Testing Distributed and Heterogeneous Systems: State of the Practice

Bruno Lima[1,2] and João Pascoal Faria[1,2]
[1]*Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal*
[2]*INESC TEC, FEUP campus, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal*

Keywords: Software Testing, Distributed Systems, Heterogeneous Systems, Systems of Systems, State of the Practice.

Abstract: In a growing number of domains, such as health-care and transportation, several independent systems, forming a heterogeneous and distributed system of systems, are involved in the provisioning of end-to-end services to users. Testing such systems, running over interconnected mobile and cloud-based platforms, is particularly important and challenging, with little support being provided by current tools. In order to assess the current state of the practice regarding the testing of distributed and heterogeneous systems (DHS) and identify opportunities and priorities for research and innovation initiatives, we conducted an exploratory survey that was responded by 147 software testing professionals that attended industry-oriented software testing conferences, and present the main results in this paper. The survey allowed us to assess the relevance of DHS in software testing practice, the most important features to be tested in DHS, the current status of test automation and tool sourcing for testing DHS, and the most desired features in test automation solutions for DHS. We expect that the results presented in the paper are of interest to researchers, tool vendors and service providers in this field.

1 INTRODUCTION

Due to the increasing ubiquity, complexity, criticality and need for assurance of software based systems (Boehm, 2011), testing is a fundamental lifecycle activity, with a huge economic impact if not performed adequately (Tassey, 2002). Such trends, combined with the needs for shorter delivery times and reduced costs, demand for the continuous improvement of software testing methods and tools, in order to make testing activities more effective and efficient.

Nowadays software is not more like simple applications but has evolved to large and complex system of systems (DoD, 2008). A system of systems consists of a set of small independent systems that together form a new system. The system of systems can be a combination of hardware components (sensors, mobile devices, servers, etc.) and software systems used to create big systems or ecosystems that can offer multiple different services. Currently, systems of systems capture a great interest from the software engineering research community. These type of systems are present in different domain like e-health (AAL4ALL, 2015) or transportation (Torens and Ebrecht, 2010).

Testing these distributed and heterogeneous software systems or systems of systems, running over interconnected mobile and cloud based platforms, is particularly important and challenging. Some of the challenges are: the difficulty to test the system as a whole due to the number and diversity of individual components; the difficulty to coordinate and synchronize the test participants and interactions, due to the distributed nature of the system; the difficulty to test the components individually, because of the dependencies on other components. Because of that, the attention from the research community increased, however, the issues addressed and solutions proposed have been primarily evaluated from the academic perspective, and not the viewpoint of the practitioner.

Hence, the main objective of this paper is to explore the viewpoint of practitioners with respect to the testing of distributed and heterogeneous systems (DHS), in order to assess the current state of the practice and identify opportunities and priorities for research and innovation initiatives. For that purpose, we conducted an exploratory survey that was responded by 147 software testing professionals that attended industry-oriented software testing conferences, and present the main results in this paper. Besides introductory questions for characterizing the respondents and contextualizing their responses, the survey contained several questions with the aim of assessing the practical relevance of testing DHS, the importance of testing several features of DHS, the current level of test automation and tool sourcing, and the desired fea-

Lima, B. and Faria, J.
Testing Distributed and Heterogeneous Systems: State of the Practice.
DOI: 10.5220/0005989100690078
In *Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016) - Volume 1: ICSOFT-EA*, pages 69-78
ISBN: 978-989-758-194-6

tures in test automation solutions for DHS. We expect that the results presented in the paper are of interest to researchers, tool vendors and service providers in the software testing field.

The rest of the paper is organized as follows: Section 2 presents the research method used to conduct the survey. Section 3 presents the results, which are further discussed in Section 4. Section 5 describes the related work. Section 5 concludes the paper and points out future work.

2 RESEARCH METHOD AND SCOPE

The research method used in this work is the explanatory survey. Explanatory surveys aim at making explanatory claims about the population. For example, when studying how developers use a certain inspection technique (Wohlin et al., 2003).

