上海海洋大学

**文献翻译**

**( 2016届本科 )**

**学 院： 信息学院**

**专 业： 软件工程**

**班 级： 软工一班**

**姓 名： 施旭峰**

**学 号： 1659121**

**指导老师： 邹一波**

**2020年3月**

目录

**[一．介绍](#_Toc25989_WPSOffice_Level1)** **[4](#_Toc25989_WPSOffice_Level1)**

[1.1定义](#_Toc22209_WPSOffice_Level2) [4](#_Toc22209_WPSOffice_Level2)

[1.2软件即服务](#_Toc12619_WPSOffice_Level2) [4](#_Toc12619_WPSOffice_Level2)

[1.3 Salesforce.com](#_Toc4717_WPSOffice_Level2) [4](#_Toc4717_WPSOffice_Level2)

**[二．软件测试(SAAS)](#_Toc22209_WPSOffice_Level1)** **[4](#_Toc22209_WPSOffice_Level1)**

[2.1技术的选择](#_Toc10040_WPSOffice_Level2) [4](#_Toc10040_WPSOffice_Level2)

[2.2概念验证](#_Toc24820_WPSOffice_Level2) [4](#_Toc24820_WPSOffice_Level2)

**[三．结果](#_Toc12619_WPSOffice_Level1)** **[5](#_Toc12619_WPSOffice_Level1)**

**[四．结论](#_Toc4717_WPSOffice_Level1)** **[5](#_Toc4717_WPSOffice_Level1)**

**[五．承认](#_Toc10040_WPSOffice_Level1)** **[5](#_Toc10040_WPSOffice_Level1)**

**[六．参考文献](#_Toc24820_WPSOffice_Level1)** **[5](#_Toc24820_WPSOffice_Level1)**

**[I. I NTRODUCTION](#_Toc10580_WPSOffice_Level1)** **[6](#_Toc10580_WPSOffice_Level1)**

[A. Definition](#_Toc10580_WPSOffice_Level2) [6](#_Toc10580_WPSOffice_Level2)

[B. Software as a Service](#_Toc5544_WPSOffice_Level2) [6](#_Toc5544_WPSOffice_Level2)

[C. Salesforce.com](#_Toc19903_WPSOffice_Level2) [6](#_Toc19903_WPSOffice_Level2)

**[II. SOFTWARE TESTING IN SAAS](#_Toc5544_WPSOffice_Level1)** **[6](#_Toc5544_WPSOffice_Level1)**

[A. Technique choice](#_Toc1525_WPSOffice_Level2) [6](#_Toc1525_WPSOffice_Level2)

[B. Proof of concept](#_Toc29910_WPSOffice_Level2) [7](#_Toc29910_WPSOffice_Level2)

**[III. RESULTS](#_Toc19903_WPSOffice_Level1)** **[7](#_Toc19903_WPSOffice_Level1)**

**[IV. CONCLUSION](#_Toc1525_WPSOffice_Level1)** **[7](#_Toc1525_WPSOffice_Level1)**

**[V. ACKNOWLEDGMENT](#_Toc29910_WPSOffice_Level1)** **[8](#_Toc29910_WPSOffice_Level1)**

**[VI. R EFERENCES](#_Toc30424_WPSOffice_Level1)** **[8](#_Toc30424_WPSOffice_Level1)**

A Case Study Using Testing Technique for Software as a Service (SaaS)

摘要:2013年第二学期，在巴西环球公司客户关系管理(CRM)系统的开发过程中，有必要进行软件测试。但是，它是使用诸如基于脚本的测试、集成测试和用户验收测试等技术来完成的。这些技术不适用于云环境。因此，选择成对技术进行软件云测试。将传统方法的测试结果与两两测试结果进行比较，发现测试用例的数量减少了，测试时间缩短了六分之一。因此，我们得出结论，在云环境下，两两测试作为一种初始测试技术是有效的。

关键词:云计算- SaaS -软件测试-两两测试-Salesforce

一．介绍

尽管云与传统软件开发有相似之处，但由于灵活性、敏捷性和开发速度[1]，它需要不同的测试模式。在此基础上，对软件即服务(SaaS)测试技术进行了分析和探讨，提出了一种简单而有效的软件即服务测试方法。

