Testing as a Service (TaaS): A Systematic Literature Map

Gustavo Girardon[1], **Victor Costa**[1], **Rodrigo Machado**[1],
Maicon Bernardino[1], **Fabio Paulo Basso**[1], **Elder de Macedo Rodrigues**[1]

[1] Universidade Federal do Pampa (UNIPAMPA)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{ggirardon, victorsc.rs, rodrigo.blizzard92}@gmail.com
bernardino@acm.org, {fabiobasso, elderrodrigues}@unipampa.edu.br

Abstract. *Background: The knowledge and application of tools to automate testing is essential to ensure software reliability and therefore its quality. Due to the increasing demand for quality in software projects executed in short time-scales, Testing as a Service (TaaS) appeared in the literature as contributions for cost reduction and productivity of automated tests. Aims: Once quality attributes from these contributions are not deeply discussed by the literature of the area, our goal is to investigate and identify these attributes from the TaaS platforms and providers commonly reported in the literature. Method: A protocol was formulated and executed according to the guidelines for performing systematic literature map in Software Engineering. Results: The TaaS providers and platform proposals found were classified according to the number of mentions in the literature, highlighting the most commonly mentioned and widespread. As well as the propagation and explanation of the main advantages and disadvantages reported in the literature on Testing as a Service. Conclusions: TaaS provides means for cost reduction and increase in productivity in comparison to traditional test approaches. This is a reality observed in 76 options for Test as a Service cloud platforms distributed over 52 papers. In addition, as their quality attributes, we also found eight groups of disadvantages and 21 of advantages. Thus, this systematic literature map is a valuable contribution for decision making on performance testing strategies.*

1. Introduction

Software testing is a key part when developing quality software products. When compared to other software development life cycle phases, which can represent up to 60% of the total development effort [Myers and Sandler 2004], software testing activities require many resources, such as time and money.

The purpose of testing includes quality assurance, verification, validation or reliability estimation. In order to balance team expectations, and due to time-to-market pressures, software projects usually present different points of view in terms of effort investment. For instance, it is common to observe conflict of interest among project managers, who deal with time schedules, test managers, who deal with quality assurance, and company managers focusing in budgeting. In this sense, empirical works provided many evidences suggesting that time invested in testing saves money in software projects [Myers and Sandler 2004, Perry 2007, Ammann and Offutt 2016].

Software testing is a fundamental activity of software quality assurance. As a mean to increase productivity, many tools have been proposed to assist developers and testers with their tasks. However, learning and writing test cases using those tools is still demanding much effort from software developers [Yu et al. 2009]. Thus, different means to reach software testing productivity together with quality expectations should be explored in details.

A possibility to gain in productivity is though services shared in clouds [Zhang et al. 2010]. For instance, cloud computing is defined by the National Institute of Standards and Technology (NIST) as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (*e.g.*, networks, servers, storage, applications, and services) [Mell and Grance 2011]. These resources can be rapidly provisioned and released with minimal management effort or service provider interaction [Garriga et al. 2016].

Cloud computing consists in distinct types of computing services delivered remotely to clients via the internet. To gain access to systems that deliver software as a service (SaaS), platforms as a service (PaaS) and infrastructure as a service (IaaS), clients typically pay a monthly or annual service fee to a provider. They subscribe to cloud computing services to reap a variety of benefits, depending on their particular business needs at a given point in time. The ability to access powerful IT resources on an incremental basis is leveling the playing field for small and medium sized organizations. Cloud providers offer the necessary tools and technology for small business to compete in the global marketplace, without the previously requisite investment in on premise IT resources. Clients who subscribe to cloud computing services are, therefore, able to greatly reduce the IT service expenditures for their organizations and gain access to more agile and flexible process in level of enterprise computing services.

Cloud computing not only brings new business opportunities but also causes a major impact on other fields. Likewise, one of the fields is Software testing. A new term, known as Testing as a Service (TaaS): is becoming a trending topic in different research communities, as well as cloud computing and IT businesses [Gao et al. 2013].

Software testing as a service leverages the vast resources of cloud environments to offer testing services to consumers on an on-demand basis. TaaS could potentially revolutionize software testing, promising reduced costs while offering more thorough testing. The main goal of TaaS is to reduce the budget of IT companies so that they can concentrate on their own business and can outsource the software testing to third parties [Harikrishna and Amuthan 2016]. Nevertheless, TaaS is a relatively new aspect of software testing and comes with many potential risks and challenges [Floss and Tilley 2013], which can be mapped through their quality attributes.

