IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 7, July 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

Cloud Based Application Testing

Sayed Suhel Mohd. Salim

Assistant Professor, Anjuman-I-Islam's IHMCT, Mumbai.

Abstract
Cloud computing has gained considerable interest in recent years as a new paradigm for developing and delivering computing applications and services via existing affordable infrastructures. Software development process has changed under the impact of cloud computing, software testing is also part of this process. Cloud testing is a rapidly developing area of research in software engineering. It's ironic that the compressed development cycle for SaaS practically increases the overall testing effort. In this paper we introduce a new framework for cloud testing that is based on the ISTQB standard framework and according to the requirements and cloud testing steps. The framework include developing test scenarios, test cases design, select cloud service provider, setup infrastructure, leverage cloud servers, start testing, monitor test progress, test report and finally test closure. Each of these steps are included some activities.

Keywords: *Cloud computing; cloud testing; cloud based application; SaaS; testing framework*

1. Introduction
Cloud computing has gained considerable interest in recent years as a new paradigm for developing and delivering computing applications and services. Cloud computing affects all stages of software life cycle, including software testing. Similar to an acceptance of standard terminology such as SaaS, PaaS, and IaaS in cloud computing literature.

Nowadays applications are presented to run on multiple platforms, including portable and virtual environments, so testing them is more complex and expensive. Validation of applications requires different tools, according to aspects of functional and non-functional. Since cloud testing is a new direction in information technology ,there is no standard in this area. Main challenges for SaaS can be noted scalability, reliability and performance.

SaaS Testing refers to the methods used to ensure that applications built using the software as a service model of development function as designed. SaaS Testing occurs after a specific iteration of the SaaS Development Process has been brought to closure. The SaaS Development Process and SaaS Testing are driven by the demands of a fast moving market of competing software services. Likewise, SaaS Testing and the SaaS Development Process must both use agile methods to achieve their intended goal of delivering quality software in a timely manner, to a very competitive marketplace [3, 6].

SaaS has gained huge popularity in the last couple of years, with an increasing number of enterprises adopting it mainly due to the benefits like pay per use and on demand service. SaaS applications entail thorough testing for their integrity, different from that of on-premise applications. This involves testing of business logic, security, data integration, performance, and scalability, among others. Some of the challenges involved in testing SaaS applications are mentioned below [4].

1. Security and Privacy Testing
2. Short Notice Period and Frequent Releases
3. Performance testing
4. Integration and Migration
5. Business Knowledge
6. Licensing

This paper is structured as follows. In section 2, a few related works on cloud testing are reviewed. Then in section 3, the

test plan for cloud applications is expressed. Later on, we introduce a new framework in fourth section. Then in fifth section we evaluate our framework. Finally, we come to conclude the paper.

2. Related Work

Cloud testing is a rapidly developing area of research in software engineering. The first research worked on cloud testing appeared within recent years. However, it is too early to talk about significant success or fundamental research and advances in cloud testing.

A couple of new testing frameworks for cloud recovery, FATE (Failure Testing Service) and DESTINI (Declarative Testing Specifications) were suggested[1]. FATE systematically tests cloud systems using various scenarios of multiple failures. DESTINI is designed for clear and precise specifications of recovery procedures.

In this paper, a framework is provided that includes some test services. This framework looks at part like manufacturing. This system is composed of three parts:

1. Management Services
2. Test Clouds
3. Set of services tester

These parts runs on separate machines. That causes the high cost of implementing this. Apart of cost, this model involves dependency of all parts. This means that if one part fails others can't do their jobs too. For example, the Service Manager is responsible for monitoring and control activities. If machine that responsible to do process of a part, that can't do its job properly. Not only this part can't provide inaccurate information but also other parts can't do their tests.

C-Meter is a framework for measuring efficiency in cloud systems[7]. It has developed to reduce the cost of allocating and release of resources. Also, a scheduler is intended to reduce costs. The following figure shows the architecture of the framework. Parts of this approach are as follows:

1. Core controls and monitors inputs. The scheduler has been placed in this part.
2. Controller that control interactions between clouds.
3. Utilities modules that configure and develop processes.