2.1 Goal

The main goal of this work is to explore the testing of DHS from the point of view of industry practitioners, in order to assess the current state of the practice and identify opportunities and priorities for research and innovation initiatives.

More precisely, we aim at responding to the following research questions:

- **RQ1**: How relevant are DHS in the software testing practice?
- **RQ2**: What are the most important features to be tested in DHS?
- **RQ3**: What is the current status of test automation and tool sourcing for testing DHS?
- **RQ4**: What are the most desired features in test automation solutions for DHS?

2.2 Survey Distribution and Sampling

Since our main goal was to collect the point of view of industry practitioners that were involved in the testing of DHS, we shared the survey to the participants of two industry-oriented conferences in the software testing area: TESTING Portugal 2015[1] and User Conference on Advanced Automated Testing (UCAAT) 2015[2]. In total we distributed 250 surveys and we obtained 167 answers. From these 167 answers, only 147 were complete and valid. Most of the invalid answers were related with respondents that did not complete the survey.

2.3 Survey Organization

The survey was composed of two main parts. The first part was an introduction, where we explained the goal of the survey and define the term "Distributed and Heterogeneous Systems" in the context of this survey. In the context of this survey we define a Distributed and Heterogeneous System as a set of small independent systems that together form a new distributed system, combining hardware components and software systems, possibly involving mobile and cloud-based platforms.

The second part of the questions is divided in three different groups. The first group is related with the professional characterization of the participants. Here, the survey participant needs to explain:

- what is his main responsibility in his current position (e.g., software testing, software developer, project manager);
- how many years he is in the current position;
- how many years he has been working in software testing;
- how many years he has been working with distributed and heterogeneous systems (related with RQ1).

The second group contains questions about the company characterization. This group has questions to identify:

- the industry sector;
- the company size;
- the role(s) under which the company conducts software tests (as developer, as customer/user, as independent test service provider, and/or as system integrator);
- the test levels performed by the company.

Regarding the last question, we asked the participants about the test levels normally considered in industrial practice (ISTQB, 2016a):

- unit (or component) testing - the testing of individual software components;
- integration testing - testing performed to expose defects in the interfaces and in the interactions between integrated components or systems;

[1] http://www.cvent.com/events/testing-portugal-2015/event-summary-a1a41d7f08674008b58e43454bb9f54a.aspx

[2] http://www.etsi.org/news-events/events/868-2015-etsi-ucaat

- system testing - testing an integrated system to verify that it meets specified requirements;
- acceptance testing - formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

The questions in the first two groups are useful for characterizing the respondents and confirming their relevance for the purposes of the survey. The last question in the first group is also important for answering RQ1.

The last group contains the questions related with the testing of DHS and the main research questions underlying the survey:

- in what role(s) does the company conduct software tests for DHS, if any (related with RQ1);
- how important is the testing of several specific features of DHS (related with RQ2);
- what is the current status of test automation for DHS (manual testing, automatic test execution and/or automatic test generation) (related with RQ3);
- what is the current status of tool sourcing for testing DHS (commercial-off-the-shelf or developed in-house) (related with RQ3);
- what are the most desired features in test automation solutions for DHS (relate with RQ4);
- how useful would be an automatic test generation solution based on interaction models as input (relate with RQ4).

In the second question above (features to be tested), we asked the participants about features that are characteristic of DHS and raise significant testing challenges:

- interactions between the system and the environment - distributed systems usually have several points of interaction (ports) with the environment (users or external systems) that are themselves distributed, complicating test coordination (Hierons, 2014);
- interactions between components of the system - monitoring the interactions between components of the system under test (SUT) is important not only in the context of integration testing, but also in the context of system and acceptance testing, to facilitate fault localization, however such interactions are often obfuscated by middleware or are even encrypted, being difficult to intercept and interpret at an appropriate level of abstraction;
- parallelism and concurrency - components of DHS almost always exhibit some sort of parallelism and concurrency, which can be the source of subtle errors (race conditions, etc.) that are difficult to detect and replicate (West et al., 2012);
- timing constraints - in many DHS, several sorts of timing constraints (behaviors triggered by timeouts, response time limits, etc.) are imposed on interactions between components of the system or between the system and the environment, requiring the application of advanced test case generation and execution techniques (Kim et al., 2014) and the simulation of realistic operational conditions in the test environment;
- nondeterministic behaviors - nondeterminism may occur for various reasons in complex DHS, requiring adaptive test strategies (Petrenko and Yevtushenko, 2011) in which the next test action depends on the observed behavior of the SUT;
- multiple platforms - different components of a DHS may run on different platforms (operating systems, devices, browsers, etc.) and the same component may be deployed on different platforms, requiring complex testing infrastructures.

