

Enhancing A Software Testing Tool to Validate the Web Services

Tanuj Wala¹, Aman Kumar Sharma²

¹Research Scholar, Department of Computer Science, Himachal Pradesh University Shimla, India

²Associate Professor, Department of Computer Science, Himachal Pradesh University Shimla, India

Abstract - At some point of time there is no specific testing strategy to consider the idea of testing web services. But in today's scenario, services along with tools are easily available on time. Web services are continuous evolving technologies which are outcome of gradual improvement of internet services. The improvement is upgraded due to tremendous increase in demand placed on them. Web services are rapidly developing and are expected to change the paradigms for development and use of software. It implements Service Oriented Architecture (SOA) and is interoperable and loosely coupled. The improvements in web services promote software reusability, provide fast functionality and also pave a way for flexibility in the form of composite service offerings. This paper studied the functionality of two web services implement for a same input on a wizdl open source web service testing tool. The study compares the results of both web services for the accuracy of output.

Keywords: - Open source software, Software reusability, SOA, Web services, wizdl testing tool.

1. INTRODUCTION

A web service is a software system designed to support interoperable machine to machine interaction over a network. It has an interface that describes collections of operations accessible through standardized eXtended Markup Language (XML) messaging. A web service is described using a standard, formal XML notion, called its service description. It covers all the details necessary to interact with the service, including message formats, transport protocols and location. The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written. Web services can be used to communicate within heterogeneous networks since the communication is handled by using Simple Object Access Protocol (SOAP) messages which is build up by using an XML schema that may be used to call remote services or exchange data. With the changing trends and technology the web is evolving and expanding continuously and so are its services. Day by day web services

are getting more complex but publically accessible and usable. The benefits and challenges in web services are offered in a very useful, flexible, effective and secure way. Web services testing assumes that the tester only have the Web Service Description Language(WSDL) document of the web service under test and not the source code, for this reason black box testing techniques are used. The test cases for black box testing are based on the requirement and specification. Two types of test cases are used in testing named as successful and unsuccessful test cases. If the test case did not find an error then it's called a successful test run whereas if it discovers an error then it's called an unsuccessful test case [1].

Web services are considered an implementation or realization of the service oriented architecture that possesses an interface that supports interoperability between different devices using standard protocol. To implement SOA, web services depends on a group of XML based standards such as SOAP, WSDL and Universal Description, Discovery and Integration (UDDI) [2].SOAP is a messaging protocol used to encode the information in web service request and response messages before sending them over a network. WSDL specifies the location of the service, the operations or methods and UDDI is a registry where web services are stored. It is an XML based standard for describing, publishing and finding web services. Web services are playing an important role in e-business and e-commerce applications. With the increase of the popularity of web services, more and more web applications are developed with this new kind of components. As web service applications are interoperable and can work on any platform, large scale distributed systems can be developed easily using web services. This new way of software development brings about new issues for software testing, which has been widely recognized as a realistic means for ensuring the quality of software systems.

Web services does not have graphical user interface. It uses a XML data for communication. A web service testing involves testing of the service interface, implementation of service, message exchange and Service Level Agreement (SLA). Composite services are built by assembling multiple web services that interact to meet a business requirement [3].

The web services architecture is based upon the interactions between three roles: service provider, service registry and service requestor and three operations: publish, find and bind. Together, these roles and operations act upon the web services artifacts. Figure 1 shows the major roles, operations and artifacts within the architecture of the web services.

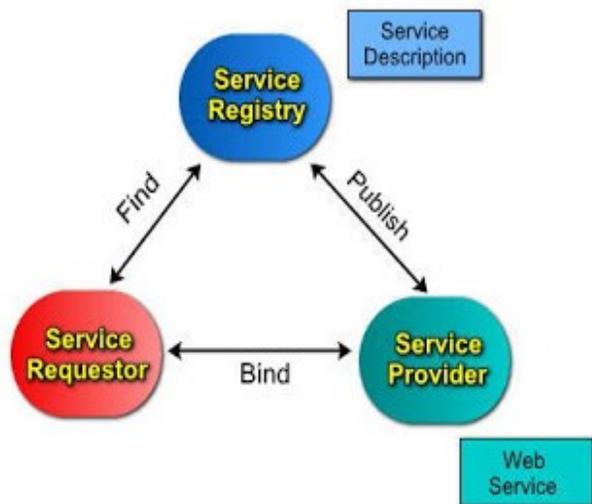


Figure 1: Web Service Roles, Operations and Artifacts [4].

