Perceived needs and gains from an industrial study in Cloud testing automation

Antonia Bertolino, Antonello Calabrò, Eda Marchetti
Istituto di Scienza e Tecnologie dell'Informazione "A.Faedo"
Consiglio Nazionale delle Ricerche
Pisa, Italy
{name.surname}@isti.cnr.it

Anton Cervantes Sala
Worldline Iberia SAU
Mobile Competence Center
Barcelona, Spain
anton.cervantes@worldline.com

Guiomar Tuñón de Hita
Naeva Tec
Las Rozas (Madrid), Spain
gtunon@naevatec.com

Ilie Daniel Gheorghe Pop
Fraunhofer FOKUS Institute
Berlin, Germany
ilie-daniel.gheorghe-pop@fokus.fraunhofer.de

Varun Gowtham
Technische Universität Berlin
Berlin, Germany
v.gowtham@tu-berlin.de

***Abstract*—Challenges, methods and tools for testing in the Cloud have been actively researched, however there is lack of evidence about the actual motivations, issues and gains for adoption of automated cloud testing technology in real world industrial contexts. In this paper we report our findings from an empirical study involving four quasi-experiments within different application domains, namely e-commerce, 5G networking, WebRTC and IoT. The study is part of the ElasTest validation strategy, aiming at assessing the impact of the ElasTest open source platform for end-to-end testing of large distributed systems.**

***Index Terms*—Cloud testing, industrial demonstrator, quasi-experiment.**

I. Introduction

Testing is a crucial activity in quality assurance, yet end-to-end testing of large complex distributed systems is increasingly complex and expensive [1]. Today these systems are typically created by interconnecting and orchestrating smaller component systems, and in this paper we refer to them as SiLs (Systems in the Large).

The Cloud promises to mitigate the difficulties and costs involved in SiL testing by allowing testers to exploit powerful elastic technologies and virtually unlimited resources. Testing SiLs in the Cloud can emulate desired non-functional properties, improve reusability and automation, and conveniently scale up testing scenarios.

A. Related work

In recent years much research has been conducted on methods and tools for Cloud testing, as overviewed in, e.g., [2]–[5]. However there is still not much evidence about the actual needs and gains that are experienced in industrial contexts while moving testing to the Cloud. This was also observed by Riungu-Kalliosaari and coauthors in [6]. Their qualitative study reports an analysis of the responses from 35 interviews with testers from 20 organizations. In contrast this work reports the quantitative results from a set of quasi-experiments conducted by four organizations on adoption of Cloud testing.

A more recent systematic mapping study by Ahmad and coauthors in 2017 [7] reviewed a selected set of 75 research papers published in the period (2010-2015), focusing in particular on what is the application domain and how the proposed Cloud testing approaches are evaluated. The study concluded that most "*studies present only preliminary results, often describing an example of the software cloud-based testing methods or a simple application experiment to evaluate the proposed approach*" [7]. In this sense, this paper aims at providing a broader feedback from a field study in which we measured a set of relevant indicators before and after moving testing to the Cloud.

B. ElasTest

Our study has been performed using an advanced platform supporting automated testing in the Cloud. Precisely, we used the open source platform ElasTest that is developed within an European H2020 project [8]. The ElasTest project aims at increasing the efficiency, productivity and reusability of SiL testing, and also, while improving the effectiveness of the testing process, at achieving higher product quality. In brief, ElasTest is a comprehensive framework for deploying and testing large distributed systems; it leverages Cloud resources for facilitating functional and non functional testing, and offers many test services, as illustrated in Figure 1. The platform innovation builds on three principles:

- Test orchestration: combining intelligently testing units for creating a more complete test suite following the *divide and conquer* principle. In particular it is focused on reusable testing services solving common testing problems including browser automation, sensor emulator, monitoring, security check, log ingestion and analysis, cost modeling , etc.

This paper describes work undertaken in the context of the European Project H2020 731535: ElasTest.

Fig. 1. ElasTest offered services

- Capabilities for the instrumentation of the Software under Test enabling to reproduce real-world operational conditions thanks to features such as Packet Loss as a Service, Network Latency as a Service, Failure as a Service, etc.
- Cognitive computing and machine learning mechanisms suitable for ingesting large amounts of knowledge (e.g. specifications, logs, software engineering documents, etc.) and capable of using it for generating testing recommendations and answering natural language questions about the testing process.