在本例中，一个用于在case对象上注册产品故障的应用程序是在Salesforce.com平台上托管和开发的。由于该对象是标准对象，所以开发基于配置而不是编码。因此，交付的时间更短，速度更快

1.1定义

根据Kavis, M. J.[1]，云计算可以定义为计算机进化的结果集，包括硬件和软件两个类别。云计算的优点是速度快、开发成本低、业务投资回报高。通常，它的构建和实现速度更快，成本更低，并且为用户提供了有效的功能。

现在一个有趣的类别是SaaS。这些云系统更加引人注目，并正在成为业务管理的领导者。

1.2软件即服务

根据Kavis, M. J.[1]的观点，云服务有三种类型，分别是基础设施即服务(IaaS)、平台即服务(PaaS)和软件即服务(SaaS)。每一个都有特定的性质和工作程序。然而，尽管满足了不同的期望，一个模型补充了另一个模型，它们组成了抽象层。在SaaS环境中开发和安装的应用程序是需要不断变化的，而不是公司的核心业务。

由于需要web访问、可伸缩性和快速的系统配置来满足不断变化的业务，大多数客户和租赁者希望通过按需软件(SaaS)来解决他们的需求。

1.3 Salesforce.com

Salesforce.com是一个基于云的CRM，它被商业化为SaaS。作为一家云公司，它提供三种类型的许可。此SaaS模块提供了配置开发，因此具有敏捷性、快速实现和灵活性。

二．软件测试(SAAS)

2.1技术的选择

由于SaaS特征不能应用传统的软件测试,测试必须快,减少了测试用例的数量和需要的能力分析结果的数据输入没有知识是什么做的代码。考虑到这整个场景，我们确定最好的测试类型是黑盒测试，最好的测试方法是成对的[3]。

黑盒测试基于需求和规范。它不需要结构、内部路径或被测软件的实现方面的知识。黑盒测试有9种技术。然而，首先确定的用于SaaS[3][4]的最佳技术是成对的策略。

成对测试，和其他的黑盒测试一样，可以减少测试用例的数量，并且它适用于单元、集成、系统和验收测试级别。然而，这种测试并不能保证对应用程序有益，只有单向的了解、尝试。这就是为什么要用[3]来进行概念证明(PoC)。

2.2概念验证

PoC的目的是验证成对测试是否是在SaaS上执行测试的最快方法，以及这种技术的使用是否足够。通过分析和选择技术，可以选择哪个更适合敏捷平台，如项目三所示。稍后，PoC比较了使用的脚本测试和选择的技术。

主要的目标是收集度量，例如测试用例的数量，检测到的缺陷的数量，以及执行测试所花费的时间。基于这些结果，所提出的技术是可接受的还是不可接受的。

三．结果

为了实现PoC目标，进行了传统的测试和成对测试。第一种测试覆盖了159个测试用例。对于两个不同的团队，测试用例的执行时间是6个小时。在测试执行期间，发现的缺陷数量为3个。应用全对算法[5]，由17个测试用例组成第二个测试用例。缺陷的数量与传统技术相当，但所花费的时间是六分之一。成对测试用例的数据结构如下:

•3输入;2记录类型;3 sla;5阶段;3对象;和1的输出。

表一展示了结果的汇编以及两种技术之间的比较

|  |  |  |  |
| --- | --- | --- | --- |
| 技术 | 测试用例 | 时间 | 发现的缺陷 |
| 传统测试 | 159 | 6小时 | 3 |
| 成对测试 | 17 | 1小时 | 3 |

表一:传统测试和成对配对测试的比较。

四．结论

由于SaaS系统需要另一种不同于传统环境的测试技术，本文在Salesforce.com平台上进行了一些测试，并对两两测试的适用性进行了分析。主要目的是证明成对测试是一种推荐的技术，首先用于基于云计算的软件。将此技术应用于Salesforce实例，可以比较传统技术和成对测试。