In order to reach these quality attributes, this paper presents a Systematic Literature Map (SLM) process applied to complete a systematic mapping study. Our contribution is threefold: 1) an analysis of 76 TaaS cloud platforms, including quality attributes discussed as advantages and disadvantages, 2) an analysis of their innovative features that makes this a well established research field, and 3) an analysis of the results reported by existing works in the test development practice.

2. METHOD

This mapping is built on the guidelines for performing systematic mappings in software engineering proposed by Petersen [Petersen et al. 2008].

2.1. Scope, Objectives and Research Questions

With the purpose of providing an empirical reference for professionals and researchers who search for new providers or platforms that have certain particularities in the execution of testing as a service, the objective of this study is to identify and characterize existing TaaS platforms in the literature. The following Research Questions (RQ) were defined.

RQ1: What are the main advantages described in the literature when using TaaS?
RQ2: What are the main disadvantages described in the literature when using TaaS?

2.2. Search Strategy

Formal literature research was conducted using only databases that: (i) have a search engine capable of using keywords; and (ii) contain computer science documents. The selection includes the following bases: Association for Computing Machinery (ACM) Digital Library[1], Engineering Village[2], IEEE Xplore[3], SCOPUS[4] and SpringerLink[5].

To define the search string the terms "Testing as a Service" and synonyms "TaaS" and "Testing in the Cloud" were used, as well as, the Boolean operator "OR" to select alternative synonyms.

2.3. Selection Process

1. **Pilot Search Strategy:** In order to verify the quality of the proposed search string, the approach called Search-Based String Generation (SBSG), proposed by [Souza et al. 2018] was applied. The approach is based on the calculation of precision and sensitivity indexes. The precision is the ability to identify the amount of *irrelevant* studies, while the sensibility is a measure to identify all of the *relevant* studies. When precision is zero, i.e., no irrelevant study is detected.
   This approach applies an Artificial Intelligence technique through the Hill Climbing algorithm proposed by [Russell and Norvig 2016], allowing the measurement of precision and sensitivity indexes for a set of keywords and an initial set of selected papers. In this context, the achieved results were 11.27% precision and 79.49% sensitivity.
2. **Initial Selection:** The strings were generated using the selected terms and synonyms and were run in the selected databases resulting in a initial aggregation of studies;
3. **Removal of Duplicates:** The results of the initial selection were filtered-out for duplicated entries;
4. **Intermediate Selection:** In this step, the researcher one, two and three read separately the title and the abstract (reading the introduction and conclusion when necessary) of each study. Here, the researchers decided to select or reject an article following defined inclusion and exclusion criteria;
5. **Data Extraction:** To extract relevant data from the selected publications, researcher one, two and three filled in a form produced to help answer the RQs.

---

[1] ACM: `https://www.dl.acm.org`
[2] Engineering Village: `https://www.engineeringvillage.com`
[3] IEEE: `https://www.ieeexplore.ieee.org`
[4] SCOPUS: `https://www.scopus.com`
[5] SpringerLink: `https://www.link.springer.com`

Our initial selection was conducted in April 2019 and provided 633 results. After filtering out duplicate entries, this reduced the number of results to 559. The number of duplicate entries was quite large and this can be attributed to papers being revised from conferences publications into journal articles, being extended and submitted in later conferences, and overlapping results from databases. After separately applying inclusion and exclusion criteria the number of results was reduced to 52 papers.

3. Results and Discussion

This section will provide detailed insights about the results of our systematic mapping as well as an interpretation of their significance.

3.1. RQ1. What are the main advantages described in the literature when using TaaS?

This section lists and explains the advantages reported by the final selection of studies.

Accuracy: Cloud testing can help organizations to simulate real-world production servers more effectively with the use of cloud resources. In addition, the load on servers can simulate a real world scenario more closely both from a scale of load perspective and from a geographical distribution perspective. The simulation of different browsers and platforms is also much easier when using the cloud infrastructure than when using the traditional one.

Cross-platform testing: Determine the behaviour of application and website in different environments. Cross-platform testing helps in identifying issues that may vary with platforms or configurations such as consistency, user interface, usability and performance issues.

Cost-Effective: The major advantage of implementing TaaS model is reduced costs. Since the entire infrastructure is hosted on cloud, it doesn't require investment of setting up servers, tools or operating systems which reduces the capital and depreciation costs. Moreover, since enterprises pay for the actual service and time taken, it helps control costs resulting in better Return-on-Investment (ROI). As mentioned, ROI takes a considerable leap when cloud testing is adopted appropriately. On-site QA team can focus on improving quality and testing features instead of spending time acquiring, setting up and configuring infrastructure. They can devote their time on process improvement activities. Substantial savings are also made on maintenance costs on finding and fixing bugs on released software.