For lack of space we refer the reader interested to learn more about ElasTest to the rich reference documentation made available from the project site [8].

C. This study

Since the ElasTest platform is agnostic of specific technologies or processes, we could leverage it for evaluating Cloud testing over a set of four different demonstrators. Each of the four demonstrators, which we describe later in Section II, involves different application domains, uses different testing methodologies, and aims at different testing purposes. Therefore, though preliminary, the present study provides a nice overall perspective in assessing Cloud testing automation.

In summary the study confirmed that automated testing in the Cloud may bring some attractive gains in terms of scalability and reusability. However, we also learned that one needs to be very careful when claiming gains of Cloud testing, because every context may yield specific challenges and unforeseen problems. In the remainder of the paper we summarise our findings from the quasi-experiments performed with four different partners of the ElasTest project.

D. Roadmap

The paper is structured as follows: In this section we have provided motivations and a brief introduction to our study, including a short overview of related work. In the next section we present the four demonstrators and their perceived needs. In Section III we report the performed study, and in Section IV some observed results. Finally we draw brief conclusive remarks.

II. BACKGROUND

The involved partners in the proposed study are:

A. Worldline

Worldline [9] is the responsible for the development of an e-commerce demonstrator. E-commerce applications deal with payment gateways and for this reason increasing the confidence on the end user side by minimizing any possible quality issue like security threats is a must for these kinds of web applications. Among other aspects, testing such applications provides the challenge of validating their behavior under many different software configurations. The application has been selected because it is the most representative web application in Worldline, before the introduction of the features of ElasTest project. The product considered in the experiment is a web application involving more than 10 single functionalities tested using a black box approach. One of the most important aspects for an e-commerce client is to be sure the platform is working perfectly from all kind of browsers and versions of those browsers.

B. FOKUS

Fraunhofer FOKUS [10] provides an Open5GCore [11] toolkit. This is a practical implementation of the carrier-grade network towards the 5G environment. It mirrors, in a prototypical form, the pre-standard advancements on the core network, radio network integration, distributed management and virtualization. The Open5GCore aims at providing support and speeding-up research, facilitating know-how transfer from Fraunhofer FOKUS towards partners. It serves as a consistent basis for research projects with meaningful results, enabling

fast and targeted innovation, hand-on fast implementation, realistic evaluation and demonstration of novel concepts and technology opportunities.

C. Naeva Tec

Naeva Tec [12] is responsible of the FullTeaching demonstrator [13], which covers multiple kinds of communication in an educational context. FullTeaching is an open-source educational web application to make online classes easy for teachers as well as students. FullTeaching provides multiple features for managing courses, contents, calendar of sessions, etc. However its main features are those related to communication either offline (forums and video forums) or real-time (video sessions and chats). In ElasTest demonstrator, the main focus is on testing Real Time communications based on the WebRTC [14]. WebRTC is a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs. Naeva Tec is a Spanish technological company devoted to development of Internet and mobility communications solutions, and technology consultant. Is has been involved in success stories like Kurento Media Server. It has customers worldwide and some of the main telco companies in Spain. WebRTC is a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs.

D. TUB IIoT

TUB [15] provides a demonstrator based on the OpenIoT-Fog [16] (OIF), which is a set of Industrial Internet of Things (IIoT) applications that involves sensors and actuators commonly found on the industry shopfloors deployed on fog/edge nodes. OpenMTC lies at the core of OIF, enabling Machine to Machine (M2M) communication between applications and used as a middleware. A typical IIoT application comprises many sensors, logic and actuators. Thus the demonstrator is able to get data from the sensors, apply logic to the data and flag the actuators based on the logic. The SiL is composed of several applications and provides a wide general variety of possible sensors and actuators.

E. Validation plan

The four above introduced demonstrators have been used in the following validation process:

1) A preliminary survey among the four different partners for requirements elicitation;
2) The setting up of the validation strategy, including the target metrics, and the preparation of all forms to collect data and questionnaires;
3) The collection and analysis of preliminary results;
4) The collection of possible improvements and suggestions for the subsequent ElasTest releases.

