International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 30393-30405

http://www.ripublication.com

Cloud Testing as a Service (CTaaS) – Analysis, Design and Implementation

V. Janani [1] , Dr. K. Krishnamoorty[2]
[1]*Asstistant Professor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.*
[2] *Professor & Head, Department of Computer Science and Engineering, Sudharsan College of Engineering, Pudukkottai, Tamilnadu, India.*
[1]*janani.cse@adhiyamaan.in,* [2] *kkr510@yahoo.co.in*

Abstract

Testing is the most important way to improve the quality and reliability of the software. Software testing accounts upto 50% of the total cost of software development. Software testing in the cloud can reduce the need for hardware and software resources and offers a flexible and efficient alternative to the traditional software testing. It also enhances the proficiency of testing to improve the quality of the application. The deployment of software testing services in the cloud makes them to easily available for various companies and project. It also provides their on-demand usage. Cloud based testing tools gives the opportunity to expand the testing capabilities and break the chains that have kept for desktop computers for so long. Migrate the testing in to cloud helps the organization to get the latest tools, on-demand resources, not worrying about managing infrastructure and licensing at a very low cost. Using cloud testing the developers, testers and website managers can automate and speed up the testing process of their web applications with real browsers in the cloud. Cloud Testing comes under Software as a Service (SaaS) cloud delivery model. In traditional testing process an organization has to set up, maintain and verify the testing environment scenarios from end to end prospective in all aspects but in CTaaS(Cloud Testing as a Service) no need to setup any hardware and software it will be done automatically. The objective of this work is to evaluate performance gains achieved with the use of the framework showing that it is possible to improve the software testing process with very little configuration overhead and low costs.

Key terms: Software Testing, Cloud Testing, Performance Testing.

Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements. Cloud Testing is a means of testing cloud based applications that use resources found in the cloud. By leveraging a cloud computing solution for testing, organizations can shorten provisioning time because the cloud enables provisioning of test servers on demand. This helps ensure unused servers are not sitting idle.

Testing Platform on Cloud aims to provide the customer on-demand testing tools. Cloud based testing is cost effective testing with reduced overheads especially targeting small and medium scale clients and organizations. Testing as a Service delivers application testing services in a highly available, consumable, pay as you go model that provides flexibility in service and pricing. It also lowers the cost of entry to full service testing options and helps to implement best practice quality management processes.

Testing in the cloud or cloud testing can have three fragments:

- The system or application under test is accessible online. This might be SaaS software or non-SaaS Software. In addition, this includes testing at different test levels e.g. performance testing.
- Testing infrastructure and platforms are hosted across different deployment models of the cloud i.e. Public, Community, Private or Hybrid clouds.
- Testing of the cloud itself. Cloud environments should be tested and measured for their performance, availability, security and scalability in order to support efficient delivery of service

Cloud Testing is consummated in the industry in two ways:

- The first way is testing applications which have migrated or are to be migrated to the cloud so as to ensure that their performance, security and reliability matches or exceeds expectations in view of the changing delivery methods. It is known as Testing Cloud.
- The second way is leveraging the cloud based hardware infrastructure and computing resources to perform traditional Testing like performance, load, stress, security and compatibility testing for regular, on-premise applications, this is called as Testing using Cloud.

Cloud testing is simply making the use of cloud infrastructure to run and manage the test. This technology provides many benefits and mitigates all the problems faced in the traditional approach of testing. Riungu et al [2], points out that software testing as an online service provides on-demand service with a daily operation of maintenance and support through web browsers. Cloud means providing services; it obviously reduces the effort of an organization. Testing with the help of cloud avoids the purchase of expensive hardware and software and allows faster scaling and with a considerable price. It reduces the licensing and capital expense by 50-75% using the virtualized resources.

Cloud Delivery Models

Cloud as a service means delivering what the customer wants as a service over the network. Using these services customers can avoid their overhead costs that come as a part of maintaining the service. The service models include Infrastructure-based, Platform-based and Software-based services. Figure 1 shows the cloud delivery models and its corresponding services.

