A SaaS Testing Approach based on Crowdsourcing

Vikram Singh Chauhan (2013H313080H) & Bofin Babu (2013H313085H)

Abstract— **In recent years the Software as a Service (SaaS) model of software flourished as organizations of different size and types are extremely interested in readily available buisness applicatons. It result in remarkable increase of cloud based SaaS testing. SaaS testing came with a new class of challenges including short validation cycle time, data integration, updates, accessibilty and compatibility validation. Traditional SaaS testing methodologies in restricted environment is not sufficient to overcome SaaS challenges. This paper proposes an efficient and cost effective SaaS testing approach using crowdsourcing followed by some techniques of effective management of crowd, based on a case study.**

I. INTRODUCTION

In 1999, a team of software engineering from the Durban University, Keele University and University of Manchester Institute of Science and Technology called the Pennine Group demonstrated that the development of new architectural styles will not advance the software field rather we need a totally different view that how software services provided to the users. Cloud based Software as a Service (SaaS) model came as revolutionary model that provides small and business level applications as per the requirements whenever needed. SaaS provides software application vendors an internet based delivery model to serve a big group of tenants. SaaS focuses differentiating the possession and ownership of software from its use[1]. In I.A we discuss in general about SaaS model, followd by an overview of existing testing methodologies in I.B. In secion I.C we introduce in brief the idea of crowdsourcing.

A. *SaaS*

Delivering the functionality of the software as a set of distributed services can overcome many outcomes of current software architectures like availability, economic considerations. The advantages of the SaaS model are a) The ease of its deployment as in most of the scenarios we need a browser and an internet connection as in case of Microsoft Office 365 and Gmail. b) SaaS is affordable as compare to traditional software architectures as the SaaS vendors are responsible to handle the complexity of underlying IT infrastructure and client needs zero infrastructure for the running and maintenance of servers and databases. c) Scalability is another gain since SaaS services resides in the cloud environment that are scalable and may have integration with the other SaaS offerings so the users do not need to buy new equipment they just have to enable new SaaS offering. d) Pay as you Use, Work Anywhere, Long Term Customer Relationship are some other add-ons to SaaS software model.

B. *Testing in SaaS*

SaaS testing environment varies from traditional software testing methods as the SaaS applications are used by the group of clients in diverse environments so it requires various add-ons to usual software testing. Further all the SaaS based applications get tested in cloud based environment. SaaS testing includes the aspect of validating SaaS applications with respect to multi-tenancy, integrity, business workflows, availability, reliability, deployability, ease of deployment, scalability, accuracy, ease of use, testability, portability live updating[2]. According to [3][4] SaaS testing is the composition of three components:

- *Application:* Application testing considered to be the most significant component of SaaS testing since it is used as final end-to-end product in various critical environments. It consists of testing the browser compatibility, end-to-end workflows for functional validation. SQL injection based testing, testing cookies for security testing. This is followed by the performance testing which is done through simulated large user and high load simulated environment.

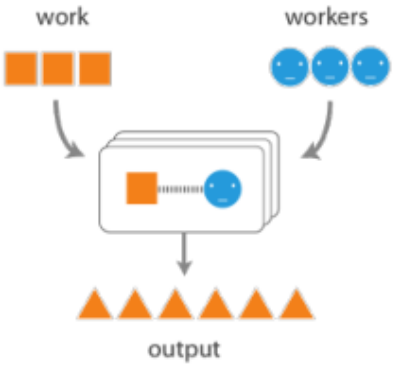

Fig. 1. The work is divides into small unites where workers(crowd) contribute a small portion to the final result.

- *Network:* SaaS applications predominantly based on cloud that are used by customers of varying network environments. So, network based testing plays a crucial role to ensure availability and transfer of data in varying bandwidth environments and other security issues like flow control, encryption techniques.
- *Infrastructure:* Testers conduct significant number of tests on infrastructure with production like configurations to enhance end user experience. Primarily live update tests, recovery tests and backups tests are performed to ensure reliability.

C. *Crowdsourcing*

Crowdsourcing is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers [5]. Well known examples of crowdsourcing include, Wikipedia, Amazon Mechanical Turk, TopCoder, KickStarter, Kaggle, Threadless etc. A broad classification can be made to these services as volunteer-based crowdsourcing and non-volunteer-based crowdsourcing. Wikipedia is a notable example of volunteer-based crowdsourcing, where it is almost entirely maintained by millions of volunteers of which around 93%[6] for non-professional reasons. A typical example of non-volunteer-based crowdsourcing is Amazon Mechanical Turk where registered users complete micro tasks for small incentives.

Even though not all jobs can be completely solved by crowdsourcing, there are portions of any job which can be accomplished by the crowd[7]. Use of crowdsourcing as technique to solve complex, large scale problems will continue to expand for the decades to come. With this growth there is also a risk that crowd work will fall into an intellectual framing focused on low-cost results and exploitative labour[8] It worth mention that several approaches like [9][10] and [11] have been introduced to deal with this issue. We will discuss more about them in section II.

