Volume 3, Issue 5, May-2016, pp. 261-265 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Improving Web Services Maintenance through Regression Testing

Divya Rohatgi, Prof. (Col) Gurmit Singh

1. Research Scholar, Computer Science and Engineering Department
Sam Higginbottom Institute of Agriculture, Technology and Sciences
Deemed University, Allahabad, India

2. Professor, Computer Science and Engineering Department
Sam Higginbottom Institute of Agriculture, Technology and Sciences
Deemed University, Allahabad, India

Abstract: - Software maintenance is considered to be the most expensive activity in software development and is used to ensure quality to the product. Regression Testing is a part of software maintainers which is done every time the software is changed. For software like Web services which represent a class of Service oriented architectures this activity is a challenging task. Since web services incorporate business functionality thus maintaining proper quality is an important concern. Due to inherent distributed, heterogeneous and dynamic in nature, regression testing is difficult and time consuming activity. Thus in order to reduce maintenance cost we have to reduce regression testing cost. Thus in this paper, we have given a comprehensive study of regression testing of web services exploring the challenges, approaches and tools used for them to ensure proper quality and inherently reducing the maintenance costs.

Keywords – Software Engineering, Software Maintenance, Regression Testing, Web Service.

1. INTRODUCTION

In today's growing and competitive scenario Service–oriented architectures are having a crucial role in the way in which systems are developed and designed. Basically, they represent an architecture in which small, loosely coupled pieces of functionality are published, used and combined over a network. The W3C consortium [1] describes web services as "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A web service supports direct interactions with other software agents using XML based message exchange via Internet protocols". Web services have become popular and the need of today business because they offer several advantages: [2]

1. Interoperability: interoperation of diverse systems within and across enterprises is certainly one of the key implications deriving from the adoption of the standardized stack of technology underlying web services (e.g., SOAP, WSDL, UDDI, etc.);

2. Separation of business processes from functions: web services allow for a "two-levels" programming approach, where business functions are exposed as services and business processes are implemented by orchestrating individual services into a workflow by means of some modeling languages (e.g., BPEL4WS [3]);

Despite of these advantages which Web Services offer, there are some problems also due to their heterogeneous nature which make traditional software engineering particularly testing a challenging task.

The key feature of SOA is the dynamic nature of the component and ability to change as per changing needs

which makes testing particularly regression testing a difficult practise. Regression testing is actually a maintenance activity which ensures that changes made to the system do not affect the previous tested system. In web services since actual configuration of the service is known only at run time, so It becomes very complex to verify whether the changes made in earlier version of the system are correct or not and it does not affect the functionality and performance of the existing system. In this paper we focus on regression testing of web services, which differs from regression testing of more traditional software.

The paper is organised as follows. Section II covers insights of Regression Testing. In section III a comprehensive list of different existing tools used for testing of web services is given. In the last conclusion covering future scope is covered.

2. REGRESSION TESTING

Regression testing is actually a maintenance activity and indispensible part of every software development and maintenance. Wikipedia defines regression testing as: "Software testing that seeks to uncover new software bugs or regressions in existing functional and non-functional areas of a system after changes, such as enhancements, patches or configuration changes, have been made to them. The intent of regression testing is to ensure that a change, such as a bug fix, did not introduce new faults. One of the main reasons for regression testing is to determine whether a change in one part of the software affects other parts of the software". An important issue in regression testing is how to reuse the existing test suite for the modified program. There are two main regression testing strategies; retest all, and selective retest [4]. Rothermel and Harrold [4] have identified two issues in the selective retest techniques: (1) the issue of how to select test cases from the existing test suite and (2) the issue of identifying where additional test cases may be required.

Rothermel et al. [5] consider three techniques for reducing the cost of regression testing. They are regression test selection, test suite minimization and test case prioritization techniques. **Regression Test Selection** These techniques attempt to reduce the cost of regression testing by selecting appropriate test cases using information from the certified program, the modified program and the existing test suite. These regression test selection techniques can be divided into few categories based on elements used in their techniques such as control flow based [4], textual differencing based [6;7], code entities based [8] and program slicing based [9; 10]. **Test Suite Minimization** Test suite minimization techniques decrease cost by minimizing a test suite that still maintains the same coverage of the initial test suite with respect to a particular test coverage metric. Harrold et al. [11] propose a minimization technique that helps to manage a test suite by determining redundant and obsolete test cases. **Test Case Prioritization** The prioritization technique let testers order their test cases, so that those test cases with the highest priority are executed earlier than those with lower priority according to some criterion [5]. Elbaum et al. [12] consider 14 test case prioritization techniques classified into three groups. The groups are based on control, statements and function level of a program. Web services are a class of SOA that represent essential business functionalities. Fig.1 [13] shows the architecture of SOA based applications