As for the most desired features in test automation solutions for DHS, besides the support for multiple platforms, we asked the participants about the test activities that can be fully or partially automated:

- support for automatic test case execution - this is the most widespread test automation facet in general, so we expect to be a popular desired feature for DHS;
- support for automatic test case generation - this provides the highest level of test automation, and some model-based testing (MBT) techniques and tools (Utting and Legeard, 2007) are attracting increasing interest from industry, but the need to build behavioral models of the SUT, together with limitations still existent in the techniques and tools (Dias Neto et al., 2007), hinder a wider adoption;
- support for test coverage analysis - test coverage analysis is better known in the context of white-box testing (using code coverage metrics) but can also be employed in the context of gray-box or black-box testing (e.g., using model coverage metrics);
- support for automatic test stub generation - test stubs are need in the context of unit (component) testing, to simulate other components on which the component under test depends, and, under some conditions, may be automatically generated

from specifications or models (Faria and Paiva, 2014).

Our final question was intended to evaluate the receptiveness of the participants to a type of input models - interaction models such as the ones depicted by UML sequence diagrams (OMG, 2015) - that are particularly well suited for generating test cases for the most relevant features of DHS (asked in the second question). In fact, UML 2 sequence diagrams provide a convenient means to specify the messages that are exchanged between components and actors of DHS under specific scenarios, and express parallelism, synchronization and time constraints, among other features (Lima and Faria, 2016).

3 RESULTS

3.1 Participants Characterization

Before drawing conclusions on the main questions of this survey it is important realize the profile of the survey participants. The results show that most of the people (70%) that responded this survey work in software testing, verification & validation (see Figure 1) and 41% are in the current position for more than five years (see Figure 2).

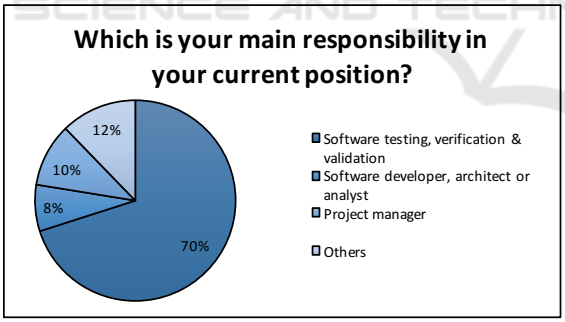

Figure 1: Current Position.

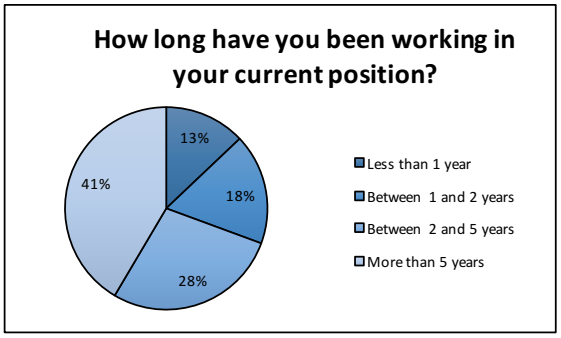

Figure 2: Time in Current Position.

Regarding the experience in software testing, the results show (see Figure 3 and 4) that the majority of the survey participants have more than 5 years of experience in software testing in general and 40% have more than 5 years of experience with DHS.