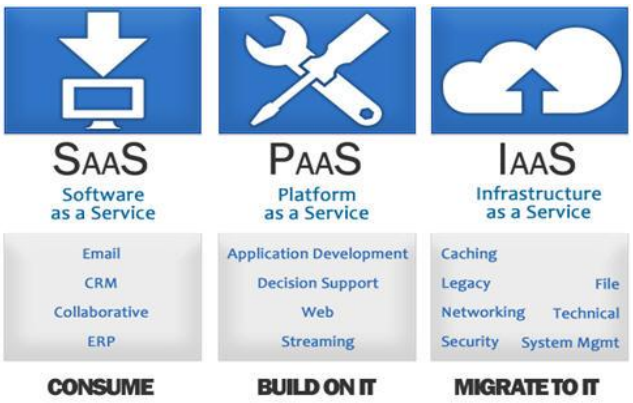

Figure 1: Cloud Delivery models

Infrastructure as a service

Infrastructure as a Service (IaaS) is a service providing model that offers hardware computing resources infrastructure to the clients. These resources might include virtual machines, data storage, firewalls, networks etc. that an organization might use to deliver their solutions. In this service, clients can run their own application on the resource pro-vided and are responsible for managing and security of the application by their own.

Platform as a service

In this model, an already built application infrastructure platform service is provided to the clients. This model provides a kind of API that manages the programming platform or other solutions for development. It usually supports different programming languages such as Python, Ruby, .NET languages and Java. PaaS are built on the top of infrastructure services and are high level abstraction to the cloud. This platform provides the framework for developing web based applications

Software as a service

Software as a Service is a model that provides application software for the consumers where they need not install any application to their local machine and need not purchase any hardware resources. Using this model, customers can be benefited as they need not manage the computing resources for the application and can use this application on demand.

cloud testing, which eliminates some of the difficulties encountered in the traditional approach such as infrastructure set up, expensive individual costs for

hardware and software and a lot of time and effort consumption required for environment set up. Cloud testing at its simplest utilizes the cloud computing infrastructure for software testing. It uses infrastructure based, platform based and software based cloud services to test the application by minimizing the cost and time with improved product quality. Testing with respect to a cloud will employ both traditional and new age methods to perform testing. Cloud testing can be broadly divided into four different categories

- **Testing of the whole cloud**: The cloud is viewed as a whole entity based on its features and testing is carried out based on that.
- **Testing within a cloud** : This is the testing carried out inside of the cloud by checking each of its internal features
- **Testing across clouds**: Based on specifications, here the testing is carried out on the different types of cloud like public, private and hybrid clouds.
- **SaaS testing in cloud**: Functional and nonfunctional testing is performed based on requirements.

Modern tests system has demands for realistic simulation of the production environment and this test cannot be accomplished from a single location. We need different tools to test multiple scenarios providing analysis, monitoring, scheduling and reporting of the applications. It needs considerable resources that might not be readily available. For example, an application might need to be tested with multiple scenarios like different operating systems, several database clients, multiple browsers and server interactions[4].

Cloud Testing Life Cycle

Cloud computing has opened up new opportunities for software testing, which provides unlimited resources with scalability, flexibility and availability of distributed testing environment. It reduces the execution time of testing of large applications and lead to cost-effective solutions. Testing is not a single activity but a series of planned tasks. Cloud Testing is defined as testing as a Service. Figure 2 shows the Cloud Testing Life Cycle processes.

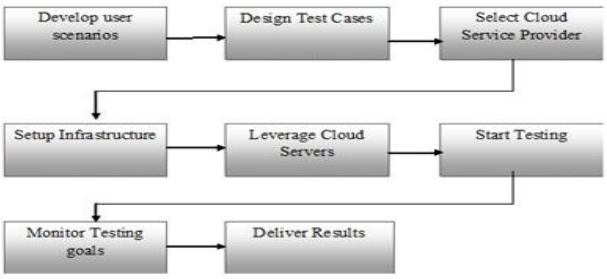

Figure 2: Cloud Testing Life Cycle

Develop User Scenarios
When designing and developing software or web applications three important terms are considered, they are user requirements, scenarios and use cases. Create set of use cases to describe how the application/sites will solve the user's scenarios.

Design Test Cases
The test case acts as a starting point for the test execution, which has a set of test data, pre- conditions, expected results and post- conditions. Test cases are designed for a particular test scenario in order to verify compliance against a specific requirement.