In [8] has been introduced a model to evaluate cloud-based software by graph, also is a method to test cloud based software. The test parts of this method are as follows:

1. Monitor the proper use of the available resources.
2. Monitoring the scalability of cloud computing.

In the paper [9], is presented an automated framework to test security of cloud based software for Android OS. Parts of this framework are as follows:

1. Identify input.
2. In this method is displayed abnormal and suspicious behavior through an output to user.
3. Upon receipt the input data search through the sets of repositories for the appropriate test cases.
4. Monitoring part is intended to control progress of the software.
5. If an error is stored in repositories that contain errors, refer to confront the observed error is detected.
6. Finally, a report is prepared that shows how the system is developed and demonstrated.

3. Cloud Test Plan

Cloud testing is a new form of software testing in which web applications that used cloud computing environments seek to simulate real-world user traffic as a means of load testing and stress testing web sites (as well-known examples in addition to others). Using cloud testing gives users unlimited resources, paying only for what users consume [17].

To provide a common set of standardized documents the IEEE developed the 829 Standard for any type of software and software-based systems testing. This standard identifies test process, software and other features like accuracy, stability, and testability [14, 15, 16]. The purpose of this standard is to:

- Establish a common framework for test processes, activities, and tasks in support of all software life cycle processes, including development, operation, and maintenance.

- Define the test tasks, required inputs and outputs.
- Identify the minimum test tasks related to integrity levels for a four-level integrity scheme.
- Define the contents of the Master Test Plan and the Level Test Plan such as component, integration, system, and acceptance test.
- Define the contents of document that is related test (Test Design, Test Case, Test Procedure, Anomaly Report, Test Log, Level Test Report, Interim Test Report, and Master Test Report)

This standard is used for SaaS testing. The following test plan is designed for SaaS testing.

1) Test Plan Identifier
2) References: List all documents that support this test plan.
3) Introduction: The test is designed for cloud applications. The test strategy and test cases are extracted their requirements.
4) Test Items: Testing cloud applications includes all of its features software, network and infrastructure[10, 11].
   - **Software:** Testing focuses on aspects like testing component functions, business workflows, browser compatibility and data security, data integrity and data access privileges.
   - **Network:** Testing of the network needs to cover aspects like testing of various network bandwidths to ensure availability of data, and its transfer.
   - **Infrastructure:** When it comes to testing SaaS applications, tests are conducted on infrastructure. As this is likely to have a great impact on the end user experience. Infrastructure testing include live upgrade, disaster recovery tests. The emphasis here is on testing backups, storage policies and secure connections.
5) Software Risk Issues: There are some software risks such as safety, multiple interfaces, impacts on client and etc.
6) Features to be tested: Figure 1, shows the features should be tested for SaaS applications.

Business Testing	Security Testing	Performance Testing	Compatibility Testing	Live Testing	SaaS Attribute Testing
Manual / Automated Testing	Application Security Testing	Scalability Testing	Multi-Browser Testing	Disaster Recovery Testing	Multi Tenancy Isolation Testing
Exploratory Testing	Network Security Testing	Volume Testing	Localization Testing	State full Scenario Testing	API Integration Testing
End to End Business Work Flow Testing	User Access & Roles Testing	Accountability Testing	Accessibility Testing Mimicking Access from Remote Locations	Live Upgrade Testing	Billing Mechanism Testing

Automation Regression Testing	Data / Security Integration Testing	Reliability Testing	Internationalization Testing
Data Integration, Data Migration Testing	Compliance Testing	Load Testing for Single Instance	Interface Backward Compatibility
Checklist Validation	Identity Federation Mechanism Testing	Load Testing in a Instance Loaded Environment	