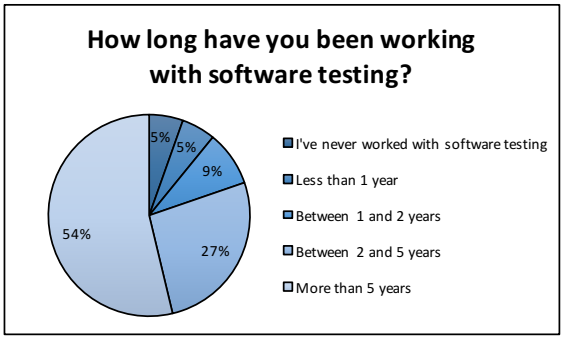

Figure 3: Time in Software Testing.

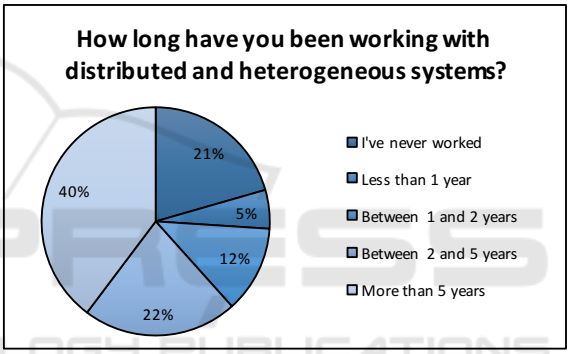

Figure 4: Time in Software Testing of DHS.

3.2 Company Characterization

The companies surveyed worked in a large range of industry sectors. The results represented in Figure 5 identify more than 10 different industry sectors.

We also analyzed the size of the companies according to their number of collaborators. Most of the companies are large companies, 37% have between 100 and 1,000 collaborators and 45% have more than 1,000 collaborators (Figure 6).

The answers to 'In what role(s) does your company conducts software test, if any?' show that half of the companies performs tests to the software developed by themselves (Figure 7).

Regarding the types of test levels performed, we realize from the answers (Figure 8) that the unit testing level is the less performed and the other three levels (integration, system and acceptance) are performed with the same frequency.

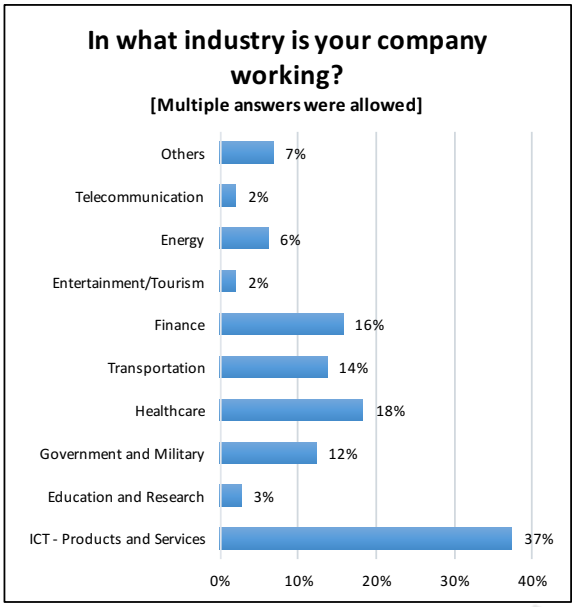

Figure 5: Industry Sectors.

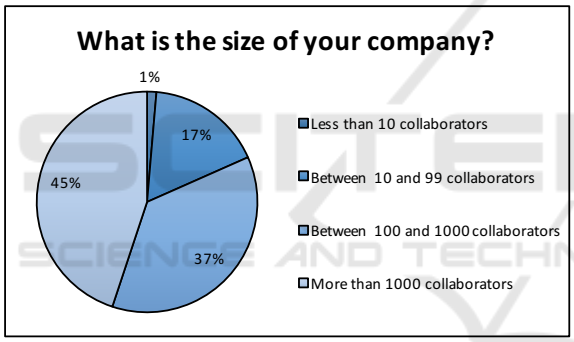

Figure 6: Company Size.