分析证据和措施，可以从这项研究中得出结论，这些技术的使用有很大的差异，可以在第三节中看到。成对测试减少了执行测试的时间，同时也减少了测试用例的数量，发现了与传统测试相同数量的错误。因此，黑盒测试的应用，例如两两配对，可能会增加开发阶段的速度，并减少SaaS测试所需的时间，这使得这种技术的使用非常值得推荐。由于云系统测试的复杂性，作者建议在这种类型的环境中执行其他黑盒测试，如果覆盖特定类型的测试场景，可能会获得更多的时间增益。这可能是SaaS的一个创新。本研究为初步研究，可作为今后研究的参考。

五．承认

作者感谢所有相关人员、机构和公司为本文的发展提供了时间、数据和知识上的支持。所以，我们应该感谢ITA。

六．参考文献

1. Kavis, M.J. “Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)”. New Jersey, Wiley, 2014.
2. Gartner “Magic Quadrant for Sales Force Automation” http://www.gartner.com/technology/reprints.do?id=1-1XENSUE&ct=140715&st=sb, October 15th 2014.
3. Copeland, L. “A Practitioner’s Guide to Software Test Design”. Boston, Artech House Publishers, 2007.
4. Crispin, L. Gregory, J. “Agile Testing – A Practical Guide for Testers and Agile Teams”. Boston, Pearson Education Inc, 2009.
5. Bach, J. “ALLPAIRS Test Case Generation Tool”, http://allpairstesting.com/allpairsgenerator, October 4th 2014.



A Case Study Using Testing Technique for Software as a Service (SaaS)

Abstract — In the second semester of 2013, during the development of the Customer Relationship Management (CRM) System at Brazilian global company, there was the necessity to execute software testing. However, it was done using techniques,such as Script-based Testing, Integration Testing and User Acceptance Testing. These techniques were inadequate to the cloud environment. Thus, the pairwise technique was chosen for the software cloud testing. Comparing the results from the traditional techniques to the pairwise testing it was noticed that the number of test cases was reduced and testing time was cut by one-sixth. So, it was concluded that pairwise testing is effective as an initial testing technique in the cloud, in conditions in which it is applicable.

Keywords: Cloud Computing – SaaS – Software Testing –Pairwise Testing - Salesforce

I. I NTRODUCTION

Even though, the similarity between cloud and a traditional software development, it needs a different testing patterns due to the flexibility, agility, and development velocity [1]. Therewith, this paper was developed to analyze and identify an adequate technique to test software as a service (SaaS), proposing a simple, but an efficient test method. In this case, an application, to register product failures occurrences on case object was hosted and developed at Salesforce.com platform. As this object is a standard, the development is based on configuration instead of coding. Due to this, the deliverables are shorter and faster.

1. Definition

According Kavis, M. J. [1], cloud computing can bedefined as a result set of computer evolution, in both categories hardware and software. The advantages of cloud computing are the velocity, low development cost and return of investments for the business. Usually, it is built and

implemented faster, at a low cost and with effective functions to the users [1]. An interesting category nowadays is the SaaS. These cloud systems are even more in the spotlight and are becoming leaders in business management.

B. Software as a Service

According Kavis, M. J. [1], there are three types of cloud services, being they the Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and the Software as a Service (SaaS). Each one has specific properties and working procedure. However, despite meet different expectations, a model complements the other, they compose the level of abstraction. The applications developed and installed in a SaaS environment are the ones that need to be continuous changing and are not the core business of the corporation.

Due to the necessity of web access, scalability, and fast system configuration to meet the continuous business changing, most of the customers and hirers expects to solve their need through a software by demand (SaaS).

C. Salesforce.com

Salesforce.com is a cloud based CRM that is commercialized as a SaaS. As a cloud company, it offers the three types of licenses. This SaaS module provides a configuration development and in consequence agility, fast implementation, and flexibility [2].

II. SOFTWARE TESTING IN SAAS

A. Technique choice

Due to the SaaS characteristics the traditional software testing cannot be applied, the test has to be fast, reducing the number of test cases and need to have the capacity to analyze the results against the data input without the knowledge of what are being done by the code.