Fig. 1 Features to be tested in cloud application

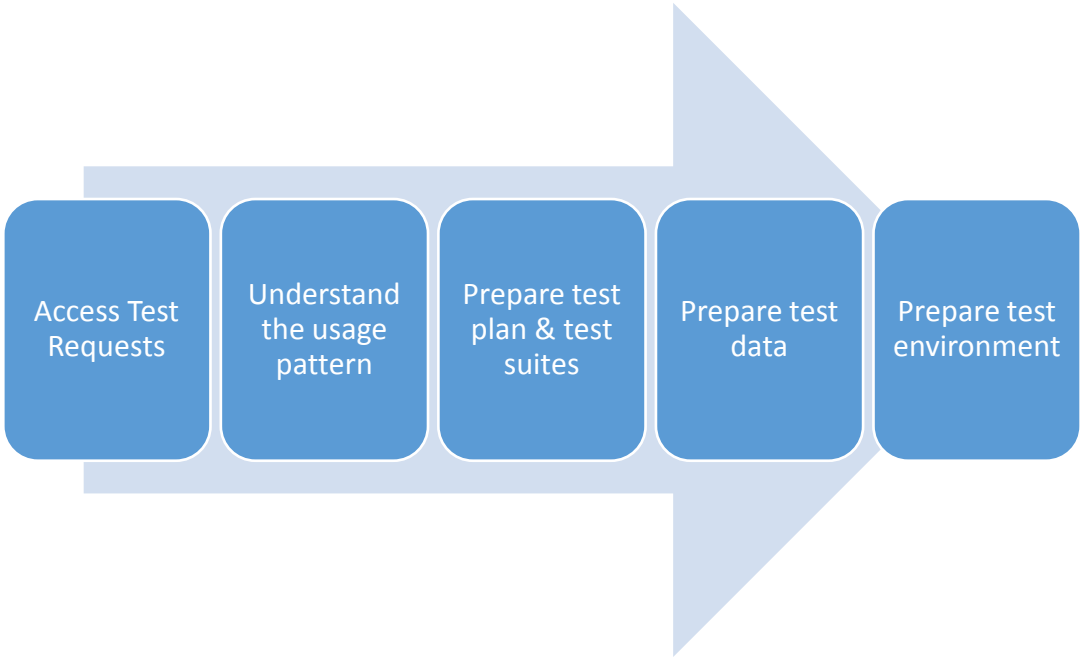

Fig. 2 SaaS Testing Process

7) Features not to be tested: Testing resource constraints force us to ignore some aspects of the Testing.
8) Approach: See figure 2 to get information about process of SaaS testing.
9) Item Pass/Fail Criteria: Including the acceptance criteria can be cited response time, transaction rate, and load and resource usage.
10) Suspension Criteria and Resumption Requirements: Know when to pause in a series of tests. If the number or type of defects reaches a point where the follow on testing has no value, it makes no sense to continue the test.
11) Test Deliverables: That includes test plans document, test cases, test design specifications, tools and their outputs, simulators, static and dynamic generators, errors log and report, execution and corrective actions.
12) Environmental Needs: Prepare software and hardware requirements.
13) Staffing and Training Needs: Educating system under test and testing tools. It also determines what should be tested and who is responsible for it.
14) Responsibilities: Determine who is responsible for what.
15) Schedule
16) Planning Risks and Contingencies
17) Approvals
18) Glossary

4. Brief overview of software Testing using ISTQB Framework

As it was said, ISTQB is an international comprehensive framework to test the software. Despite its comprehensiveness, it investigates the test generally and is not dependent upon the structure of software [12]. This issue has this advantage for ISTQB that it can be used as a reference to test each model of software but due to generality it can't deal with all the details achieved of various architectures of software. ISTQB investigates the test aspects in the following domain:

- Layout of the test in system life cycle
- Test process
- Test management
- Test risks
- Test techniques

As our discussion about the test process issue we don't enter other aspects but test process is affected by life cycle, the life cycle is investigated.