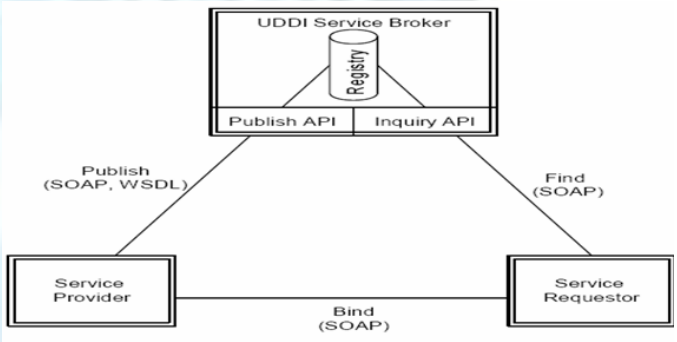

Fig.1 Web Service Architecture

Testing SOA is somehow an intricate and a challenging computing problem, and that is due to several reasons, some of which are outlined below [14, 15]:

1. SOA are distributed in that they are composed of web service components dispersed over different hardware and operating system platforms; thus, testing must cover the different deployment configurations.

2. SOA are dynamic in that they implement adaptive behaviours such as adding new services, integrating new services, and removing old ones; consequently, performing an effective regression testing can be a challenging task.

3. SOA are complex in that they can be seen as a mesh of interacting services each having specific functionalities and capable of different operations; thus, designing test cases for test automation can be a complicated and a demanding task

4. SOA are closed in that they are made out of closed services that run on the provider's side and clients have no control over their implementations; thus, preventing white-box testing methods that are essential to conduct exhaustive system validation.

5. SOA are remote in that their services are commonly located on the provider's server; and therefore, testing SOA can be costly, especially, if services are charged on a per-use basis. Moreover, service providers could suffer from denial-of service (DoS) in case of massive testing.

6. SOA are heterogeneous in that their services deliver no standard interfaces for intercommunication as they are built using incompatible technologies, platforms, and programming languages; thus, it would be necessary to build multiple types of test engines each pertaining to a particular service platform.

7. Various issues related to the testing of SOA-based application are test case management, testing tool requirements and evaluation criteria, testing the underlying implementation (e.g., Web services), testing quality attributes, evaluating the applicability of traditional testing techniques to new problems [16, 17].

8. Several factors, such as multiple runtime configurations, remote hosting of services, lack of access to service source code, and unanticipated changes in service semantics, present challenges in testing service-oriented applications [18].

9. Lack of Observability of Service code and structure: For users and system integrators services are just interfaces, and this prevents white-box testing approaches that require knowledge of the structure of code and flow of data [19].

10 Lack of control: While components/libraries are physically integrated in a software system, this is not the case for services, which run on an independent infrastructure and evolve under the sole control of the provider [19].

3. TESTING TOOLS

This section provides the comprehensive list of testing tools available for testing of web services which are as follows [20]

1. TestMaker testing tool: It is freely available tool and it is developed at the Department of Industrial and Organizational Psychology at the RWTH Aachen. It is web-based software for presentation, administration and evaluation of psychometric tests. First and foremost TestMaker is tailored to the needs of Web-based self-tests with performance feedback, but it can also be used in other online surveys. Using TestMaker neither programming nor HTML knowledge is required.

2. Soaptest: Soaptest is a testing tool suite for testing and validation in a Service Oriented Architectures. In this testing tool Basic testing functionality include static analysis through WSDL testing, functional unit testing, regression testing, security testing, and load testing.

3. SoapUI: SoapUI is an open source web service testing application for service-oriented architectures (SOA). Its functionality covers web service inspection, invoking, development, simulation and mocking, functional testing, load and compliance testing. A commercial version, SoapUI Pro, which mainly focuses on features designed to enhance productivity, was also developed by eviware software.

4. E - Test: Suite for Web services provides ways to generate Web services test scripts, validate XML responses, and identify performance bottlenecks by server-side monitoring.

5. IBM Rational tester for SOA Quality: A testing tool which is used for the various types of testing of SOA based application. The testing of the application is done through generating the script for various actions. This tool significantly reduces the time and effort required for the SOA based application. A functional and regression testing tool that enables code-free testing of GUI-less web services [9].

There are also other available tools as given below [19]

6. ANTS: Load supports testing Web services behaviour and performance under the stress of a multiple user load.

7. J-Blitz: carries out stress, performance, and functionality testing by generating different loading levels and records anomalies as they occur.

8. SOAP Scope: Supports testing SOAP transactions by monitoring communications among SOAP endpoints and analysing Web Services Description Language (WSDL) and SOAP messages against industry standards, such as Web Services- interoperability.

9. SOA Test: Supports WSDL validation and functionality, performance, and scalability testing. It features a collaborative work flow in which engineers create test cases that the quality assurance team can leverage into scenario based testing.