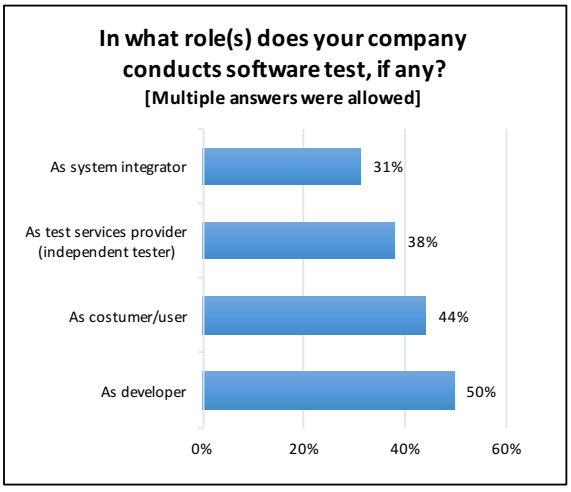

Figure 7: Company Roles.

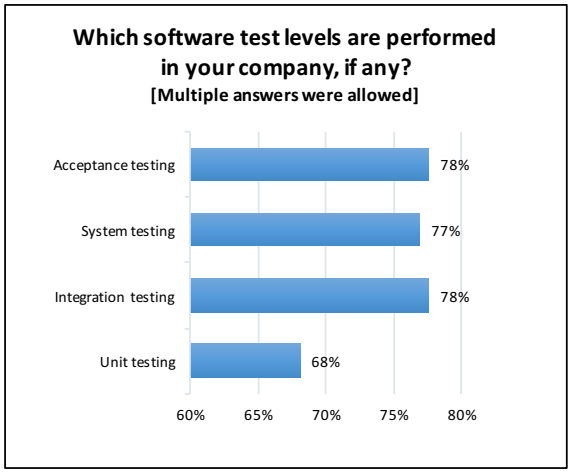

Figure 8: Test Levels.

3.3 Distributed and Heterogeneous Systems Testing

Focusing now on the main questions of this survey, specifically related to the testing of DHS, the answers to 'In what role(s) does your company conducts software test (for DHS), if any?' show that a vast majority of 90% of the companies (all but 10%) conducts tests for DHS in at least one role, with 42% of the companies performing tests for DHS developed by themselves (Figure 9).

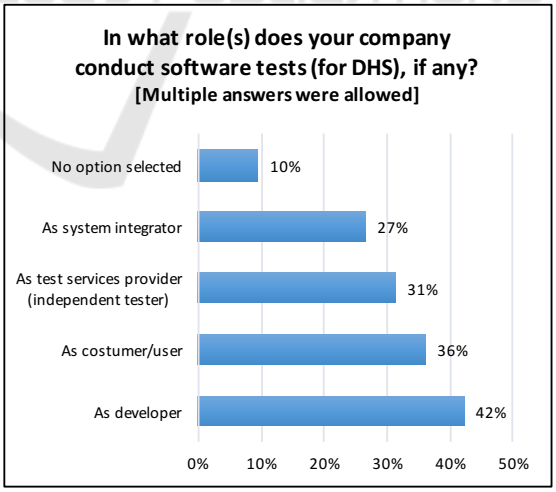

Figure 9: Test Roles DHS.

We also tied to understand what kinds of levels are most commonly used in the testing of such systems. Regarding the responses obtained (Figure 10), there is a higher emphasis on system testing (71%) followed by integration testing (65%). Only 8% of the respondents did not mention any test level for DHS.

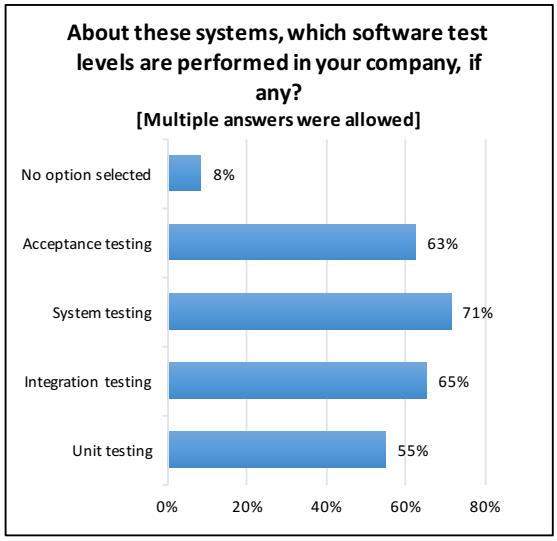

Figure 10: Test Levels DHS.