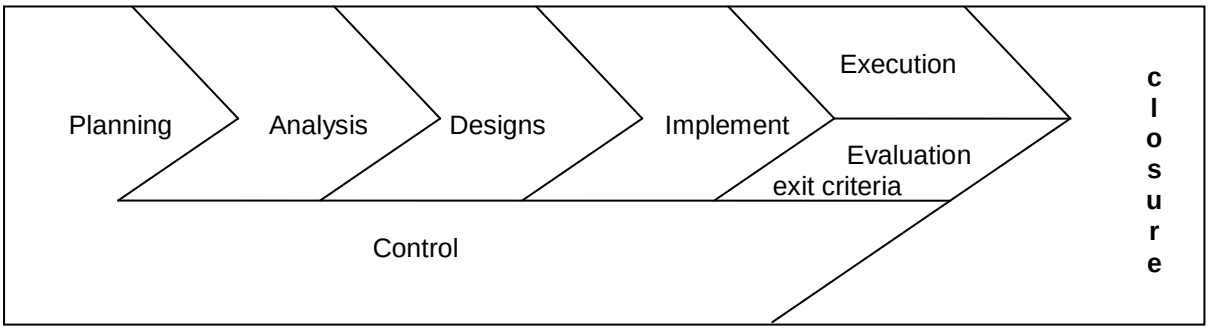

Fig. 3 Test process in ISTQB

Software test can't be considered as a separated activity of system development [2] otherwise we should pay considerable cost for the existing problems in the system development. ISTQB proposes V-model to test in system life cycle and at first it introduced system development levels based on ISO-IEC 12027 and then the test was proposed equal to each of the levels.

ISTQB divides test process as the following stages (Fig. 3):

- Test planning and control
- Test analysis and design
- Test Implementation
- Test evaluation
- Test closure

Logically, the above activities (except control) are done consecutively but based on the project can be done parallel and repetitious [13]. In the following we introduced each briefly.

Test planning and control: In this stage, the test strategy is selected and defines how this strategy is used in testing. After the planning here in test control, we define how we proceeded based on the plan. It is obvious that test control continues parallel with other activities and from the beginning of test process to the end of test process.

Analysis and designing the test: In this stage, we should design the test cases. Indeed, in this stage, the general goals of test are turned into tangible plans and the test infrastructures and tools are determined.

Test Implementation: In this stage, the test cases are applied and execute.

Test evaluation: In this stage, we evaluate to what extent, the test was successful in detecting the errors.

Test closure: In this stage by finishing the test and delivering the results, we analyze the extracted data of test.

5. Our proposed Framework for Cloud Testing

In this section we introduce a new framework for SaaS applications testing. This framework is presented due to basis the ISTQB and the cloud testing steps. Figure 4 gives a structural overview of the constituent components of our software testing framework.

Then more details is expressed that activities will be performed at any step. In the development test scenario phase be done following steps:

1) The first step is to identify the trend test (That means the testing purpose of system has been specified). These goals can be grouped into some categories such as performance, security, and etc.
2) In the determine test scenario step is studied several issues such as the environment, required resources and other factors.
3) After examining test scenario is collected the required information about the costs, economic feasibility, availability of resources.