II. Related Works

Crowdsourcing is an emerging field for software development and testing. Coming to the related work in this field most of the research papers are centric to crowd based software development as crowdsourcing can be implanted in existing software development models effectively. In [12] they are exploring the crowdsourcing goals like Quality software, Rapid acquisition, Talent identification, Cost reduction and cloud based software crowdsourcing architecture and proposed the ways to manage crowd of testers and developers based upon Game theory(reputation based), Economic model(reward based) and Collaborative model. In [4] authors from Infosys group splitting the SaaS testing into three components as mentioned in the section I.B. They further enumerated the key challenges associated with SaaS testing specifically pointing to Testing SaaS upgrades, Validating interface compatibility, Data Security and Privacy. As a solution to [4] in [13]the authors proposed some ideas to overcome SaaS testing challenges but later we will show how our proposed approach can be more efficient and cost effective. Coming to the practical considerations Microsoft used crowdsourcing for windows 8 and spent $100,000 for crowd based security testing. They crowdsourced the beta version for office 2010 productivity suite and as an outcome, the product got download and tested by the crowd of 9 million resulting in 2 million valuable comments that enhanced applications productivity. Unlike Conventional Web apps the SaaS testing is more challenging since the development life cycles are shorter and need different test approaches. We find that not so much research have been done taking the SaaS

testing as a Centric approach. In this paper we are quantifying the challenges in SaaS testing and try to resolve them using crowd based software testing.

III. SaaS testing approach based on crowdsourcing

We would like to point some challenges in the existing SaaS tsting models. We introduce our idea of crowdsourced SaaS testing followed by a brief discussion on how this approach can address the existing challenges.

A. *Challenges in traditional SaaS Testing*

Due to its exclusive model, testing of a SaaS implementation from the service provider's point of view possesse unique challenges, include:

- *Unbiased perfomance and accessibilty testing*: In performance and accesssibility Testing, simulating a massive number of users with different characteristics is a big challenge. The existing approacch is to use tools like HP Loadrunner and SOASTA's CloudTest to simulate thousands of concurrent users to put the application through the worst case of real-life user loads. But this simulation might not alaways represent the real life scenario. We will need to address this gap between software based perfomance testing and real user imapct.
- *Data Security and Privacy*: A major charesteristic of SaaS applications is multi-tenacy. Being built to serve multple customers, SaaS applications should make sure that one customer's data cannot be accessed by another customer and vice-versa.
- *Compatibility validation*: In SaaS upgrades, especially interface upgrades happen frequently. After every update the inteface needs to be validated by the customers for usability. The customers should also be able to work with the older interface, in the case where they have internal integration of their application with the SaaS solution.
- *Testing upgrades and QA*: When SaaS providers give upgrades, they provide the existing customers a few days for QA notifications. Within this timeframe, the existing customers need to be able to read release notes, assess the impact of change and validate the impact of the change, to ensure that the service remains steady. In the case of live upgrade, which happens most of the times, without proper measures, testing on the application can hamper the experience of existing users.

B. *Crowd based testing approach*

This section gives a broad outline about how crowdsoucing can be implimented in SaaS testing and how much this approach would be effective in addressing the challeges faced in the existing methods.

In today's era of mobile and internet users have access to customer centric SaaS applications at varying geographic locations with different levels of computer knowledge. In cloud based SaaS environment these applications are tested by dedicated team of testers which may fail to analyse the application based on customers point of view. Also the technology is scaling at a fast pace and SaaS appliactions runs seamlessly on differet devices, operating systems and diveresed network bandwidths. Establishing such a vast simulation environment in the organization is not a cost-effective solution. So, increased momentum of crowd source testing can cover a big volume of testing aspects in varying geographical locations as compare to internal testing team without increasing the expenses significantly.

Security testing is an essential part of every SaaS application. Security testing needs experts and it is often considered one of the most expensive and dominant part of the entire SaaS testing. Crowdsourcing can also effectiviely implimented for SaaS security testing. In fact many MNCs including Facebook, Google, Microsoft etc have bug-bounty programs, where anyone is eligible for a small incentive or a reward if he finds a noticable bug in the software. There are even platforms like BugCrowd, started recently which provides crowdsourced security for web and mobile applications by running managed bug bounty programs.[15] Companies can register and announce their reward programs while users, ranging from amatures to well known security researchers test the SaaS product for bugs, which would eventually lead

them to a reward. Testers can also build up a profile in BugCrowd showcasing their achievements.

Both technology and customer demands are growing at fast pace. To handle the growing demands Updates become an essential part of any software. Update plays most significant role in bug fixing, product enhancement and finding security vulnerabilities. Testing the updates with the team of internal testers is not sufficient. Many bugs and issues may bypass all the testing phases. The biggest challenge in SaaS updation is that the live upgrades should not impact the existing connected SaaS users [4]. Recently Apple's mobile operating system iOS 8 upgrade reported many issues after its final release and affected millions of user worldwide. Problems like this can be effectively solved by implementing crowd based testing to products after internal testing. Latest releases and upgrades can be forwarded to the crowd of users who voluntirely wants to be the part of testing in collaborative manner.