Less required knowledge about testing: TaaS allows teams to accomplish automation with minimal technical knowledge and effort. Some platforms assist in the writing and execution of test codes, either by automation or human intervention.

Less time consuming: With the high availability of resources and allocation of these, the tests tend to run faster. This impacts the speed at which you will get the test results and the total cost of these.

Availability of tools and options: In contrast to performing on-premise testing, which cause project delays, testing on cloud provide limitless and round-the-clock access to servers, tools and programs from anywhere without waiting. This speeds up the entire testing cycle resulting in better time-to-market of applications.

On-demand: On-demand testing services promises to transform the way product development groups and IT houses buy testing services and manage their QA budgets utilizing a pay-as-you-test billing model. This model allows a shift from capital expenditure to operational expenditure, thereby drastically reducing your QA budgets, reducing IT maintenance overhead, and eliminating the need for IT and Testing staff to install and configure multiple test environments and tools.

Efficiency: TaaS model allows testers to focus on important processes while synchronizing tools, people and processes. Moreover, with standardized infrastructure, there are less errors of inaccurate configurations which ultimately attributes to drive efficiency in the entire test process.

Global Availability: Global Availability is the probability a system is functioning when needed to, under normal operating conditions. When the system is alive and well, the organization can continue to produce output and running tests. item

Scalability: TaaS environment allows testers to easily and quickly scale the applications under test and expand the capacity thresholds for thousands of users concurrently to suit agile and devOps demands. This model allows organizations to add or remove hardware or software dynamically, according to their needs. For example, a high-end server may not be needed until the load test or stress test or performance test starts. The server can be decommissioned as soon as the testing is complete. Similarly, in cloud testing, a license for a load-test application may be required only for the duration when load test is being run.

Maximize resource utilization: Effective use of resources. In general, cloud testing will require less administration effort compared to traditional testing. The organization can thus use resources more effectively. There is the ever-increasing need for industries to be environmentally responsible by going "green" and the IT industry is no exception. Cloud testing organizations enhance green testing. By sharing test resources in the cloud, businesses use IT resources solely on demand and this eliminate wastes by eradicating infrastructure idleness.

Compatibility: Platform compatibility ensures that software works consistently across all required platforms ensuring an optimal user experience. Compatibility can refer to interoperability between any two products: hardware and software, products of the same or different types, or different versions of the same product. Products that are designed to be compatible with future versions of themselves are referred to as forward compatible; products designed for compatibility with older versions are said to be backward compatible.

Geographical distribution: Geographically distributed clouds allow the simulation of multiple users from different locations.

Multi-tenancy: In cloud computing, the meaning of multi-tenancy architecture has broadened because of new service models that take advantage of virtualization and remote access. A testing-as-a-service (TaaS) provider, for example, can run one instance of its application on one instance of a database and provide web access to multiple customers. In such a scenario, each tenant's data is isolated and remains invisible to other tenants. This also applies to common SaaS platforms.

Flexibility: TaaS model imparts enormous flexibility by providing facilities to ramp-up and tear down the testing environment without hooking up to unused tools and infrastructure. Most importantly, the on-demand and unit-based testing allows enterprises to pay for a particular unit of testing rather than spending on the whole stack that in a way removes the budget lock-in.

Automated testing: Automated testing is the act of conducting specific tests via automation (e.g. a set of regression tests) as opposed to conducting them manually, while test automation refers to automating the process of tracking and managing the different tests.

Quality certification by third parties: Third-party certification means that an independent organization has reviewed the development process of a software and has independently determined that the final product complies with specific standards for safety, quality or performance.

Transparency: TaaS is measured as service and resources can be measured on both side. This providing transparency for both the provider and consumer - it allows pay-as-you-test pricing system.

Accessibility: The capability to provision server dynamically and the provision to deploy or remove software dynamically allows organizations to reduce the initial investment cost. The availability of different licensing models, like pay per use, can further help in reducing costs. In the event the software development organization decides to stop operation due to financial or strategic reasons, it is easier for the organization to opt out as there is not much resources "locked-in", if at all any. This is unlike the difficulty to exit operations faced by an organization that has made a lot on investment in acquiring these infrastructures, storage and operating systems to carry out its testing activities on site.

Agility: Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.As previously mentioned TaaS makes it possible to significantly reduce testing time and costs without compromising quality and enables organizations to be more agile in delivering critical business applications to their users.

Easily updated: The test-service provider can easily update the tool whenever necessary and this is completely transparent for the user. In traditional testing bugs or undesirable behavior must be reported by the user to the tool provider, who must then reproduce them, correct them and send updates to the user, who must then install them.