Regarding the most important features that need to be tested in DHS, the results in Figure 11 show that the feature that was considered the most important to be tested was 'Interactions between components of the system' (with 76% of responses high or very high), followed by 'Interactions between the system and the environment' (71%) and 'Multiple platforms' (66%). All the features have been considered of 'very high' or 'high' importance by a majority of respondents (50% or more).

Regarding the level of test automation for DHS, the results presented in Figure 12 show that 75% of the tests follow some automated process, however only 16% are fully automatic, which is lower than the 25% who claim to perform only manual testing.

For people who responded that there is at least some automatic process, we asked what kind of tool they use. With this question we can understand the level of effort required to automate the testing process. Looking at the results (Figure 13) we realize that only 31% use a commercial tool to automate the process, and the majority, 69%, use a tool developed in-house, reusable or not in different SUTs.

Regarding the desired features of a test automation solution for DHS, the results presented in Figure 15 show that the most important features (based in the percentage of responses high or very high) in an automated testing tool for DHS are 'Support for automatic test case execution' (75%) and 'Support for multiple platforms' (71%).

As a possible solution to test DHS, we asked the participants in this survey if they would find useful a tool to test these systems that use only a model of interactions (UML sequence diagram) as an entry model. The results (Figure 14) show that 86% consider useful a tool with these characteristics.

3.4 Multivariate Analysis

For questions specifically related to the opinion of the participants, a multivariate analysis was held with the aim to determine whether the participants' responses depend on their current function (Software testing, verification & validation versus all the others).

The results of the chi-square test for independence show that there is no statistically significant association (for a 95% significance level) between the current function (Software testing, verification & validation versus all others) and the answers to the questions shown in Figures 11, 15 or 14.

4 DISCUSSION

4.1 Relevance of Respondents

The results presented in the previous section show that this survey met the original purpose with regard to their target audience, since 70% of respondents' primary responsibility is related to 'Software testing, verification & validation'. With regard to their experience, the results showed that they are not only people who are mostly in their current position for several years, as work with software testing in general and specifically with DHS. With respect to the type of companies, the results show that this survey covers companies with diverse activity sectors and also large companies (45% have more than 1000 collaborators) which provides a great support to the conclusions reached.

Concerning the main conclusions we can draw from the results, they are next organized according to the initial research questions.

4.2 RQ1: How Relevant are DHS in the Software Testing Practice?

The results (Figure 9) show that a vast majority of approximately 90% of the companies surveyed (all with software testing activities in general) conducts tests for DHS, in at least one role and at least one test level, hence confirming the high relevance of DHS in software testing practice.