C. *Managing Crowd*

In section II.B we have seen that how crowdsourcing solved SaaS testing challenges. Managing crowd in a crowdsourced environment is still a hot topic of research in this area. This section deals with some important researches happened in the last few years to address this issue.

In [14] Bretschneider et al talks about three challenges in a crowd environment: managing the process, managing the crowd and managing the technology. Managing the crowd is a big area which again includes issues like creating confidentiality among crowd and crowdsourcing company, Secrecy of new product innovations, ranking the expertise among crowd etc. Authors presented a case study of German based organization called TestCloud (website: www.testcloud.de) that provides crowd based software testing services. The very first thing they are concentrating is the confidentially agreements since in a crowdsourced environment there is always a risk of losing the information since the people engaged in testing may not be the employee of the firm. To resolve the mentioned issue the firm is enforcing a non-disclosure agreement (NDA) with the desired applicants to protect the secrecy of innovative products of different clients. Between crowd and crowdsourcer to retain confidentiality and protection of critical information NDA acts as a strict measure. Another challenge for testcloud is the varying quality of submission form crowdsourcees. To handle this testcloud adopted a new phase called induction phase. *testcloud Academy* where the desired applicants are taught on how to apply for the testing, how to find bugs and debug or record them. There are pre-tests according to which the new members have to conduct 1-2 tests in a span of 2 days. On the basis of tests results the skills of the new members are evaluated. *testcloud Academy* also assist the testers throughout their project and brush up their testing skills. In order to pay the testers testcloud uses the economic model based reward-structure[12] to motivate the crowd. They follow a transparent monetary reward system according to which the testers are paid on the basis of how critical the identified bug is or how helpful and appropriate their suggestion is[14]. There are some crowd testing organizations like TopCoder where participants are motivated towards gaining the reputation rather than receiving a reward price. In such scenarios we can rank them by evaluating their past testing practices and quality of suggestions.

IV. CONCLUSION

Crowdsourcing is the process of getting work done, usually online, from a crowd of people. Over the past few years, crowdsourcing brought out dramatic solutions to some hard problems both computers and human failed to solve in traditional approaches. Crowdsourcing can be effectively used to achive security, compatibiltiy and perfomance testing of SaaS applications. In buisiness testing and live testing, we still need to find ways in which we can crowdsource the process. Also implementing the crowdsourcing for system and integration testing is still a challege due to the resources needed and risks involved like lack of confidentiality regarding shared knowledge of software product. Furthermore the degree to which crowdsourcing can efficaciously implemented to the classes of SaaS testing, is also important to be studied in-depth. We leave them for future work.

REFERENCES

[1] M. Turner, D. Budgen and P. Brereton (2003) Turning Software into a Service, The IEEE Computer Society Computer, vol. 36, no. 10.

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

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

[4] Vijayanathan Naganathan, Shubha Bellave (2011) The Challenges Associated with SaaS Testing, Infosys.

[5] "Crowdsourcing - Definition and More". Merriam-Webster.com. August 31, 2012. Retrieved 2014-11-15.

[6] Wikipedia community. (2014, September 22). In Wikipedia, The Free Encyclopaedia. Retrieved 12 01, November 15, 2014, from http://en.wikipedia.org/w/index.php?Title=Wikipedia_community&oldid=626571064.

[7] Horton, J.J. and Chilton, L.B. The labour economics of paid crowdsourcing. Proceedings of the 11th ACM conference on Electronic commerce, (2010), 209218.

[8] Kittur, N., Nickerson, J., Bernstein, M., et al. The Future of Crowd Work. Conference on Computer Supported Cooperative Work and Social Computing, (2013).

[9] C. Raykar, S. Yu, L .H. Zhao, A. Jerebko, C. Florin, G. H. Valadez L. Bogoni, and L. Moy. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In ICML 2009: Proceedings of the 26th International Conference on Machine Learning, pages 889896, 2009.

[10] Welinder, Peter, and Pietro Perona. "Online crowdsourcing: rating annotators and obtaining cost-effective labels." Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. IEEE, 201

[11] Parameswaran, Aditya G., et al. "Crowdscreen: Algorithms for filtering data with humans." Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, 2012.

[12] Michael N. Huhns1, Wei Li, and Wei-Tek Tsai(2013) Cloud-based Software Crowdsourcing Dagstuhl Seminar 13362.

[13] Vijayanathan Naganathan and sreesankar sankarayya(2012) Overcoming Challenges associated with SaaS Testing, Infosys.

[14] Shkodran Zogaj Ulrich Bretschneider and Jan Marco Leimeister(2014) Managing crowdsourced software testing: a case study based insight on the challenges of a crowdsourcing intermediary,Springer-Verlag Berlin Heidelberg.

[15] BugCrowd. https://bugcrowd.com