Here below we report in cumulative way some main requests among those collected from the four demonstrators during the requirement elicitation step (a more detailed report can be found in [17]):

- Req1: reducing the delivery time of the final product
- Req2: improving the reusability of the test cases
- Req3: speeding up maintenance activity
- Req4: making easier the execution of test plans by not expert testers
- Req5: testing the SUT in different environments

III. Quasi Experiment study

The requirements collected during the elicitation phase have been successively used for setting up the validation process that includes the following steps: definition of the target metrics; selection of the experiment to be performed; definition of the forms and questionnaires to be used for data collection during the experiment running.

Considering the metrics used during the validation process, we provide below a list of the most relevant ones. For aim of simplicity we report the main target metrics selected for each of the requirements specified in the previous section.

- M1: reduction the overall time to market of SiL
- M2: improvement of the reusability of code, tools and architectures devoted to non-functional software testing on SiL
- M3: decrease in the corrective maintenance effort of SiL
- M4: increase of the tester perceived usefulness when involved in testing tasks for SiL
- M5 increase of the scalability (measured as the total number of concurrent supported sessions) of SiL

In the above list M1, M2, M3 and M5 are objective metrics collected through specific forms, while M4 relates to a subjective metric that has been collected through questionnaires distributed among the testers involved in the experimentation.

The data collection has been performed through four related Quasi Experiments (QEs) [18]. A QE is a controlled study used to estimate the causal impact of an intervention on its target population without random assignment to treatment or control.[1] In particular, for all four vertical demonstrators, the general procedure for the QE assessment includes two different teams of testers performing the testing activity on the same application:

- the Without ElasTest group (WO): the team following the best test practices in the company and using the commonly used tools and facilities;
- the With ElasTest group (WE): the team performing the testing activities with the support of the ElasTest platform.

During the testing activity both groups collected the testing data useful for the metrics evaluation through some specific provided forms. At the end of the experiment, the questionnaires have been distributed and the replies collected.

A same version of the ElasTest platform has been frozen and distributed to the four partners (Release "0.6.0-pre4-beta4" [17]). Indeed the ElasTest platform is being developed using the agile development paradigm, thus different releases are systematically issued every four months. This provides a dynamic development environment where: from one side,

[1]Metrics example: goo.gl/gAutTH - Survey example: goo.gl/Ff8H1h

the product is improved according to iterative user/tester suggestions and issues (bugs) reported; from the other, a continuous integration is performed in order to guarantee the prefixed quality level. According to this schema each release provides a set of ready-to-use features through a Docker hub platform [19].

For the ElasTest version adopted during the QE the features available were:

- a rich-full web interface able to create Test projects, SuTs (Systems Under Test): managed by ElasTest (docker-compose) and outside ElasTest (instrumented using hooks provided by ElasTest), TJob;
- an on-line interface providing a quick and clear glimpse on what is happening during the SUT deployment and during the test execution;
- a powerful log analyzer able to keep track of what is happening on the several components involved in testing in each specific time interval (eg. check CPU consumption when a specific test is executed or check network usage when images are downloaded or test streaming is started);
- the User Impersonation Service (EUS) supporting the tester while executing end-to-end testing of web applications;
- the TestLink [20] management system integrated within ElasTest.

From a procedural point of view, the QE has been implemented considering the following steps:

1) The testing team (WO and WE) starts analyzing the documentation
2) The Development Team develops (finalizes) the first version of the SUT
3) The Quality Analyst (QA) prepares a common set of test directives
4) The test directives are translated into a WO test plan and a WE test plan by the WO testers and WE testers respectively
5) In parallel testers in each team start testing and possibly raise bugs, namely: Testers in WO (WE) start testing and raise bugs with label WO (WE respectively)
6) Dev team collects both the WO and WE bugs and fixes them. Specifically the bug fixes are not deployed until both teams (WO and WE) have finished the first test iteration.
7) Once both teams (WO and WE) finish their first test iteration, Dev Team deploys a new version of SUT with all the bugs fixed.
8) The QA starts a new iteration of testing.

IV. Results

In this section, considering the metrics listed in Section III, we report an overview of results collected during the QE.

In particular in each of the Worldline, Naeva Tec and TUB IIoT demonstrators, during the testing activity the same number of functionalities has been tested using the same test plan by the same number of testers in both the WE and WO branches. For the Open5GCore demonstrator four customers have been selected, two for the WE and two for the WO. All customers had to test the software within a pre-established time window, and precisely they all submitted the results of the testing process after a four weeks evaluation period.

