ISSN NO: 9726-001X
Volume 06 Issue4 Dec 2018

JOURNAL OF CURRENT SCIENCE

A New Test Strategy for Web-Based Applications

CH.CHARAN KUMAR
ASSISTANT PROF.OF VASAVI ENGINEERING COLLEE A.P

Abstract

Recent years have seen a surge in interest in "the cloud" as a novel approach to building and distributing software as a service through preexisting, low-cost networks.The introduction of cloud computing has altered every step of the software development process, including the testing phase. Testing in the cloud is an active topic of study within the software engineering discipline. Ironically, the shorter development time for SaaS also means more time spent testing. Based on the ISTQB standard framework and the requirements and cloud testing processes, a new framework for cloud testing is introduced in this article. Test cases are designed, a cloud provider is chosen, infrastructure is set up, cloud servers are leveraged, testing begins, progress is monitored, a report is generated, and the process is completed. Some exercises are included into each of these stages.
Cloud computing; cloud testing; cloud based application; software as a service (SaaS); testing framework

1. Introduction

The concept of "cloud computing" as a novel approach to building and distributing software and web services has attracted a lot of attention in recent years. The software testing process is only one of several that cloud computing impacts. In the same vein as the widespread use of cloud computing's standard jargon, including SaaS, PaaS, and IaaS.

The testing of modern apps is more difficult and costly since they are intended to operate on numerous platforms, including portable and virtual environments. The functional and non-functional parts of an application need distinct methods of validation. Since cloud testing is a relatively new field in IT, no established norms exist. Scalability, dependability, and performance are three of the biggest obstacles for SaaS.

Testing procedures used to guarantee the correct operation of applications developed using the SaaS methodology. When an iteration of the SaaS Development Process is complete, the next phase is testing. Competition in the software services industry drives both the SaaS development process and SaaS testing. If they want to succeed in today's highly competitive software industry, SaaS testing and the SaaS development process must use agile techniques [3, 6].

SaaS has exploded in popularity over the last several years, with many businesses switching over to it in order to take advantage of its many advantages, such as pay-as-you-go pricing and instantaneous access. Testing for the security of SaaS apps is more involved than for on-premise systems. Business logic, security, data integration, performance, and scalability testing are just a few of the aspects that need to be examined. The next section [4] discusses some of the difficulties encountered while testing SaaS apps.

1. Evaluation of Safety and Confidentiality

2. Regular and Minimal Advance Warnings

3. Evaluation of Capabilities

4, Transnational Movements

5. Expertise in Business

6. Accreditation

ISSN NO: 9726-001X
Volume 06 Issue4 Dec 2018

The following is the outline for this paper. Some relevant literature on cloud testing is discussed in Section 2. The cloud application testing strategy is then outlined in section 3. In the next part, the fourth, we provide a new framework. The framework's efficacy is assessed in the fifth part. The paper's conclusion is presented.

