent information about the quality of software and risk of its failure to users and / or sponsors. [5] For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an Agile approach, requirements, programming, and testing are often done concurrently [6].

## II. THE PROBLEMS WHICH OCCUR AT THE TIME OF TESTING ARE LISTED AS BELOW

### ➤ Defects

Software faults occur through the following processes. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure.[7]

### ➤ Input combinations and preconditions

A fundamental problem with software testing is that testing under all combinations of inputs and preconditions (initial state) is not feasible, even with a simple product.[4][8] This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing.

## III. WEB TESTING

It is software testing that focuses on web applications. Complete testing of a web-based system before going live can help address issues before the system is revealed to the public. **A web application performance tool (WAPT)** is used to test web applications and web related interfaces [8]. These tools are used for performance, load and stress testing of web applications, web sites, web API, web servers and other web interfaces. WAPT tends to simulate virtual users which will repeat either recorded URLs or specified URL and allows the users to specify number of times or iterations that the virtual users will have to repeat the recorded URLs. By doing so, the tool is useful to check for bottleneck and performance leakage in the website or web application being tested.

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

Browser compatibility Operating System compatibility Windows application compatibility where required WAPT allows a user to specify how virtual users are involved in the testing either increasing users or constant users or periodic users load.

**B. Web security testing** tells us whether Web based applications requirements are met when they are subjected to malicious input data [9]. There is a web application security testing plug-in collection for FireFox[2]

## IV. TESTING AUTOMATION IN WEB APPLICATIONS

In software testing, test automation is the use of special software to control the execution of tests and the comparison of actual outcomes with predicted outcomes [10]. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually. Test automation is critical for continuous delivery and continuous testing. Some software testing tasks, such as extensive low-level interface regression testing, can be laborious and time consuming to do manually. In addition, a manual approach might not always be effective in finding certain classes of defects. Test automation offers a possibility to perform these types of testing effectively. Once automated tests have been developed, they can be run quickly and repeatedly. There are many approaches to test automation [11], however below are the general approaches used widely.

### 4.1. Graphical user interface testing

A testing framework generates user interface events such as keystrokes and mouse clicks, and observes the changes that result in the user interface, to validate that the observable behavior of the program is correct.

### 4.2. API driven testing

API is testing framework that uses a programming interface to the application to validate the behaviour under test. API driven testing bypasses application user interface altogether. It can also be testing public (usually) interfaces to classes, modules or libraries are tested with a variety of input arguments to validate that the results that are returned are correct. Test automation tools can be expensive, and are usually employed in combination with manual testing. Test automation can be made cost-effective in the long term, especially when used repeatedly in regression testing.

## V. STATE TRANSITION TESTING

The State Transition Testing(STT) which is commonly used in real time, embedded and web-based type of software systems. An ACO optimization technique is used to generate the automatic state-transition test sequence, which gives a strong level of software coverage [12].

### 5.1 Test Cases

In automated testing the Test Engineer or Software quality assurance person must have software coding ability, since the test cases are written in the form of source code which, when run, produce output according to the assertions that are a part of it. One way to generate test cases automatically is model-based testing through use of a model of the system for test case generation, but research continues into a variety of alternative methodologies for doing so. In some cases, the model-based approach enables non-technical users to create automated business test cases in plain English so that no programming of any kind is needed in order to configure them for multiple operating systems, browsers, and smart devices [13].

### 5.2 Unit testing

A growing trend in software development is the use of testing frameworks such as the xUnit frameworks (for example, JUnit and NUnit) that allow the execution of unit tests to determine whether various sections of the code are acting as expected under various circumstances. Test cases describe tests that need to be run on the program to verify that the program runs as expected [14]. Test automation mostly using unit testing is a key

feature of agile software development, where it is known as test-driven development (TDD). Unit tests are written to define the functionality before the code is written. However, these unit tests evolve and are extended as coding progresses, issues are discovered and the code is subjected to refactoring [15]. Only when all the tests for all the demanded features pass is the code considered complete.

### **5.3 Graphical User Interface (GUI) testing for mobile applications**

Many test automation tools provide record and playback features that allow users to interactively record user actions and replay them back any number of times, comparing actual results to those expected. The advantage of this approach is that it requires little or no software development. This approach can be applied to any application that has a graphical user interface. A framework utilizes entirely different techniques because it is rendering HTML and listening to DOM Events instead of Windows API events. Headless browsers or solutions based on Selenium Web Driver are normally used for this purpose [15][16][17].

Another variation of this type of test automation tool is for testing mobile applications. This is very useful given the number of different sizes, resolutions, and operating systems used on mobile phones.

## **VI. CONTINUOUS TESTING**

Continuous testing is the process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate.[18][19] For Continuous Testing, the scope of testing extends from validating bottom-up requirements or user stories to assessing the system requirements associated with over arching business goals [20].

## **VII. TEST AUTOMATION INTERFACE**

Test automation interface are platforms that provide a single workspace for incorporating multiple testing tools and frameworks for System/Integration testing of application under test. The goal of Test Automation Interface is to simplify the process of mapping tests to business criteria without coding coming in the way of the process. Test automation interface are expected to improve the efficiency and flexibility of maintaining test scripts [21].

### **6.1 brief review of work already done**

Web applications are dynamic and interactive, as compared to traditional applications. Therefore, traditional testing techniques and tools are not sufficient for web applications testing. A variety of web application testing techniques has been proposed. As many approaches are out of the scope of this research, we only provide a brief overview of relevant ones.