Select Cloud Service Provider
The number of managed cloud service providers offering hybrid cloud services continues to expand from the basic infrastructure, platform and application services to vertical industry solutions and enterprise architecture. The hybrid cloud service pack of managed services will provide migration tools to enable fast relocation of existing services between dedicated, private cloud and public cloud infrastructures, that may include virtual servers and SAN storage, integrated on-demand servers, dedicated databases, firewalls, load-balancers, content delivery network (CDN) and streaming services. Leading global providers include Amazon, Verizon Terremark, Rackspace and CenturyLink.

Setup Infrastructure
Here the required hardware computing resources are selected and setup by the client.

Leverage Cloud Servers
More and more organizations are demanding that their software tools work where they do – in the office, at home, and on the road. Cloud-based software delivery makes this much easier. Depending on the solution, many of which are web-based, a user can access systems anywhere they have internet connectivity. Many cloud-based solutions are app enabled for smartphones including iPhone and Android as well as tablet operating systems too.

Start Testing
Once all the setup and initialization is over in the cloud then start testing the application using any of the available open source software testing tools.

Monitor Testing Goals
The purpose of controlling and monitoring in the testing process is to provide visibility to its associated activities and to ensure that the testing process proceeds according to plan.

Deliver Results
After completing the testing the reports will be generated for further analysis.

Cloud Testing Types

Testing in a cloud has to not only ensure that the functional requirements are met, but a strong emphasis needs to be laid on nonfunctional testing as well. Let's take a look at the different kinds of testing that are performed.

Functional Testing

Functional testing is a type of testing that verifies whether the application meets the functional system requirements. This applies usually to the system and especially the UI level functional testing where the detailed implementation of the system is ignored and the system is only tested focusing on how it was supposed to work according to the specification and design documents. Functional testing is an important step in software development life cycle which ensures that the product works as the user's expectation. The traditional approach of functional testing uses a systematic process including requirement definition, test planning, test design, test execution and analysis. Functional testing can be carried out using manual or automated testing. Functional testing on the cloud environments is easier than in the traditional environments as they provide a very easy setup where one can run many test suits in parallel. It also provides a faster and more efficient way of testing in multiple browsers for web applications in desktop and mobiles. Some of the functional tests are described below:

System Verification Testing

This makes sure that the various modules function correctly with one another, thus making sure that their behavior is as expected.

Acceptance Testing

Here the cloud based solution is handed over to the users to make sure it meets their expectations.

Interoperability Testing

Any application must have the flexibility to work without any issues not only in different platforms, but also must work seamlessly when moving from cloud infrastructure to another.

Non-Functional Testing

Non-functional tests mainly focus on the web application based tests ensuring that they meet the desired requirements. Here are few forms of nonfunctional tests discussed below:

Availability Testing

The cloud supervisor/ vendor have to make sure that the cloud is available round the clock. As there could be many mission critical activities going on, the administrator has to make sure that there is no adverse impact to the consumers.

Multi Tenancy Testing
Here, multiple users use a cloud offering. Testing must be performed to ensure that there is sufficient security and access control of data when multiple users are using a single instance.

Performance Testing
Verification of the response time needs to be done to ensure that everything is intact even when there is a large amount of requests to be satisfied. The network latency is also one of the critical factors to evaluate performance. Also workload balancing needs to be done when there is a reduction in load, by decommissioning resources. Thus load and stress testing are done in the cloud offering to make sure applications are performing optimally with increase/decrease in load and stress.

Security Testing
Since with the cloud everything is available anytime, it's essential to make sure that all user sensitive information has no unauthorized access and the privacy of users remains intact. When maintaining the applications in cloud, user data integrity must also be verified.

Disaster Recovery Testing
As already stated in availability testing, the cloud has to be available at all times and if there are any kind of failures like network outages, breakdown due to extreme load, system failures, etc, measure how fast the failure is indicated and any data loss during this period.

Scalability Testing
Test to make sure that offering has the capability to provide scale up or scale down facilities as per the need.