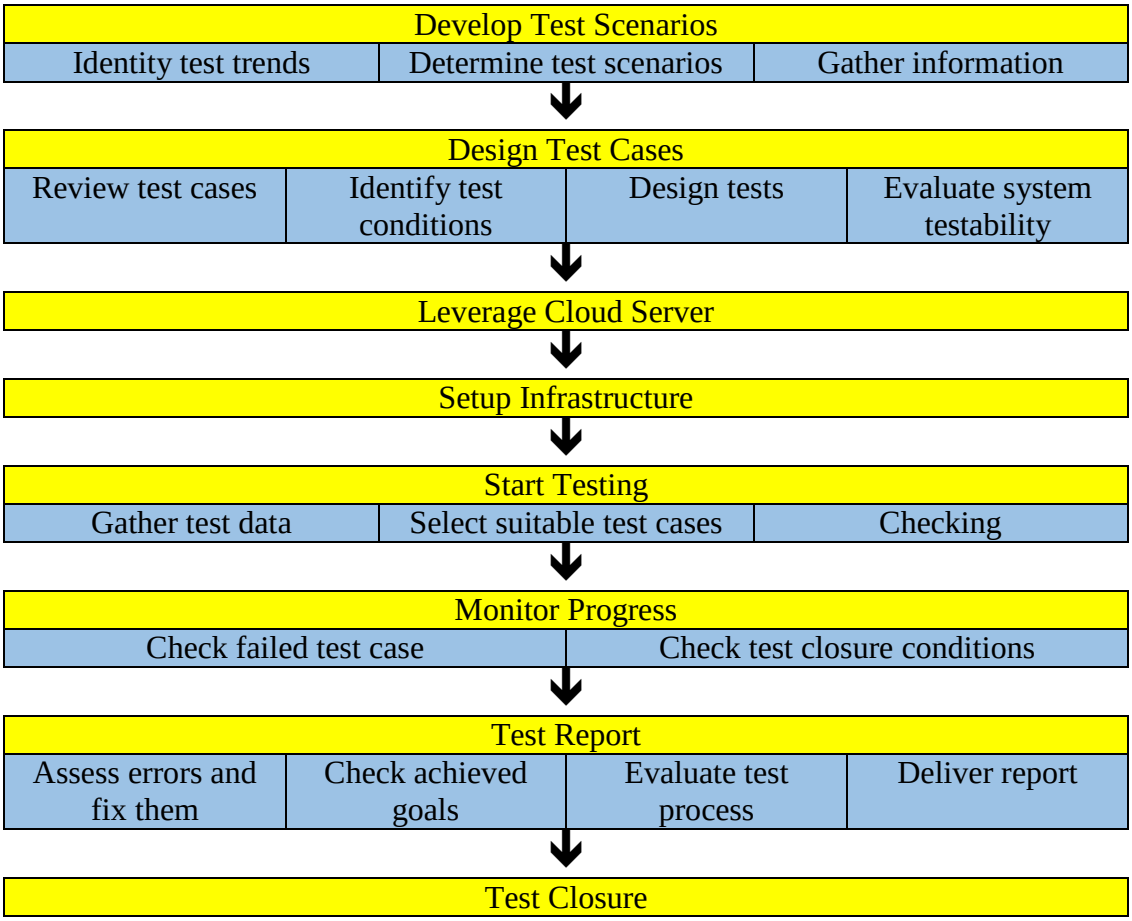

Fig. 4 Overall architecture of proposed testing framework

Some points should be considered in the design test cases that cause the test to cover all factors. For this reason the following steps is essential:

1) Review test cases
2) Identify test conditions
3) Design tests
4) Evaluate system testability

Selecting the cloud server and installation cloud-based infrastructure is performed based on organization's needs, stakeholders and test team. The start test step consider the fact that have collected enough data to test the feasibility of whether or not. A set of test cases considered, those are selected that are in line with the trend test. This step includes the following activities:

1. Gather test required information to

test
2. Select suitable test cases
3. Check and evaluate the log file

Monitor progress step includes the following activities:

1. A review of states that have failed at runtime.
2. Check test closure conditions.

The test report covers the following steps:

1. Problems assessment and ensure that they meet.
2. Check whether the goals have been achieved or not?
3. The overall evaluation test process and present report to use in the future.
4. Prepare report for organization that use the system under test. Also prepare a report for the backup team.

The review and examine results is the final step of presented framework. The log file is used to review and study the results. The following points can be taken of log file:

- What goals have failed?
- What goals have been achieved?
- Strategies should be adopted that failed goals considered as achieved goals.

This phase includes the following activities:

1. Parts of the system that have been tested.
2. Parts of the system that have not been tested.
3. The number of errors that have been discovered.

6. Evaluation the proposed framework

The advantages of presented framework towards the framework is introduced in paper [5] are follows:

- Monitoring and control perform in several such as start testing, monitor testing progress, review and report results.
- In this method there is no mention of the testability of system. It also show that the test system, the control of the system. The test system is more, it can be further analyzed to enhance system efficiency and better output than it has received.
- In this model are not prepared reports of processing and system problems. This is one of the major weaknesses of that. Report prepared by the test can be used to improve the system, finding defects and it is useful for those who are supposed to benefit from system.

The proposed method is compared to the C-Meter framework has the following advantages:

- In the proposed framework, we collect information in developing test scenarios and start testing steps to have adequate and complete understanding of requirements. This leads to the situation that increases the cost of processing such as allocating and freeing the resources will be avoided.
- Design test cases step can be helped in the controlling the system.
- In this system, there is no unit to prepare reports and log files. This framework is based on ISTQB standard, so we are not only prepared the report, we also use them to check the progress of the work and present it to the user. The statistics of utilities for sending feedback to the system, we use the log file to do this job.