The “service provider” implements the service and makes it available on the Internet. From a business perspective, this is the owner of the service. From an architectural perspective, this is the platform that hosts access to the service. The “service requestor” utilizes an existing web service by opening a network connection and sending an XML request. From a business perspective, this is the business that requires certain functions to be satisfied. From an architectural perspective, this is the application that is looking for and invoking an interaction with a service and the “service registry” provides a central place where developers can publish new services or find existing ones.

For an application to take advantage of web services, these three operations are important. In “publish” operation a service description needs to be published so that the service requestor can find it. In the “find” operation, the service requestor retrieves a service description directly or queries the service registry for the type of service required. In the “bind” operation the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact and invoke the service. As shown in Figure 1 these roles and operations act upon the web services artifacts modules.

“Web service” is an interface described by a service description; its implementation is the service. A service is a software module deployed on network accessible platforms provided by the service provider. The service description

contains the details of the interface and implementation of the service. These descriptions can be categorized into two types structural and non structural. The structural description contains declaration, message type, transmission protocol and location of the service and non structural descriptions of services are the text that are written in some native natural language and embedded in the WSDL [5]. The web services development lifecycle includes the design, deployment, and runtime requirements for each of the roles.

Web Service Testing Tools: -

Testing automation tools enables developers and testers to easily automate the entire process of testing in software development. There are so many tools available in the market to generate test cases automatically and also which helps the user in analyzing the request and response. Automated testing always concentrates on automatic generation of test cases and execution of such generated test cases [6]. This kind of testing saves time and costs less effort. There is a significant increase seen in web services over the internet world. So, different tools have been developed and designed to enhance the use of web services. Test tool selection is a very important part of test automation. This requires the study of the scope of testing and the test plan. It is also needed to know whether the test tool meets the test suite requirements for the particular product or not. The important factors that also come into picture are reusability, reliability and cost. Various open source and commercial web service testing tools are available in the market with different features and functionality. Commercial testing tools are not free for the user to use.

Open Source Software (OSS) is computer software with the availability of source code that is made freely available. These tools can be downloaded from the internet or can be obtained by the vendor without any charges. This software is distributed under licensing agreement. This licensing agreement allows the source code to be shared, viewed and modified by users or organization. The various open source web service testing tools are available on the internet such as Soapui Pro, Wef Storm, Apache Jmeter, wizdl and WebInject. The core functions of these tools are similar to some extent but differ in functionality, features, usability and interoperability. The selection of the best tool is based on the parameters analysis such as response time etc. [7].

The study in this paper is based on the wizdl open source web service testing tool. wizdl is a .NET utility written in C# that allows quick import and test web services within the comfort of a windows forms GUI. It supports calling complex web services that take arrays and deeply nested objects as

parameters. Being pointed at web service it dynamically creates and interface where the required data can be entered and execution of service can be done. The tool allows saving the data as an XML file which can be loaded later for regression testing [8]. The tool can be downloaded from <http://wizdl.codeplex.com>.

The original wizdl tool does not show any parameters which are important for efficient web service testing tools. After modification in wizdl, tool provides three additional parameters: response time, throughput and number of data processed in terms of bytes. But still wizdl has some limitations like the tool calculate the results of only one web service at a time [9]. So, the approach used in this study is based on to running the concurrent web services for the same input, so that the results can be checked and validated for the accuracy. The OSS tool is modified to make it more robust and sturdy. The results for the same input are conformed from two web services and the results are displayed from both. The purpose of using two web services for same input is to increase the performance and efficiency of the tool which directly determines the accuracy and precision value of the web services.