In general the data reported during the QE execution evidenced some positive experience of the four involved demonstrators, but also interesting feedbacks and suggestions for the future improvement of ElasTest platform. In the following more details about the metrics evaluation are provided as well as the feedbacks collected. For space limitation, we only provide the main results and refer to [17] for a complete report.

A. Time to market

Considering the reduction of the overall time to market (TTM) of SiL, Figure 2 depicts the situation experienced during the QE execution for the four partners involved. TTM has been computed considering different stages of the testing process that are: Unitary, Integration, System and End2End. Indeed not all the partners included all the four stages in their testing activity or could clearly distinguish among them. At top of each diagram, the cumulative percentage of TTM reduction for each of the demonstrators is reported. As shown in the diagrams different percentages have been collected. They span from the Naeva Tec and TUB IIoT extremely positive results in favor of WE stage (93.26% and 64.89% TTM reduction respectively), to the very negative result for WE reported by Worldline (-50% TTM reduction).

In the last case the figure evidences that, in case of integration test, the total amount of time necessary for the completion of the WE branch is greater than for the WO. A deeper analysis revealed that the most influencing factor during all the three stages for the WE part was the analysis and debugging activity. Worldline general expectation is that the additional facilities provided by ElasTest in the next releases could considerably reduce this time. However, comparing WO and WE settings in number of defects found, the WE revealed one more fault than in WO during the testing phase. The tester reported that, even if the debugging activity was time consuming, the bug was discovered because the ElasTest real time log display helped him to visualize what was happening in the server side.

The TTM collected from the FOKUS Open5GCore demonstrator did not show significant differences between the WO or WE cases. However, the experiment evidenced that the most influencing phase in both cases was the Integration one: it almost doubled the completion time of the other stages. This provided FOKUS with a significant amount of feedbacks towards improving its current testing activity.

For Naeva Tec the figure evidences that for WO the most critical phase for the TTM was the Unitary test; End2End testing was still in favor of the WE even if with less impact. A deeper analysis evidences that the difference was mainly due to the coding activity required in case of WO for developing the required test suite.