Kung et al [22] developed a test generation method for web testing to capture structural and behavioral test artifacts of webapplications. The entities are represented as objects, and their structures, relationships and dynamic behaviors are described. An object oriented Web Test Model (WTM) has been proposed which represents artifacts from object, behavior and structure features. Song et al. [23] have proposed a model based on web frameset and browser interactions. Web framesets are employed in applications where layouts of some pages are identical. The process begins with modeling of application with framesets, followed by considering browser interactions along with framesets, modeling the web navigation, formalizing the navigation model, and finally generating and executing test cases. Lee and Offutt [24] proposed a system that generates test cases using

a form of mutation analysis. It focuses on validating the reliability of data interactions among Web-based software components. Specifically, it considers XML based component interactions. Ricca and Tonella [25] proposed a UML model of Web application for high level abstraction. The model is based entirely on static HTML links and does not incorporate any dynamic aspects of the software.

## VII. METHODOLOGY

Web application is a system which typically is composed of a database (or the back -end) and Web pages (the front end), with which users interact over a net work using a browser. A Web testing approach is proposed and an automated Web application testing system, which implements this approach, is described. In the proposed approach, hyperlinks of the website to be tested are automatically followed one by one to retrieve all HTML texts of its pages starting from the home page. The HTML text of each encountered page is analyzed to extract the needed information about it. Then, the collected information is used in the error checking process.

Further we are purposing an algorithm by applying an ant colony optimization technique, for generation of optimal and minimal test sequences for behavior specification of software. The approach can generates test sequence in order to obtain the complete software coverage. The comparison between two metaheuristic techniques (Genetic Algorithm and Ant Colony optimization) will also purposed for transition based testing.

In order to get a sufficient amount of information, the study has been divided into three main parts where each will form a part of the report;

- Literature survey.
- Case study.
- Validation.

An extensive literature study has been conducted which was indented for the identification of which automated testing methods, approaches and practices are considered state-of-the-art.

## REFERENCES

- [1] Li,Yuan-Fang, Das, Paramjit K. and Dowe, David L. 2014. Two decades of Web application testing—A survey of recent advances. *Information Systems*, Vol. 43, pp. 20–54
- [2] Sampath, E., Gibson, S., Sprenkle, S. And Pollock, L 2005. Coverage Criteria for Testing Web Applications. Technical Report 2005-17, Computer and Information Sciences, University of Delaware.
- [3] Mansour, N. and Hourri, M. 2006. Testing web applications. *Information and Software Technology*, Vol. 48, pp. 31–42.
- [4] Di Lucca, G. A. And Fasolino, A. R. 2006. Testing Web-based applications: The state of the art and future trends. *Information and Software Technology*, Vol. 48, pp. 1172-1186.
- [5] Kung, David C, Liu Chien-Hung and Hsia, Pei 2000. An Object Oriented Web Test Mode
- [6] 1 for Testing Web Applications. In: Proceedings of IEEE 24th Annual International Journal of Computer Applications (0975–8887) Volume 99–No.7, August 2014 .
- [7] Aditya P. Mathur “Foundation of Software Testing”, First Edition , Pearson Education, 2007.
- [8] Ian Sommerville, ” Software Engineering” , eight Edition, Pearson Edition, 2009.

- [9] Pressman, R.S, "Software engineering: A Practitioner 's Approach", 6<sup>th</sup> Ed, India: TMH,2005.
- [10] Briand, L. C, " on the many ways Software Engineering can benefit from Knowledge Engineering", Proc. 14th SEKE, Italy, pp. 3-6,2002.
- [11] Pedrycz W and Peters, J. F , " Computational Intelligence in Software Engineering", World Scientific Publishers,1998.
- [12] Mark Harman, "The Current State and Future of Search Based Software Engineering", International Conference on Software Engineering ,Future of Software Engineering ,IEEE Computer Society press, Washington, DC, USA, pp 342-357,2007 .
- [13] Praveen Ranjan Srivastava, Vinod Ramachandran, Manish Kumar, Gourab Talukder, Vivek Tiwari, Prateek Sharma , "Generation of test data using Meta Heuristics Approach", IEEE TEN CON 2008, India,2008.
- [14] Song, Bo, Miao, Huaikou, Chen, Shengbo 2009. Considering Web Frameset and Browser Interactions in
- [15] Aditya P. Mathur "Foundation of Software Testing" ,First Edition
- [16] ,Pearson Education,2007.
- [17] Phil. McMinn, "Search-Based Software Test Data Generation: A Survey", Software Testing, Verification and Reliability, Vol 14,No 3, p. 212-223,2004.
- [18] Briand, L. C, " The many ways Software Engineering can benefit from Knowledge Engineering", Proc. 14th SEKE, Italy, pp.3-6,2002.
- [19] Pedrycz W and Peters, J. F , " Computational Intelligence in Software Engineering", World Scientific Publishers,1998.
- [20] Mark Harman, "The Current State and Future of Search Based Software Engineering", International Conference on Software Engineering ,Future of Software Engineering ,IEEE Computer Society press, Washington, DC, USA, pp 342-357,2007 .
- [21] Lee suet Chun and Offutt, Jeff 2001. Generating test cases for XML-based Web component interactions using mutation analysis. In Proceedings of the 12<sup>th</sup> International Symposium on Software Reliability Engineering, pp. 200-209, Hong Kong China, November
- [22] 2001, IEEE Computer Society Press.
- [23] Yang, J.,Huang,J., Wang, F. And Chu, W.1999. An object-oriented architecture supporting Web application testing. In First Asian -Pacific Conference on Quality Software (APAQS '99), pp. 122-129, Japan, December 1999.
- [24] Yang, Ji-Tzay, Huang, Jiun -Long ,Wang, Fen-Jian and Chu,William C. 2002. Constructing an object – oriented architecture for Web application testing. Journal of Information Science and Engineering, Vol. 18, No. 1, pp. 59-84, January 2002.