The paper can be studied under the following sections: Section I, gives the brief introduction about web services and web service testing tools. Section II provides the implementation of wizdl web service testing tool using two web services. Section III gives the result and discussion and Section IV concludes the paper along with scope for future work.

II. IMPLEMENTATION OF WIZDL USING WEB SERVICES

This section studied the functionality of two web services on wizdl open source web service testing tool. The wizdl web service tool was modified in its source code. The aim of modification was to accept the input and fetch the result for the web service tool, from two web services and the user has access to both the results. Further graphically the results are displayed to highlight the variation in results. The tests were conducted at the same instance of time and at a same network speed.

1) System Requirements

The tool was installed on an Intel Core i3 2.40 GHz processor machine with 3GB RAM, Microsoft Windows 7 Home Basic, and 2Mbps Internet connection. Testing of the tool requires configuration which in turn includes installation, setting up test environment, data collection, selection of parameters and analytical survey. The sample web services i.e. DelayedStockQuote and StockQuote are tested via implementation of configured tool. The detail of web services is presented in Table 1.

Table 1 shows the Delayed StockQuote and StockQuote web services. ID field gives unique identification for each web services. W1 is the ID for first web service i.e. DelayedStockQuote and W2 is the ID for second web service i.e. StockQuote.

Table 1: Sample of Web Services

ID	Web Service Name	Description	Methods	Publisher	WSDL URL
W1	Delayed StockQuote	Get Stock quote for a CompanySymbol +Stock price+Stock information	-GetQuickQuote. -GetQuote. -GetQuoteDataset.	Cdyne Corporation	http://ws.cdyne.com/delayedstockquote/delayedstockquote.asmx?WSDL
W2	StockQuote	Get Stock quote for a company Symbol	-GetQuote.	WebServiceX. Net	http://www.webserviceX.net/stockquote.asmx?WSDL

“DelayedStockQuote” published by Cdyne corporation contains three methods named as GetQuickQuote, GetQuote and GetQuoteDataset. “GetQuickQuote” method retrieves just a stock price. It does not provide any others variable related to stock like previous close, open amount, earn per share, fifty two week range, change percent, stock change, stock symbol and stock volume. “GetQuote” method retrieves a current stock quote. It provides all the variables value related to stock. “GetQuoteDataset” method retrieves the stock information and returns it in a dataset. “StockQuote” web service published by WebServiceX.Net contains only one method i.e. “GetQuote” and provides information of StockQuote for a specific company symbol.

2) Approach Followed

The research followed a practical approach for wizdl web service testing tool for the study. Initially wizdl tool whose results are shown in Figure 2 displayed the results of one web service. So the effort was placed to accommodate these changes into the web service tool. Respective formulas and coding have been incorporated in the initial source code of wizdl tool to calculate the results of two web services simultaneously. Thus observed results were collected and compiled for the analysis and evaluation to determine the efficiency and accuracy of the web services.

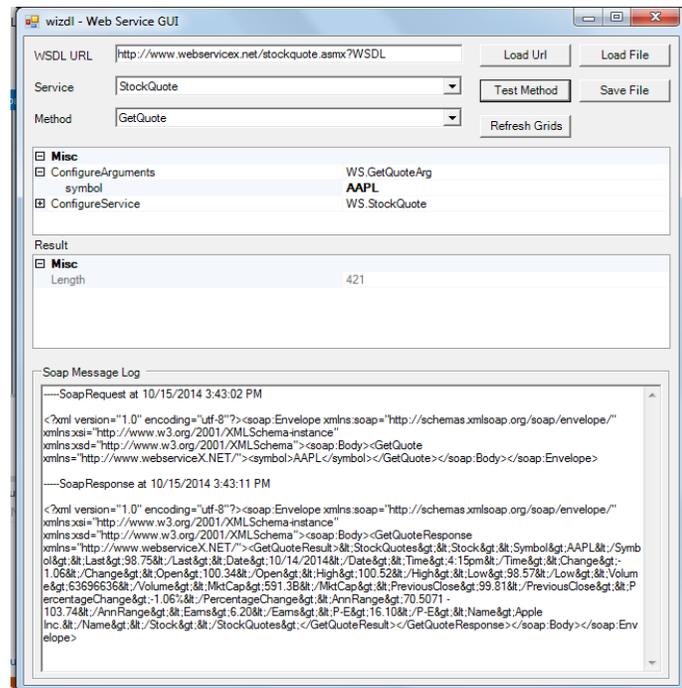


Figure 2: Original wizdl testing tool.