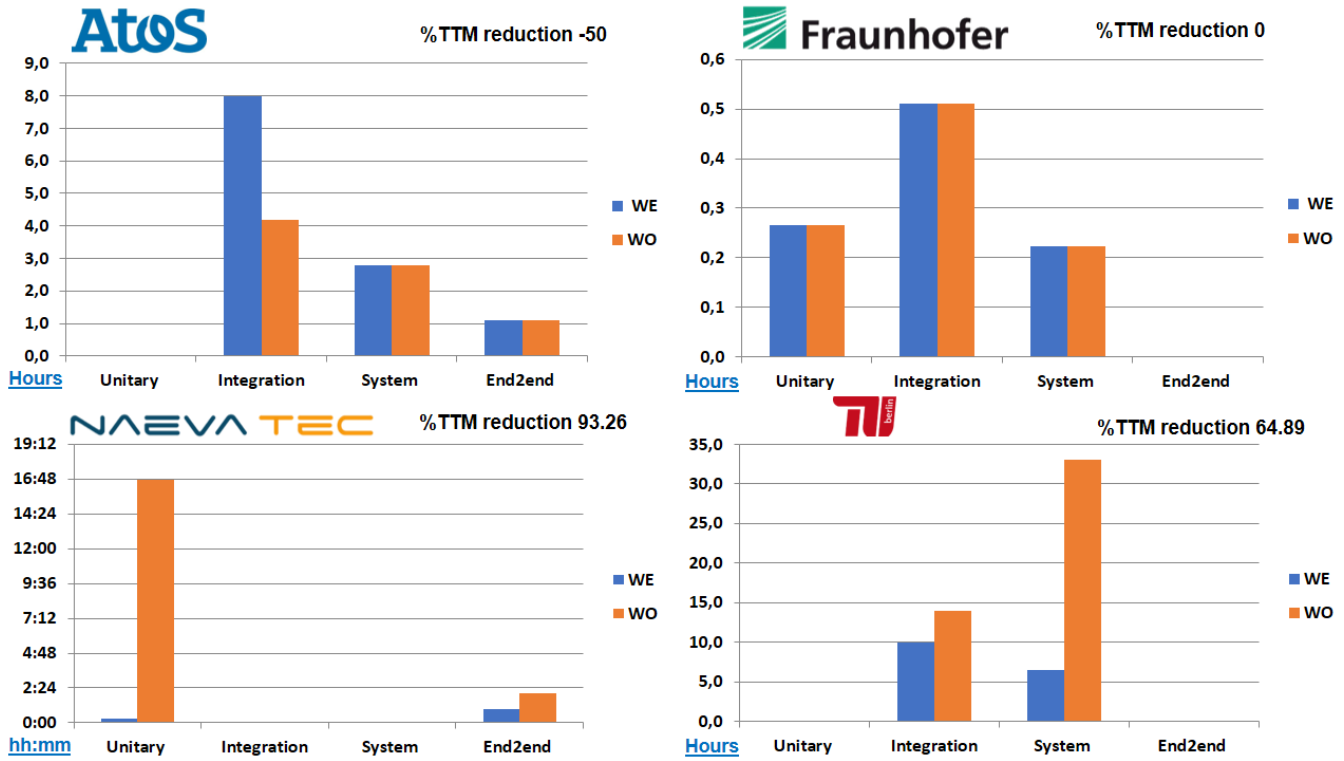

Fig. 2. Time to Market

Finally for TUB IIoT in the WO, the most influencing phase was the System one. The difference was mainly due to the time dedicated in results analysis and debugging during the WO branch. Differently from the Worldline experience, this partner believes that currently available ElasTest features supporting result analysis were indeed useful.

B. Reusability

The reusability of code, tools and architectures devoted to non-functional software testing on SiL is an important issue for all the partners involved in the experimentation. However, in this preliminary study for the limited set of facilities provided by the ElasTest, only TUB IIoT was able to collect reusability measures.

In this case the partner focused on reusability of test cases and facilities. In particular during the WE and WO stages the same test plan has been developed producing 18 test cases for the WE and 19 for the WO. In terms of reusability the overall percentage of test reused was greater in the WO (42.11% vs 22.22%) than in the WE. This was because the WO tester wrote a specific application for launching the different rounds of test execution emulating in some sense the role of ElasTest. This was evident only after a post analysis of the data collected and it was not possible to recover data about the time and effort necessary for writing this kind of application. However, the same percentage of reused facilities has been experienced in the WE and WO experiment evidencing the feasibility of the automated approach.

C. Maintenance

The corrective maintenance has been measured considering the time necessary for analyzing the collected test results and identifying the possible faults in the code. Figure 3 reports the corrective maintenance evaluation for three of the four verticals in the WE and WO stages. Unfortunately data on this metric were not provided by the FOKUS customers involved in the experiments.

As shown in the figure the use of ElasTest provided positive results for Naeva Tec and TUB IIoT. In this last case the time for WO was almost 10 time greater than WE branch. The opposite situation has been reported by Worldline where the WE time is greater than WO. This reflected the problems highlighted during the TTM evaluation (see Section IV-A) for test debugging.

D. Usefulness

The tester perceived usefulness when involved in testing tasks for SiL has been evaluated through a dedicated questionnaire distributed among the testers involved in the QE steps. For space limitation we report here only two of the cumulative scores collected during the experiment, referring to [17] for more details. In particular in Figure 4 we report the results relative to the question: *To what degree are you able to use the testing environment without the support of a technical expert?*, where the adopted scale was: 1 very difficult to use; 3 neutral; 5 very simple to use. As shown in the figure the testers in the WE stage considered the automatic facility

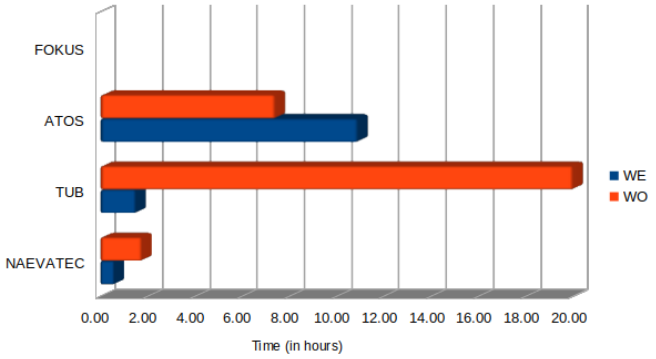

Fig. 3. Corrective maintenance

a good improvement for their testing activity: the arithmetic mean was 3.86 and 2.71 for the WE and WO respectively.

Additionally in Figure 5 we report the results relative to the question: *To what degree do you think your testing environment is intuitive?* with scale 1-5 as in the previous question. In this case the results for the WE were evidently in favor of ElasTest (with arithmetic mean of 4.17 and 2.71 for WE and WO respectively).