3.2. RQ2. What are the main disadvantages described in the literature when using TaaS?

This section lists and explains the disadvantages reported by the final selection of studies.

Requires special technical skill: Testing in the cloud might require special technical skills to appropriately utilize the TaaS platform to generate test cases and scripts.

Data Security and Integrity: Data security refers to the process of protecting data from unauthorized access and data corruption throughout its life cycle. Data security includes data encryption, tokenization, and key management practices that protect data across all applications and platforms.

Many providers allow users to submit binaries or executables for testing. This option places less confidential or proprietary information in the cloud environment, protecting intellectual property. When development code is examined rather than binary packages, testing software on the cloud creates a potential risk of intellectual property being compromised. The usage of binaries mitigates this risk, but does not eliminate it. Additionally, testing may be dependent on specific test data that necessitates actual client information.

While most end users have benevolent intentions, others could use these tools to locate and exploit weaknesses in a software product. The potential of malicious usage, though, may also result in a push for better software quality. This is also a major concern particularly when the testing activities are completely outsourced to a cloud test service provider. The cloud vendor's personnel could be easily "tapped" for information about the development organization's product and be offered a reward for such unscrupulous act. For example, the cloud vendor's staff could trade in signify cant features of a product yet to be released to the rival organizations. The rival company could then strategically match or even better the feature and incorporate this into their similar product. This puts the development organization in severe risk of losing their competitive advantage in such scenario. To reduce this risk, consequences of information disclosure must be reasonably severe, explicitly communicated and stated in a non-disclosure agreement offered to cloud vendor personnel.

Possible non Availability: Downtime, outages, and the resilience of the IT environments which store their core assets. Several of the provider challenges have the potential to translate into consumer issues. One example is the demand for constant global availability. TaaS providers must have their services and technical support available at all time, especially when working in international markets. While a difficult problem to solve on the provider side, the inability to test or the loss of test data due to an outage can cripple a consumer. Furthermore, providers must develop and maintain testing services and technical and support skills that match or exceed consumer needs.

Reliability: Probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system availability.

Lack of common standards: Until to the present moment, no studies have been found in the literature that report processes or methodologies that offer standards to the whole process of TaaS. Therefore, a lack of common standards can be listed like a disadvantage in this context.

Lack of full automation: Some platforms require the test user to enter parameters at each stage of the test run, these tests are not fully automated and may require user knowledge.

Vendor Lock-In: Known as proprietary lock-in or customer lock-in, makes a customer dependent on a platform for testing services, unable to use another vendor without substantial switching costs. Lock-in costs that create barriers to market entry may result in antitrust action against a monopoly.

4. Conclusion

In order to characterize and define some characteristics as advantages and disadvantages of the testing as a service platforms, this study had as objective to analyze the reports of experiences and publications of the literature. In this sense, a systematic literature mapping was carried out to identify and consolidate the related work, thus providing a good document that can assist and guide the decision-making processes in the test engineering area, as well as directing future research to contribute to this body of knowledge. With the summarization, it was possible to verify the number of papers published in the area. The earliest paper was published in 2008 and the last one in late 2018. These numbers suggests a growth curve from 2015, reaching its apex in 2017. Thus, our conclusion is that the research area is very recent, in expanse, and demonstrates an increase of interest by the academic community.

Referências

Ammann, P. and Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.

Floss, B. and Tilley, S. (2013). Software testing as a service: An academic research perspective. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, pages 421–424.

Gao, J., Bai, X., Tsai, W., and Uehara, T. (2013). Testing as a service (taas) on clouds. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, pages 212–223.

Garriga, M., Mateos, C., Flores, A., Cechich, A., and Zunino, A. (2016). Restful service composition at a glance: A survey. *Journal of Network and Computer Applications*, 60:32 – 53.

Harikrishna, P. and Amuthan, A. (2016). A survey of testing as a service in cloud computing. In *2016 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–5.

Mell, P. M. and Grance, T. (2011). Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States.

Myers, G. J. and Sandler, C. (2004). *The Art of Software Testing*. John Wiley & Sons, USA.

Perry, W. E. (2007). *Effective methods for software testing: Includes complete guidelines, Checklists, and Templates*. John Wiley & Sons.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, pages 68–77, Swindon, UK. BCS Learning & Development Ltd.

Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Souza, F. C., Santos, A., Andrade, S., Durelli, R., Durelli, V., and Oliveira, R. (2018). *Automating Search Strings for Secondary Studies*, chapter 558, pages 839–848. Springer International Publishing.

Yu, L., Zhang, L., Xiang, H., Su, Y., Zhao, W., and Zhu, J. (2009). A framework of testing as a service. In *2009 International Conference on Management and Service Science*, pages 1–4.

Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18.