With this entire scenario in mind, it was identified that the best type of test was the black box testing and the best method to do it was pairwise [3].

Black box testing is based on the requirements and specifications. It requires no knowledge of structure, the internal paths or the implementation of the software under test. There are nine techniques on black box testingstrategy, however the pairwise was the best technique identified to be used at first for SaaS [3] [4].

Pairwise testing, as others black box testing, can reduce the number of test cases and it is applicable at the unit, integration, system, and acceptance test levels. However, this kind of testing do not guarantee that is benefic for the application, there is only one-way to know, trying. That is why a proof of concept (PoC) was done [3].

B. Proof of concept

The purpose of the PoC was to verify if the pairwise testing is the fastest way to execute tests on SaaS and if the use of this technique at first is adequate. It was possible, by analyzing and selecting the technique, to select which is better to fit an agile platform, as done in the item III. Later, the PoC compared the scripting test used and the technique chosen.

The main objective was to collect the measures such as the number of test cases, number of detected defects, and the time spent to execute the test. Based on the results, the proposed technique could be acceptable or not.

III. RESULTS

Traditional test and pairwise testing had been done toaccomplish the PoC objective. The first kind of test

covered one hundred fifty nine test cases. The executiontime of the test cases was six hours, for two different

teams. During the test execution, the numbers of defects found were three.Applying the All-pairs algorithm [5], seventeen test cases composed the second one. The number of defects has been equal to the traditional techniques, but the time spent was one-sixth. The data for pairwise test cases were structured in:

• 3 Inputs; 2 Record Types; 3 SLAs; 5 Phases;3 Objects; and 1 Output.

Table I presents the compilation of the results and a comparison between the two techniques

|  |  |  |  |
| --- | --- | --- | --- |
| **Technique** | **Test Cases** | **Time Spent** | **Defects** |
|  |  |  | **Found** |
| Traditional | 159 | 6 hours | 3 |
|  |  |  |  |
| Pairwise | 17 | 1 hour | 3 |
|  |  |  |  |

Table I – Comparison between Traditional and Pairwise Testing

IV. CONCLUSION

Due to the necessity of another testing technique for SaaS systems, that is not the same as the traditional environments, this paper provided some tests and an analysis of the applicability of pairwise test on the Salesforce.com platform.

The main objective was to proof that pairwise testing is a recommended technique to be used at first for software based on cloud computing. Applying this technique at Salesforce instance, it was possible to compare the traditional technique and the pairwise testing.

Analyzing the evidences and measures, it can be concluded from this research that there is a great difference between the use of these techniques, as can be seem at section III. The pairwise testing has reduced the time spent for the execution of test as much as it has reduced the number of test cases, finding the same quantity of faults as the traditional testing. So, the application of a black box testing, such as pairwise may increase the velocity of the development phase and reduce the necessary time for SaaS testing, what makes the use of this technique pretty recommendable.

Due to the complexity of cloud system testing, the authors recommend the execution of other black box testing in this type of environments, it may result in more gain of time if specific types of testing scenarios are covered. That may be an innovation regarding SaaS. This research is an initial step and can be used as a reference for future researches.

1. ACKNOWLEDGMENT

The authors want to thankful all the involved, such as people, institutes and companies that support them with time, data and knowledge for the development of this paper. So, many thanks should be made to ITA.

VI. R EFERENCES

[1] Kavis, M.J. “Architecting the Cloud: Design Decisions for CloudComputing Service Models (SaaS, PaaS, and IaaS)”. New Jersey,Wiley, 2014.

[2] Gartner “Magic Quadrant for Sales Force Automation”

http://www.gartner.com/technology/reprints.do?id=1-1XENSUE&ct=140715&st=sb, October 15th 2014.

[3] Copeland, L. “A Practitioner’s Guide to Software Test Design”.Boston, Artech House Publishers, 2007.

[4] Crispin, L. Gregory, J. “Agile Testing – A Practical Guide for Testersand Agile Teams”. Boston, Pearson Education Inc, 2009.

[5] Bach, J. “ALLPAIRS Test Case Generation Tool”,http://allpairstesting.com/allpairsgenerator, October 4th 2014