A deeper analysis of the replies collected revealed in particular that for the FOKUS customers involved in the WE branch, one of the major issues was the long time necessary to adapt themselves to the new software evaluation/testing process. However, eventually the WE customers reported that the testing activity was easier with the help of ElasTest. In particular they appreciated: the test Instrumentation, the test definition and execution, the test composition features, the test templating/ reusability/ test repository and finally the automatic deployment and execution of all validation tests as a showcase with test orchestration, as part of the deployment process.

In the case of Naeva Tec, initially some issues have been raised during installation due to the complexity of the application itself. Specifically difficulties arose because the application runs on a webserver, has a database and requires a specific framework to be executed. As a general feedback the tester in charge of the WE experimentation evaluated the ElasTest version as user friendly and very helpful to setup and run the test cases. In particular the Docker technologies made it easier to manage the different components and to include and run the Docker containers. Moreover, the log facilities were very useful for bug analysis because they also allowed to track resources usage during the execution of the tests.

E. Scalability

During this round of QE scalability was not measured because some important features were still missing in the version of ElasTest used for the experimentation. We are planning to analyze scalability data on the next already scheduled QE iterations.

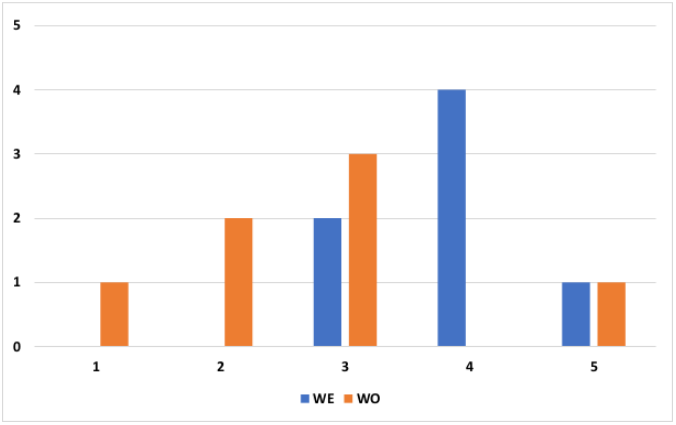

Fig. 4. To what degree are you able to use the testing environment without the support of a technical expert?

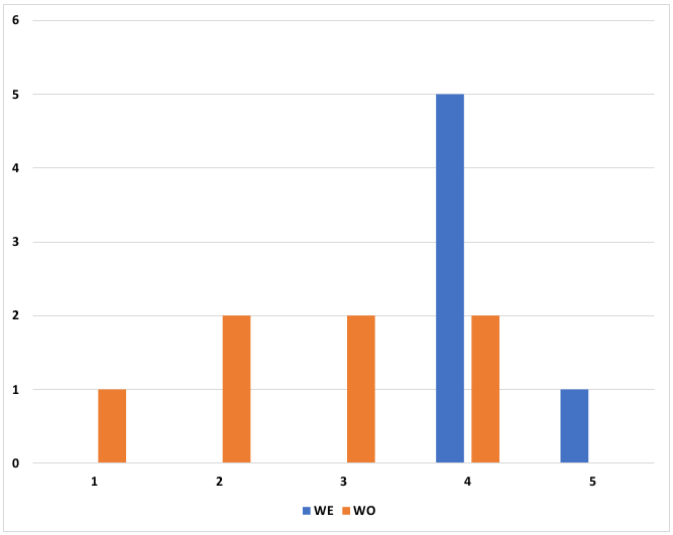

Fig. 5. To what degree do you think your testing environment is intuitive?

F. Discussion

The results reported in the previous sections evidenced that the version of ElasTest used in the QE study is still incomplete, and some additional features would be needed to become the winning solution in the context of Cloud testing. We were aware of this, as the project has still more than one year to go before its completion. So, the important result for us is to understand what are the improvements needed by the partners involved in the QE.

Most feedbacks and improvement suggestions were relative to the strategy for visualization and management of test cases: for instance, including real-time log alarms during test case execution; including counters of occurrences of each search-pattern; coloring the lines that meet search-patterns, and so on.

Improvements were also required to speed up the test execution performance, e.g., by distributing test execution over multiple machines. In the WE branch of the experiment the

most encouraging result was the possibility of detecting certain anomalies / bugs that in the traditional way of testing would have remained undetected.