The Problem Statement

At the software industry, cloud computing has become more popularized because of its flexibility, scalability and reduced costs. Using the cloud for testing is immensely helping organizations to acquire the required tools, software licenses, infrastructures at a very low cost without having to set it up themselves and then worry about its maximum utilization. Testing the performance of the system under peak load is highly desired since they intend to prove the capability of the system under test (SUT). They are faced with a major challenge because carrying out such tests can require running tests for several days in order to massively load the system. This scenario requires a complex testing framework that must be supported with a relatively large amount of computing resources. Cloud computing offers several advantages for resolving this problem.

Performance Testing is a process of evaluating system's behavior under various extreme conditions. The main intent in Performance testing is monitoring and improving key performance indicators such as response time, throughput, memory,

CPU utilization etc.Performance testing is in general, testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. Performance Testing is used to evaluate the application's Reliability, Scalability & Interoperability. The major type of Performance Testing includes:

- Load: Modeling the expected usage by simulating multiple users accessing the programs service concurrently.
- Stress: Determining the stability of a given system testing beyond normal operational capacity.
- Reliability: Determines how long the application can sustain optimum performance levels under expected loads.
- Scalability: Determines how long the application can scale up, be it user load supported, no. of transactions, data volume etc.
- Interoperability: It is a property referring to the ability of diverse system and organizations to work together (inter-operate).
- Volume: Testing a software application for a certain data volume.

Performance testing is a type of testing which measures the system throughput and latency, indicating how well a software system meets its requirements under certain workload. It tests whether or not the software has met speed, scalability and stability requirements under expected workload with a variety of concurrent users. It determines how much traffic a website can handle, and ensures that applications are ready for a market launch. Before launching any application, it is very important to identify the performance accepting criteria of a system to know the number of visitors supported in a site and how the response time and error rates are handled at the arrival of a large number of users. Performance testing is responsible for testing the load, stress and scalability of any system.

The traditional approach of performance testing was carried out in an internal environment in a company which required servers, hardware, and production software to host the system under test that is quite expensive. So, performance testing in cloud addresses the issues encountered in the traditional model. Cloud computing can carry out performance, scalability, and stress testing in a timely and economical manner. Organizations performing this testing probably get the bigger benefit from cloud computing, considering both the economic aspect and scalability .Testing in the cloud means paying only for what is needed for the required time. This means it is no longer necessary to purchase expensive hardware and software with annual licenses. In addition, configuring the test environment becomes very easy and almost instant, allowing for faster test execution which has the potential to increase the number of test sessions that can be run in the same time period .

The actual resources used by the application may grow or shrink based on the application load. Performance testing in cloud allows us to test the maximum number of users that are allowed to access the web from different geographic locations in the real environments. The cloud model amplifies elasticity in the application platform. There might be a massive increase in the Internet traffic at some point, in terms of number of users, where an application might get hit from all corners of the world. At this point, application should be able to handle the maximum number of users without any error and degrading its speed. Performance might also change because of the

traffic load coming from different geographical locations. Poor application performance directly impacts customer loyalty, reduces revenue and damages your brand. Therefore, effective Web service load testing is very important for evaluating the performance of the component Web service applications.

The load testing results can significantly help service provider/developer with improving the performance of the web applications. There are three challenges that Web service load testing can face. They are as follows:

- Simulation of real characteristics of massive user requests, including real concurrency, diversity of geographical location, and system configuration.
- Flexible and elastic provisioning of testing environments, consisting of underlying resources and runtime middleware.
- Automating the load testing process.

The goal of load testing is to determine both the performance and scalability of a Web application. The most important performance parameters commonly used to measure web applications are:

- Response time: It is the time a system takes to respond to various types of requests. It is a function of load intensity, which can be measured in terms of arrival rates (such as request per second) or number of concurrent requests.
- Throughput: It is the rate at which a system or service can process requests, i.e. the request is completely processed by the server. It is normally measured in transaction per second (tps). The throughput is a function of the load assigned to a system and of the maximum capacity of the system to process work.
- Availability: It is the fraction of time that any system is operating and available to its users. The two main reasons for systems to be unavailable are failures and overloads.

Implementation

Testing as a service (TaaS) offers a large pool of benefits. Testing in the cloud makes testing environment available, visible and makes its control and automation possible [6]. Using testing as a service in the cloud makes all the testing resources pooled and virtualized which provides an efficient implementation that is independent of any infrastructure. Besides, users and testers can benefit from the elastic scaling of this service which means that the testing environment can be scaled up or down depending on the user's needs. The testing cloud service offers the maintenance of a set if test beds which assures a complete assurance of the application maintenance. Using testing as a service over the cloud reduces the cost since the pricing policy is "pay as you use" which makes the pricing of the testing tools tailored to the user or tester's need.