2. Related Work

1. Testing in the cloud is an active topic of study within the software engineering discipline. Only in recent years has there been any serious study of cloud testing. While cloud testing has shown promise, it is still too early to declare any major successes or announce any groundbreaking research and advancements.
2. Two additional test cases were proposed for cloud recovery[1]: FATE (Failure Testing Service) and DESTINI (Declarative Testing Specifications). FATE is used to methodically put cloud systems through many failure scenarios. DESTINI was created so that retrieval processes may be specified in as much detail as possible.
3. This article presents a framework along with various demonstration services. The framework is modeled after the industrial sector. There are three components to this system:
4. Services in Management 1
5. Trial Clouds
6. Third-party service testing suite
7. Each of these processes uses its own computer. That's why it'll cost a lot to put into effect. The model's dependence on all other pieces comes at an additional expense. This indicates that other parts are dependent on the functioning of the faulty one. The Service Manager, for instance, is in charge of keeping tabs on everything. If a machine is tasked with processing a certain portion, and it fails to do so, that machine has failed in its duty. Inaccurate data from this portion would prevent the rest of the system from functioning, hence it's not acceptable.
8. C-Meter is a framework for measuring cloud system efficiency[7]. It has evolved to lessen the burden of resource allocation and release. A scheduler's primary goal is, of course, cost cutting. The framework's architecture is shown in the following image. The following are some components of this strategy:
9. 1. The hub manages and keeps tabs on all inputs. This section now houses the scheduler.An inter-cloud interaction controller.
10. Toolsets comprised of configuration and growth modules for processes.
11. A model for assessing cloud-based software using graphs has been published in [8], and with it comes a technique for testing such applications. Here are some of the criteria used to evaluate this approach:
12. First, make sure everything is being used effectively.
13. 2. keeping an eye on cloud computing's scalability.
14. In the publication [9], an automated methodology for testing the safety of cloud-based apps for Android is described. The following are some components of this structure:
15. 1) Recognize the source of information.
16. Second, this technique uses an output to alert the user if something out of the ordinary occurs.
17. Third, after receiving the input data, scour the collections of repositories for relevant test cases.
18. The purpose of the monitoring system is to keep tabs on how far the program has come.
19. If you find an issue in a repository that also stores other problems, you should address the problem head-on.
20. Finally, a report is written up to detail the system's creation and demonstration.

2. 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 ameans of load testing and stress testing web sites (as well-known examples in addition to others). Using cloudtesting gives users unlimited resources, paying only for what users consume [17].

To provide a common set of standardized documentsthe 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 testprocesses, activities, and tasks in support of all software life cycle

ISSN NO: 9726-001X
Volume 06 Issue4 Dec 2018

processes, including development, operation, and maintenance.

- Define the test tasks, required inputs and outputs.
- Identify the minimum test tasks related tointegrity 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 followingtest 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 theirrequirements.
4) Test Items:
   Testing cloud applications includes all of its features software, network and infrastructure[10,11].

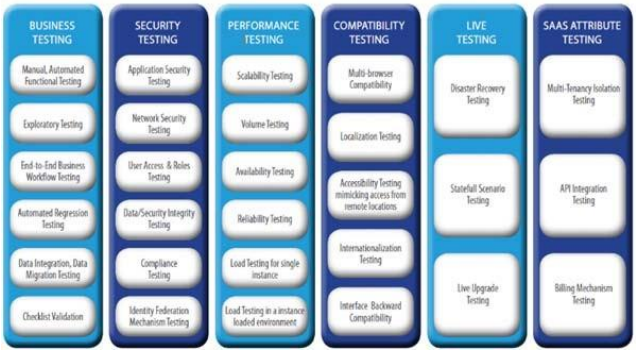

Fig. 1 Features to be tested in cloud application

7) Features not to be tested: Testing resource constraints force us to ignore some aspects of theTesting.
8) Approach: See figure 2 to get information about process of SaaS testing.
   - Software: Testing focuses on aspects like testing component functions,businessworkflows, browsercompatibility and data security, data integrity and data access privileges.
   - Network: Testing of the network needsto 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 conductedon infrastructure. As this is likely to have a great impact on the end user experience. Infrastructure testing includelive upgrade, disaster recovery tests. Theemphasis 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.

ISSN NO: 9726-001X
Volume 06 Issue4 Dec 2018

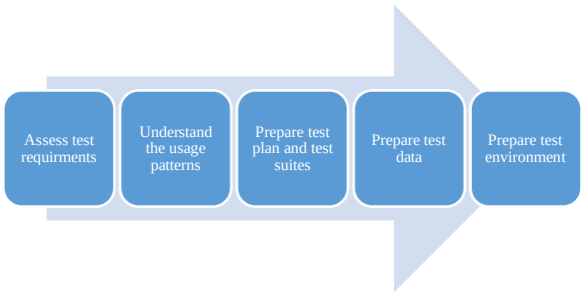

Fig. 2 SaaS Testing Process

9) Item Pass/Fail Criteria: Including the acceptancecriteria can be cited response time, transaction rate, and load and resource usage.
10) Suspension Criteria and Resumption Requirements: Know when to pause in a seriesof 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 plansdocument, 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 andhardware requirements.
13) Staffing and Training Needs: Educating system under test and testing tools. It also determines what should be tested and who is responsible forit.
14) Responsibilities: Determine who is responsiblefor what.
15) Schedule
16) Planning Risks and Contingencies
17) Approvals
18) Glossary