Concerning threats to validity of the presented QE, four aspects can be considered: the version of the Elastest platform, the nature of the demonstrators, the metrics adopted and the experience of involved testers. Indeed, all these elements could have influenced the reported results. Moreover the choice of performing a controlled study for assessing the metrics does not assure that different choices (i.e., a comparative case study) would provide comparable results. Finally, the size of the sample is of course not large enough to provide any statistical significance for the collected measures. Indeed, this paper has to be taken as a presentation of ongoing experimentation. As said, the main goal of the study was to provide a first feedback from a field study of automating testing in the Cloud. More studies are planned in the course of the project to address the above threats.

V. Conclusions

In this paper we reported the preliminary results collected from four Quasi Experiments conducted over real world large-scale applications. The study focused on assessing potential advantages of automated testing in the Cloud. In particular we used the open source platform ElasTest, one of the most innovative and comprehensive frameworks for deploying and testing large distributed systems.

The subjects of the study were four applications representative of different technologies, processes and testing approaches adopted in widely differing development contexts. Notwithstanding such diversity, the technology-agnostic nature of ElasTest permitted us to smoothly collect a series of observations and metrics from the four demonstrators.

At this stage, we observed both favorable and unfavorable results. However, given the preliminary nature of the study, from the project perspective we consider all of them useful and positive. In particular, we are learning from less favorable results how ElasTest, and more in general automated testing in the Cloud, can be improved in future releases. The running of the QEs not only demonstrated the potential of ElasTest but also highlighted the different on-the-field stringent specification requirements coming from different domains, otherwise not easily identifiable.

As future work we would like to integrate elicited requirements in the ElasTest platform as well as to extend the ElasTest assessment over further application domains.

References

[1] CapGemini, Sogeti, and Micro Focus, "World quality report, 9th ed." 2017-18.

[2] J. Gao, X. Bai, and W.-T. Tsai, "Cloud testing-issues, challenges, needs and practice," *Software Engineering: An International Journal*, vol. 1, no. 1, pp. 9–23, 2011.

[3] S. Vilkomir, "Cloud testing: A state-of-the-art review," *Information & Security*, vol. 28, no. 2, pp. 213–222, 2012.

[4] K. Incki, I. Ari, and H. Sözer, "A survey of software testing in the cloud," in *Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 18–23.

[5] (2018) D2.2 SotA revision document v1. [Online]. Available: https:///elastest.eu//resources//deliverables/D2.2_SotA_revision_document_v1_FINAL.pdf

[6] L. Riungu-Kalliosaari, O. Taipale, K. Smolander, and I. Richardson, "Adoption and use of cloud-based testing in practice," *Software Quality Journal*, vol. 24, no. 2, pp. 337–364, 2016.

[7] A. A.-S. Ahmad, P. Brereton, and P. Andras, "A systematic mapping study of empirical studies on software cloud testing methods," in *IEEE Int. Conf. on Software Quality, Reliability and Security Companion (QRS-C)*. Prague, Czech Republic: IEEE, July 2017, pp. 555–562.

[8] (2018) ElasTest. [Online]. Available: https://elastest.eu/

[9] Worldline. [Online]. Available: https://worldline.com/

[10] Fraunhofer. [Online]. Available: https://www.fokus.fraunhofer.de/

[11] Open5gCore. [Online]. Available: https://www.open5gcore.org/

[12] Naeva Tec. [Online]. Available: https://www.naevatec.com/

[13] Full Teaching. [Online]. Available: https://github.com/OpenVidu/full-teaching

[14] Web RTC. [Online]. Available: https://webrtc.org/

[15] TUB. [Online]. Available: https://www.av.tu-berlin.de/next_generation_networks/

[16] OpenIoT Fog. [Online]. Available: https://openiotfog.org/en/

[17] (2018) D7.1 ElasTest Validation methodology and its results v1. [Online]. Available: https:///elastest.eu//resources//deliverables//D7.1_ElasTest_validation_methodology_and_its_results_v1_FINAL.pdf

[18] B. Gribbons and J. Herman, "True and quasi-experimental designs. eric/ae digest." 1997.

[19] Elastest dockerhub. [Online]. Available: https://hub.docker.com/r/elastest/

[20] Testlink open source test management. [Online]. Available: http://testlink.org/