Working over the cloud, the configurations and environment takes few minutes compared to the configuration on the premise that takes more than a day [6]. Over the cloud, the configuration parameters for the network can be personalized and automated which is not possible while testing on the premise where the tests are generally manual.

Testing tools that are used to test the conventional applications needed to be re-evaluated when applied to applications hosted in a cloud in order to take into consideration network analysis, application and environment changes implied by the cloud deployment, and the interoperability of the applications towards other platforms and infrastructures [9]. Besides these conventional tools, there are specific tools that were developed to test cloud services and applications. In this part of the paper, we will go through selenium which is very distinct and open source tools that commercialized for cloud testing.

Selenium

Selenium is a known and widely used open source test tool [12]. It enables users and testers to perform functional tests of their applications at the selenium Integrated and Development Environment (IDE). It is specialized on web applications and tests them over various browsers (Firefox, Chrome, and Internet Explorer) [10]. It provides functional tests, load and performance test. It reduces the test operating costs. It also provides browsing compatibility tests and multiple operating environments. In addition, it does the transformation from selenium tests to java unit tests and provides writing tests in different kind of languages such as: C#, Java, Python, etc. Selenium provides the following components: Selenium IDE, Selenium Client API, Selenium Remote Control, Selenium Web Driver, and Selenium Grid. Figure 3 shows the multiple features of selenium open source testing tool.

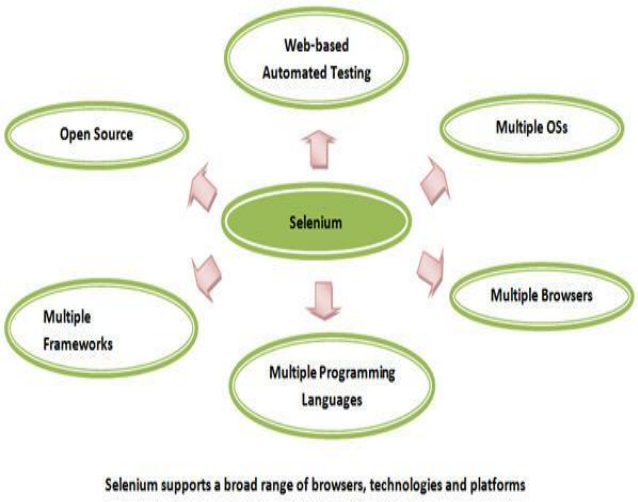

Figure 3: Features of Selenium

The following figure 4 shows the actual set up for the cloud based testing environment.

CTaaS Infrastructure

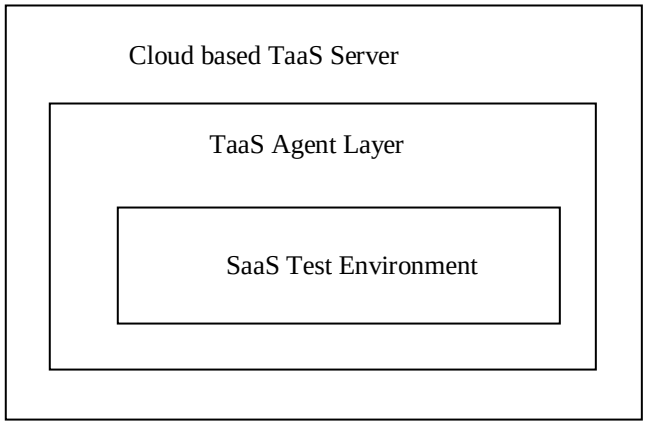

Figure 4: CTaaS Setup

In terms of costs, the CTaaS model is beneficial since enterprises pay for the actual time utilized for testing a leading to controlled costs. Compare this to a situation where an enterprise manages its own infrastructure and has to incur capital expenditure and yearly depreciation costs on testing environments.