The Figure 3 shows the two web services named as DelayedStockQuote and StockQuote are applied in the wizdl testing tool. Firstly add the wsdl file for both the web services. To perform test the study needs to provide the input parameters for methods. For both the web services “GetQuote” method is selected and gives the “AAPL” company symbol as input to the wizdl. The same input is applied for both the web services, So that the results can be compared for accuracy of the output.

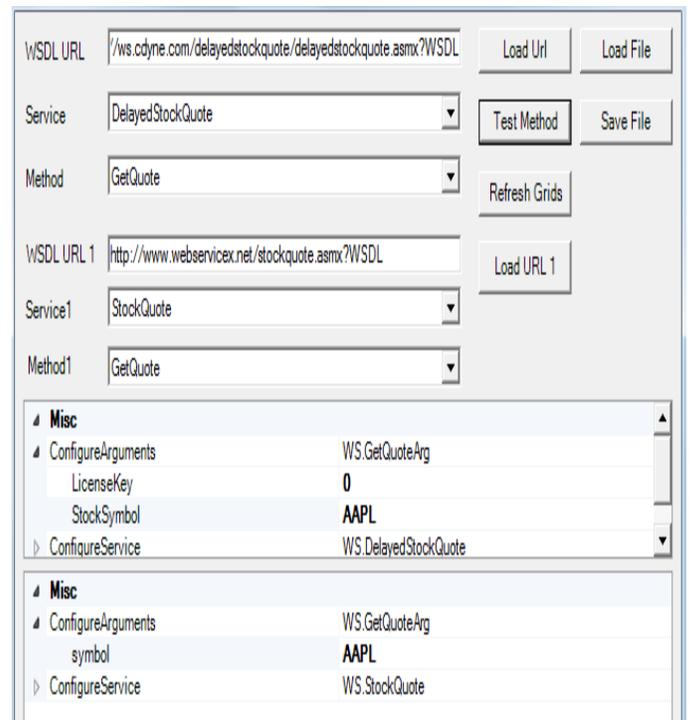


Figure 3: Modified wizdl testing tool

After providing the input the results of the web services along with the SOAP response is returned to the user. wizdl testing tool provide StockQuote service details for the Apple Inc. company in the form of company name, last trade amount, last trade date time, open amount, day high, day low, change percentage, fifty two week range, earn per share, stock symbol, stock change, previous close and stock volume. Figure 4 shows the results of DelayedStockQuote and StockQuote web services.

```
-----SoapRequest at 10/3/2014 11:02:04 AM

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetQuote
xmlns="http://ws.cdyne.com/"><StockSymbol>AAPL</StockSymbol><LicenseKey>0</LicenseKey></GetQuote></soap:
Body></soap:Envelope>

-----SoapResponse at 10/3/2014 11:02:07 AMResponse time =2894.1656millisec.
Throughput : 41.4627276338299 request/min
Throughput(KB/s) : 0.572946871112005 KB/s

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetQuoteResponse
xmlns="http://ws.cdyne.com/"><GetQuoteResult><StockSymbol>AAPL</StockSymbol><LastTradeAmount>99.90</Last
TradeAmount><LastTradeDateTime>2014-10-
02T16:15:00</LastTradeDateTime><StockChange>0.72</StockChange><OpenAmount>99.36</OpenAmount><DayHigh
>100.22</DayHigh><DayLow>98.04</DayLow><StockVolume>47757828</StockVolume><PrevCls>99.18</PrevCls><
ChangePercent>+0.73%</ChangePercent><FiftyTwoWeekRange>68.3257 -
103.74</FiftyTwoWeekRange><EamPerShare>6.20</EamPerShare><PE>16.00</PE><CompanyName>Apple
Inc.</CompanyName><QuoteError>false</QuoteError></GetQuoteResult></GetQuoteResponse></soap:Body></soap:E
nvelope>

-----SoapRequest at 10/3/2014 11:02:07 AM

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetQuote
xmlns="http://www.webserviceX.NET/"><symbol>AAPL</symbol></GetQuote></soap:Body></soap:Envelope>

-----SoapResponse at 10/3/2014 11:02:09 AMResponse time =1352.0774millisec.
Throughput : 88.7523155109315 request/min
Throughput(KB/s) : 1.42864648503111 KB/s

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><GetQuoteResponse
xmlns="http://www.webserviceX.NET/"><GetQuoteResult>&lt;StockQuotes&gt;&lt;Stock&gt;&lt;Symbol&gt;AAPL&lt;/Symb
ol&gt;&lt;Last&gt;99.90&lt;/Last&gt;&lt;Date&gt;10/2/2014&lt;/Date&gt;&lt;Time&gt;4:15pm&lt;/Time&gt;&lt;Change&gt;