The superiority of the proposed approach

to this method include:

- In this paper, focusing on a graph-based and design and testing are separated from each other while the design of a cloud application can be very effective on testing it. In our model, the process of data collection and design test cases can help to modelling. Also checking system testability is included in the testing step. This helps designers to develop the system so that it can be monitored and assessed.
- The method presented in this paper shows no sign of reporting on the progress of the application and check it.

Reasons for the superiority of the proposed method in this way, we can mention the following:

- In this method, a separate section identifies the input. So it costs a lot in the process. But in our method, we study and collect data before system implementation, and classify inputs and let them separated.
- In this system only errors is reported and stored in a repository and don't occur any process to correct them. In our method, we check failed state to correct it.

7. Conclusion

We have developed a foundation and suitable framework by using ISTQB test process for testing applications in cloud environments. The presented framework have the following features:

- Monitoring and control perform in several such as start testing, monitor testing progress, review and report results.
- In the proposed framework, we collect information in developing test scenarios and start testing steps to have adequate and complete understanding of requirements.
- Reports and log files using the strengths of the proposed method. These files is used to report progress of the application, also is used to find failed states.

This framework is an appropriate and effective architecture and model for evaluating cloud application that is used in many public and private organizations or institutions. The Proposed model is still an abstract model and should be drilled down which is subject of the next paper to be published.

References

[1] Haryadi S. Gunawi, Thanh Do, Pallavi Joshi, Peter Alvaro, Joseph M. Hellerstein, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau, Koushik Sen, and Dhruba Borthakur, FATE and DESTINI: A Framework for Cloud Recovery Testing, Proceeding of 8th USENIX conference on Networked system design and implementation, April 2011.

[2] K. Laskey, P. Brown, J. Estefan, F. McCabe, D. Thornton, "Reference Architecture Foundation for Service Oriented Architecture Version 1.0"

[3] OASIS(Organization for the Advancement of Structured Information Standards), July 2011.

[4] Prakash V., Ravikumar Ramadoss and Gopalakrishnan.S, Software as a Service(SaaS) Testing Challenges – An In-depth Analysis, in International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012.

[5] Jerry Gao, Xiaoying Bai, and Wei-Tek Tsai, Cloud Testing- Issues, Challenges, Needs and Practice, September 2011.

[6] Yatendra Singh Pundhir, Cloud Computing Applications and their Testing Methodology, Bookman International Journal of Software Engineering, Vol. 2 No. 1, 2013.

[7] G. Gowri, M. Amutha, Cloud Computing Applications and their Testing Methodology, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 2, February 2014.

[8] Nezih Yigitbasi, Alexandru Iosup, and Dick Epema, C-Meter: A Framework for Performance Analysisof Computing Clouds, May 2009.

[9] T. Parveen, and S. Tilley, "When to Migrate Software 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.

[10] Philipp Zech, Michael Felderer, Ruth Breu, Towards a Model– Based Security Testing Approach of Cloud Computing Environments, IEEE Sixth International Conference on Software Security and Reliability Companion (SERE-C), 2012 .

[11] Naganathan Vijayanathan and Sankarayya Sreesankar, Overcoming Challenges associated with SaaS Testing". Infosys View Point. 2012.

[12] Naganathan Vijayanathan, Bellave Shubha, The Challenges Associated with SaaS Testing". Infosys View Point, September 2011.

[13] D. Graham, E. Veenendaal, I. Evans and R.Black, "Foundations of Software Testing ISTQB Certification", Patrick Bond, 2008.

[14] R. Black, "Guide to the ISTQB Advanced Certification as an Advanced Test Manager", Vol.1, Rocky Nook, USA, 2009.

[15] IEEE Standard 829-2008 Standard for Software and System Test Documentation.

[16] IEEE Standard 1012-2012 Standard for System and Software Verification and Validation.

[17] IEEE Standard 1044-2009 Standard Classification for Software Anomalies.

[18] en.wikipedia.org/wiki/Cloud_testing.