3. Brief overview of software Testing using ISTQBFramework

ISTQB is a global, all-encompassing framework for testing software, as was previously stated. Despite its depth, it is not tied to any particular software architecture and instead explores the test in a more broad sense [12]. The fact that ISTQB can be used as a standard against which all software models can be evaluated is a major plus, but the standard's overarching nature also means it can't go into the specifics of how different architectures work. ISTQB examines testing in the following areas:

The Test Plan in the Context of the System Life Cycle
• Method of testing
Management of testing
Potential Dangers of the Test
Methods of Examination

The life cycle is examined since it has an impact on the testing process but is otherwise left out of our discussion.
.

2.1. Test in software life cycle

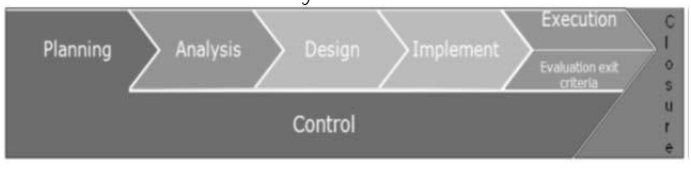

Fig. 3Test process in ISTQB

ISSN NO: 9726-001X
Volume 06 Issue4 Dec 2018

The costs associated with fixing the faults in the system development would be prohibitive if software testing were treated as a distinct activity. After introducing system development stages based on ISO-IEC 12027, ISTQB recommends a V-model for testing across the life cycle of a system.The testing procedure is broken down into the following steps according to ISTQB (see Fig. 3):

• Managing and organizing tests

Analysis and planning of tests

• The Execution of Tests

• Analysis of Results

Result of the Tests

All of the aforementioned steps (with the exception of control) are typically carried out in a sequential order, however depending on the project, some of them may be performed in parallel or be repeated. Brief descriptions of everyone involved follow.

The test strategy is chosen and its implementation in testing is specified at this phase of test planning and control. Here under test control, we outline the steps we took to implement the strategy. As can be seen, test control is an ongoing process that starts at the very beginning of the testing phase and lasts until the very conclusion.

In this phase, we analyze and design the test, and we create the test cases. The test objectives become specific plans, and the test infrastructure and tools are selected at this step.

The test is implemented when the test cases are used and run.

The purpose of this step is to assess the level of success the test had in identifying bugs.

In this last phase of the testing process, before releasing the findings, we examine the data we collected from the tests.

The Cloud Testing Framework that we propose

In this article, we provide a novel methodology for testing SaaS programs. The ISTQB and the cloud testing procedures provide the foundation for this framework. The main building blocks of our software testing system are shown in Figure 4 below.

After then, the specific actions that will be carried out at every given stage are spelled out. The following should happen during the testing phase of scenario development:

ISSN NO: 9726-001X
Volume 06 Issue4 Dec 2018

1. The first step is to identify the trend test (Thatmeans 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.

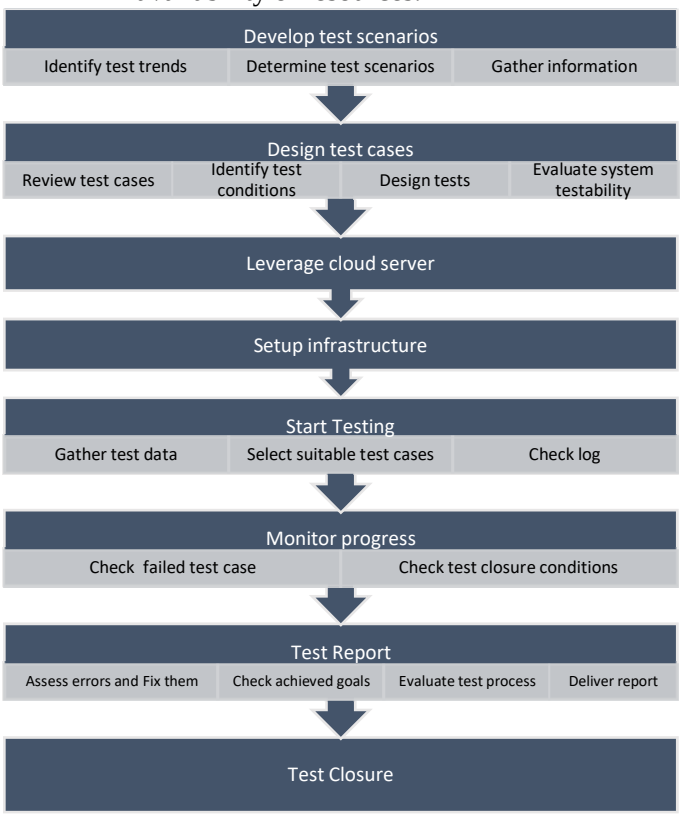

Fig. 4Overall architecture of proposed testing framework

There are a few things to keep in mind while designing test cases so that they test for everything. Therefore, the following procedures are mandatory:

1. Evaluate Test Scenarios

2.Establishing the Test Environment

3 Create Evaluations

Assess the capacity to test the system

The needs of the business are taken into account when deciding which cloud server to use and setting up the cloud-based architecture.

needs, stakeholders and test team. The start test stepconsider the fact that have collected enough data to test the feasibility of whether or not. A set of test casesconsidered, 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 followingactivities:

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 goalsconsidered asachievedgoals.

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 beendiscovered.

4. Evaluation the proposed framework

The advantages of presented framework towards theframework 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 thetestability 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 forthose who are supposed to benefit from system.

The proposed method is compared to the C-Meterframework 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. Thisleads 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 preparereports and log files. This framework is basedon 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 eachother while the

design of a cloud application can be very effective on testing it. In ourmodel, 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 inthis 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 classifyinputs 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 checkfailed stateto correct it.

5. Conclusion

Using the ISTQB test procedure, we have laid the groundwork and created a viable framework for testing apps on the cloud. The characteristics of the provided framework are as follows:

Testing, evaluation, and reporting are just a few of the many tasks that fall within the purview of monitoring and control.

throughout order to have a thorough and full grasp of requirements, we gather data throughout the proposed framework's test scenario development and testing phases.

Log files and reports that capitalize on the benefits of the suggested approach. These documents are used for tracking down error conditions and reporting application progress.

Numerous public and commercial institutions rely on cloud applications, and this framework provides an acceptable and effective architecture and methodology for doing so. The proposed model is still at a high level of abstraction and requires more investigation, which will be the focus of the next study.

.

References

[1] FATE and DESTINI: A Framework for Cloud Recovery Testing, by 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; published in the Proceedings of the 8th USENIX conference on Networked system design and implementation; April 2011.

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

[3] Released in Julay 2011 by OASIS (the Organization for the Advancement of Structured Information Standards).

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

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

[6] Source: [5] Yatendra Singh Pundhir, "Cloud Computing Applications and its Testing Methodology," Bookman International Journal of Software Engineering, Volume 2, Issue 1, January 2013.

[7] Based on the article "Cloud Computing Applications and their Testing Methodology" by G. Gowri and M. Amutha in the February 2014 issue of the International Journal of Innovative Research in Computer and Communication Engineering, Volume 2, Number 2.

[8] [7] C-Meter: A Framework for Performance Analysis of Computing Clouds, Nezih Yigitbasi, Alexandru Iosup, and Dick Epema, May 2009.

[9] When to Move Software Testing to the Cloud?, by T. Parveen and S. Tilley, [8]. Software Testing in the Cloud: Proceedings of the Second International Workshop, STiC 2018, Third IEEE International Conference on Software