+0.72&lt;/Change&gt;&lt;Open&gt;99.36&lt;/Open&gt;&lt;High&gt;100.22&lt;/High&gt;&lt;Low&gt;98.04&lt;/Low&gt;&lt;Volum
e&gt;47757828&lt;/Volume&gt;&lt;MktCap&gt;598.2B&lt;/MktCap&gt;&lt;PreviousClose&gt;99.18&lt;/PreviousClose&gt;&lt;P
ercentageChange&gt;+0.73%&lt;/PercentageChange&gt;&lt;AnnRange&gt;68.3257 -
103.74&lt;/AnnRange&gt;&lt;Eams&gt;6.20&lt;/Eams&gt;&lt;P-E&gt;16.00&lt;/P-E&gt;&lt;Name&gt;Apple
Inc.&lt;/Name&gt;&lt;/Stock&gt;&lt;/StockQuotes&gt;</GetQuoteResult></GetQuoteResponse></soap:Body></soap:Env
elope>
```

DelayedStockQuote
(W1)Results

StockQuote
(W2)Results

Figure 4: Web Services Results.

From the above results it can be seen that the tool provides DelayedStockQuote and StockQuote web services results in terms of stock amount details for the Apple Inc. company. Further the modified tool also calculates the total response time and throughput. The total response time for the results of DelayedStockQuote web service is 2894.1656 milliseconds, throughput is 41.46request/min and for the StockQuote web service the total response time for the results are 1352.0774 milliseconds, throughput is 88.75request/min.

III. RESULTS AND DISCUSSION

The observed results of both the web service are shown in Table 2 and in Table 3 respectively. AAPL is the company symbol for the Apple Inc. company and is provided as an input to the wizdl web service testing tool. Table 2 and Table 3 show the evaluated results of the DelayedStockQuote and StockQuote web service in terms of respective input.

Table 2: DelayedStockQuote Web Service

DelayedStockQuote Web Service(W1)	
Change percent	+0.73%
Company Name	Apple.Inc
Day High	100.22
Day Low	98.04
Earn Per Share	6.20
Fifty Two Week Range	68.3257-103.74
Last Trade Amount	99.90
Last Trade Date Time	10/2/2014 4:15PM
Open Amount	99.36
PE	16.00
Prev CIS	99.18
Quote Error	False
Stock Change	0.72
Stock Symbol	AAPL
Stock Volume	47757828

Table 3: StockQuote Web Service

StockQuote Web Service(W2)	
Percentage Change	+0.73%
Name	Apple.Inc
High	100.22
Low	98.04
Earn	6.20
AnnRange	68.3257-103.74
Last	99.90
Date Time	10/2/2014 4:15PM
Open	99.36
P-E	16.00
Previous close	99.18
Mkt Cap	598.2B
Change	0.72
Symbol	AAPL
Volume	47757828

From the tables it can be studied and concluded that, in both web services results, each variable value is same. StockQuote web service provide one additional variable i.e. Mkt Cap (market capitalization) and its value is 598.2B. Market capitalization is the market value of a company's outstanding shares. Market capitalization is calculated by multiplying a company's shares outstanding by the current market price of one share. The observed results of both the web services are represented graphically as shown in Figure 6.