10. Web Service Tester: Is an integrated testing suite for functionality, performance, and regression testing Web services.

Apart from the above listed tools there are some others such as Push to Test and JUnit which are also used for testing SOA-based system [19].

4. CONCLUSION

Thus we have seen that Regression Testing in case of web services is a challenging task and requires a comprehensive approach because of the dynamic nature of the system. In this paper we have made an effort to highlight important issues involved in the Regression Testing of web services which can be fruitful in the automation of testing approach of web services in order to ensure proper quality and reducing mantainence costs.

REFERENCES

[1] W3C Working Group. Web services architecture requirements,2002, *http://www.w3.org/TR/ws-desc-reqs/.*

[2] Marcello Bruno, Gerardo Canfora, Massimiliano Di Penta," Regression Testing of Web Services"

[3] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, S. T. D. Smith, I. Trickovic, and S.Weerawarana, "Business Process Execution Language for Web Services", *.http://www-106.ibm.com/developerworks/webservices/ library/ws-bpel/*

[4] Gregg Rothermel, Mary Jean Harrold. "A safe, efficient regression test selection technique", ACM Transactions on Software Engineering Methodology (TOSEM), 6(2):173{210, 1997

[5] Gregg Rothermel, Roland H. Untch, Chengyun Chu, and Mary Jean Harrold, "Prioritizing test cases for regression testing. IEEE Transactions on Software Engineering", 27(10):929{948, 2001.

[6] F. I. Vokolos , P. G. Frankl, "Pythia: a regression test selection tool based on textual differencing", In Proceedings of the Internatinal Conference on Reliability, Quality and Safety of Software-intensive Systems(ENCRESS'97), pages Chapman & Hall, Ltd., 1997.

[7] Filippos I. Vokolos and Phyllis G. Frankl," Empirical evaluation of the textual differencing regression testing technique", In Proceedings of the International Conference on Software Maintenance (ICSM'98), pages,1998

[8] Yih-Farn Chen, David S. Rosenblum, and Kiem-Phong Vo, "Testtube: A system for selective regression testing", In Proceedings of the International Conference on Software Engineering (ICSE'94), 1994.

[9] David Binkley," The application of program slicing to regression testing", Information & Software Technology, 40(11-12):583{594, 1998.

[10] Lei Xu, Baowen Xu, Zhenqiang Chen, Jixiang Jiang, and Huowang Chen, "Regression testing for web applications based on slicing.",In Proceedings of the IEEE International Computer Software and Applications Conference (COMPSAC'03)

[11] Mary Jean Harrold, Rajiv Gupta, and Mary Lou ," A methodology for controlling the size of a test suite.", ACM Transactions on Software Engineering and Methodology (TOSEM), 2(3):270{285, 1993.

[12] Sebastian G. Elbaum, Alexey G. Malishevsky, and Gregg Rothermel., " Prioritizing test cases for regression testing.",In Proceedings of the ACM/SIGSOFT International Symposium on Software Testing and Analysis , 2000

[13] Rajani Kanta Mohanty1, Binod Kumar Pattanayak2 and Durga Prasad Mohapatra, "Uml based Web Service Regression Testing Using Test Cases: A Case Study", Vol. 7, No. 11, November 2012 ISSN 1819-6608 ARPN Journal of Engineering and Applied Sciences

[14]C. Bartolini, A. Bertolino, S. Elbaum, and E. Marchetti, "Whitening SOA testing ", in Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering, Amsterdam, The Netherlands, 2009, pp. 161–170.

[15] Scott Barber, "SOA Testing Challenges, Technical Report", PerfTestPlus, 2006.

[16] Tilley, S. R., Bai, X., and Lewis, G. A. First international workshop on service-oriented architecture testing (soat 2009). In ICSM'09 (2009), pp. 583-584.

[17] Kontogiannis, K., Lewis, G. A., and Smith, D. B. "A research agenda for service-oriented architecture", In Proceedings of the 2nd international workshop on Systems development in SOA environments (New York, NY, USA, 2008), SDSOA '08, ACM, pp. 1-6.

[18] Mani, S., Sinha, V. S., Sinha, S., Dhoolia, P., Mukherjee, D., and Chakraborty, S. ,"Efficient testing of service-oriented applications using semantic service stubs",. 2009 IEEE International Conference on Web Services (2009), 197-204.

[19] Prachet Bhuyan, Chandra Prakash, Durga Prasad Mohapatra," Survey of Regression Testing in SOA", International Journal of Computer Applications (0975 – 8887) Volume 44– No19, April 2012,

[20] Bharat Choudhary ,VineetRichhariya,"An approach of Regression testing for Service Oriented Architecture",International Journal of Research in Computer and Communication Technology, Vol 2, Issue 7, July-2013