The CTaaS model also offers licensing benefits since test tools, hardware, application licensing or even operating platform (unix, linux etc) are managed by the cloud. Additionally, using standardized testing processing and tools can yield a 10%-20% cost reduction due to increased quality and Test Automation from cloud deployments show a 5%-10% revenue enhancement. Enterprises also witness productivity gains of 5%-10% year-over-year due to test method improvements. Leveraging a global talent pool through an extended cloud ecosystem can lead to a 10%-20% savings in personnel costs.

By switching to CTaaS, customers get access to a centralized test environment, with standardized software library and test suites. It also has a self service portal, which cuts down time required to provision test environments. A Role based access control (RBAC) provides access to different functionalities of TaaS, based on user roles.

Algorithm for testing a web application using selenium with TestNG is as follows:

Step 1: Gather the functional flows from Client.

Step 2: Plan the selenium framework and test plan Preparation.

Step 3: Do Software installation and Test environment setup in the cloud.

Step 4: Create and enhance test scripts using selenium IDE using XPath Technique.

Step 5: Convert Scripts to TestNG using Eclipse.

Step 6: Execute all tests in selenium Web Driver with Eclipse as an IDE.

Step 7: Run the Scripts in multiple browsers using Web Driver.

Step 8: Run all Tests for every build and multiple builds.

Step 9: Generate HTML /XML Reports and publish the graphs.

Conclusion and Discussion

Testing-as-a-Service has demonstrated significant improvements over traditional testing environments. A most important benefit of using the Testing as a service, especially a Public Cloud is that it is a highly scalable model; a major progress as compared to an internally managed model. Cloud testing uses cloud infrastructure for software testing. It is a subset of software testing in which simulated, real-world Web traffic is used to test cloud-based Web applications. Cloud testing also verifies and validates specific cloud functions, including redundancy and performance scalability. When deploying testing configurations, the environment is created using certain specific tools. These testing tools can be easily deployed on a public cloud at a faster rate than in the enterprise's own environment as it requires customized testing. A public cloud deployment is also much quicker compared to testing in an internal environment as it has a standardized hardware and computing power.

References

[1] Leah Muthoni Riungu, Ossi Taipale, Kari Smolander, "Software Testing as an Online Service: Observations from Practice," In Third International Conference on Software Testing, Verification, and Validation Workshops (ICSTW), 2010, 418-423.

[2] Leah Muthoni Riungu, Ossi Taipale, Kari Smolander, "Research Issues for Software Testing in the Cloud",CloudCom, IEEE Second International Conference,2010,557-564.

[3] T.Vengattaraman, P.Dhavachelvan, R.Baskaran, "Model of Cloud based Application Environment for Software Testing",International Journal of Computer Science and Information Security,7(3),2010.

[4] T. Parveen, and S. Tilley, "When To Migrate Testing to the Cloud," In proc. 2nd International Workshop on Software Testing in the cloud (STITC), 3rd IEEE International Conference on Software Testing, Verification and Validation (ICST), April 2010, pp. 424-427, doi:10. 1109/ICSTW. 2010.

[5] Jerry Gao, Xiaoying Bai and Wei-Tek Tsai, "Cloud Testing- Issues, Challenges, Needs and Practice",SEIJ 1(1),2011,9-23.

[6] A.Vanitha Katherine, K.Alagarsamy, "Software Testing in Cloud Platform: A Survey", IJCA, 2012.

[7] Bouchaib Falah,Abir-Ezzahara El Ayadi,Othmane Atif, "Roadmap to Testing the Cloud Techniques, Methodology and Tools, IJCSET 3(3),2013 84-89.

[8] Sergiy Vilkomir, "Cloud Testing: A State of the art Review", Information and Security, an International Journal, 28(2), 2012, 213-222.

[9] Yatendra Singh Pundhir, "Cloud Computing Applications and Their Testing Methodology", International Journal of Software Engineering, 2(1), 2013.

[10] White Paper by Cognizant, "Taking Testing to the Cloud".
[11] White Paper by Infosys, "Cloud Testing Vs Testing Cloud".
[12] White Paper on, "Testing Services through Cloud",
[13] White Paper on, "Performance Testing in the cloud: A Pragmatic Approach".
[14] White Paper on, "Test Automation Tool Comparison – HP UFT/QTP Vs Selenium.
[15] http://www.toolsqa.com/selenium-webdriver/testng-reporting/