In the graph LTA means-Last Trade Amount, CP-Change Percent, OA-Open Amount, PC-Previous Close, and SC-Stock Change. The analysis showed that the results of both the web services are same and accurate for each output parameter. Hence the web services are validated and verified for accurate results.

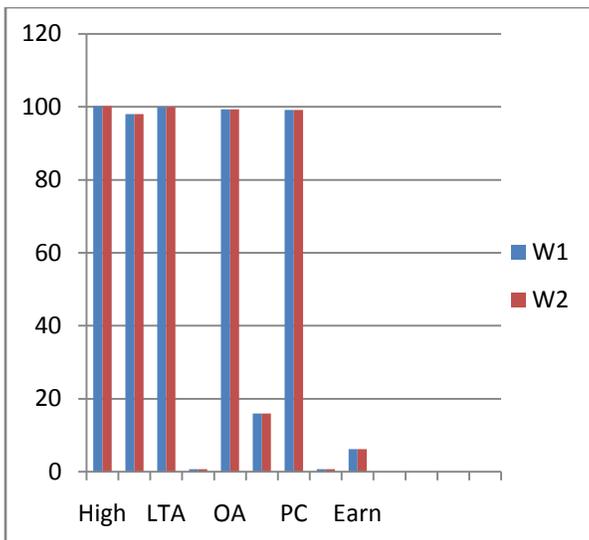


Figure 6: Graph of both web services results.

IV. CONCLUSIONS AND FUTURE SCOPE

Web services are application components which communicate using open protocols and are self-contained and self-describing. Web services can be discovered using UDDI for the use by other applications. It allows different applications from different sources to communicate with each other without time-consuming. HTTP and XML is the basis for Web services. Existing web service tools are either proprietary or open source software tools. The tools fetch results from a web service and display results which may or may not be accurate at a time due to wrong data available on the service site of web service. Therefore there is a need to have a tool which fetches information from two web services. wizdl tool was modified to incorporate the two services. The research work for finding the details of wizdl testing tool and enriching information about web services had many interesting issues out of which one is discussed in this paper. The paper focused on the evaluation of two web services based on common input. The results were related to stock market and displayed in a tabular form. For comparison the results were graphically represented for validation. The results showed that both services were accurate and equally efficient. The results can be still improved by taking into account few more services for testing the common input.

References:-

- [1] Myers Glenford J., "The art of software testing", New York, 2nd Edition 2004.
- [2] J. Mahesh Babu, P. Rajarajeswari² and A. Ramamohan Reddy, "Testing of web services Based on Ontology Management Service", "International Journal of Modern Engineering Research (IJMER)", July 2013.

- [3] Poonkavithai Kalamegam and Zayaraz Godandapani, "A Survey on Testing SOA Built using Web Services", "International Journal of Software Engineering and Its Applications(IJSEA)", October 2012.

- [4] <http://www.csd.uoc.gr/~hy565/newpage/docs/pdfs/papers/wsca.pdf>, "Web Services Conceptual Architecture", Retrieved on: 02 October, 2014

- [5] Asif Sohail Abid, Atif Sohail Abid, Muhammad Younus Javed, and Adnan Zahid, "Did you Find Any Public Web Service Description over the World Wide Web?", "Journal of Advances in Computer Network", June 2013.

- [6] Y.prasanth and L S S reddy, "Result and Statistical Analysis of Web Services – A Testing View", "Research Directions", August 2013.

- [7] Tanuj Wala and Aman Kumar Sharma, "A Comparative Study of Web Service Testing Tools", "International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSE)", February 2014.

- [8] <http://wizdl.codeplex.com/>, "Wizdl", Retrieved on: 02 October, 2014.

- [9] Tanuj Wala and Aman Kumar Sharma, "Improved Software Testing Tool", "International Journal of Computer Science and Mobile Computing(IJCSMC)", September 2014.