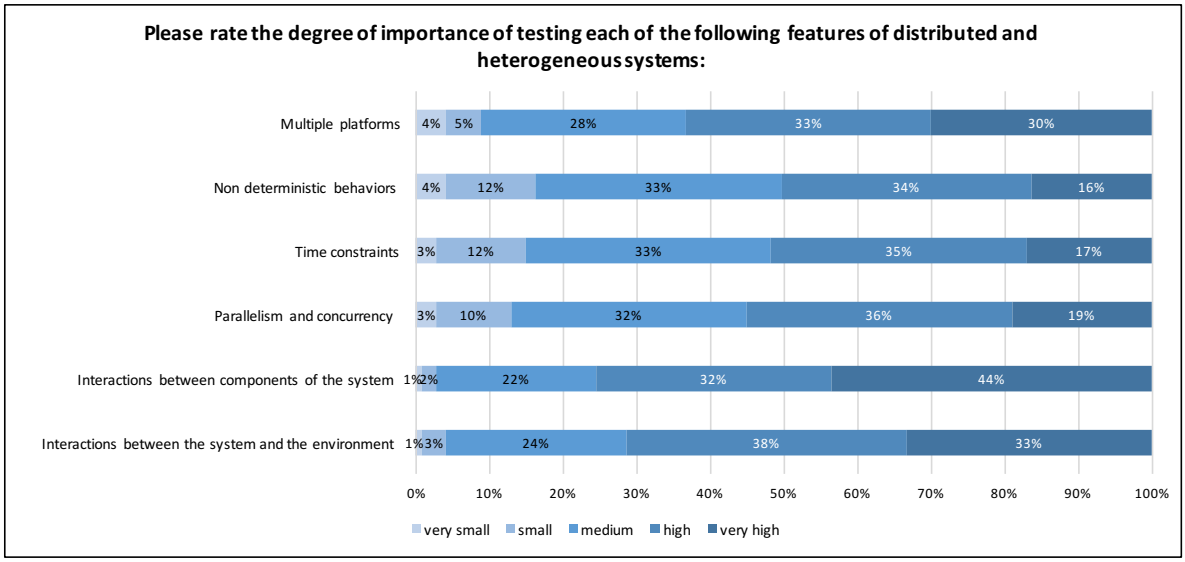

Figure 11: Features.

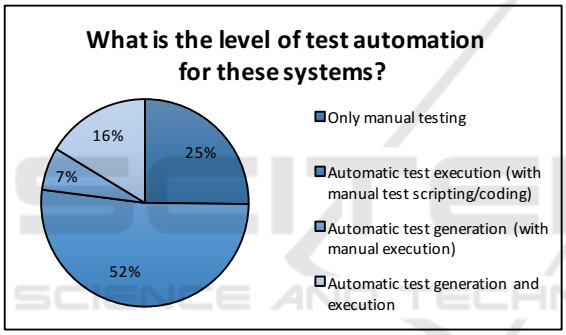

Figure 12: Automation Level.

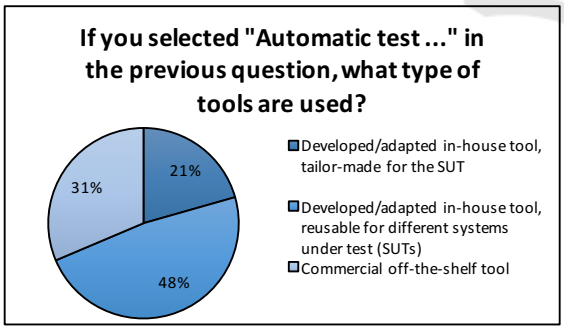

Figure 13: Automation Tool.

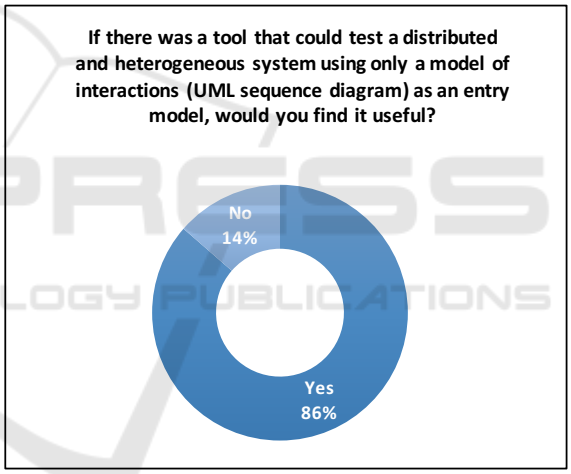

Figure 14: New Tool.

4.3 RQ2: What are the Most Important Features to be Tested in DHS?

Regarding the most important features that need to be tested in DHS, the results in Figure 11 show that the feature that was considered the most important to be tested was 'Interactions between components of the system' (with 76% of responses high or very high), followed by 'Interactions between the system and the environment' (71%) and 'Multiple platforms' (66%).

Nevertheless, all the features inquired were considered of high or very high importance by a majority of respondents (50% or more).

4.4 RQ3: What is the Current Status of Test Automation and Tool Sourcing for Testing DHS?

The results show that the current level of test automation for DHS is still very low, and there is large room for improvement, since 25% of companies in the survey claim that they only perform manual tests, against only 16% who claim to test DHS with a full automatic

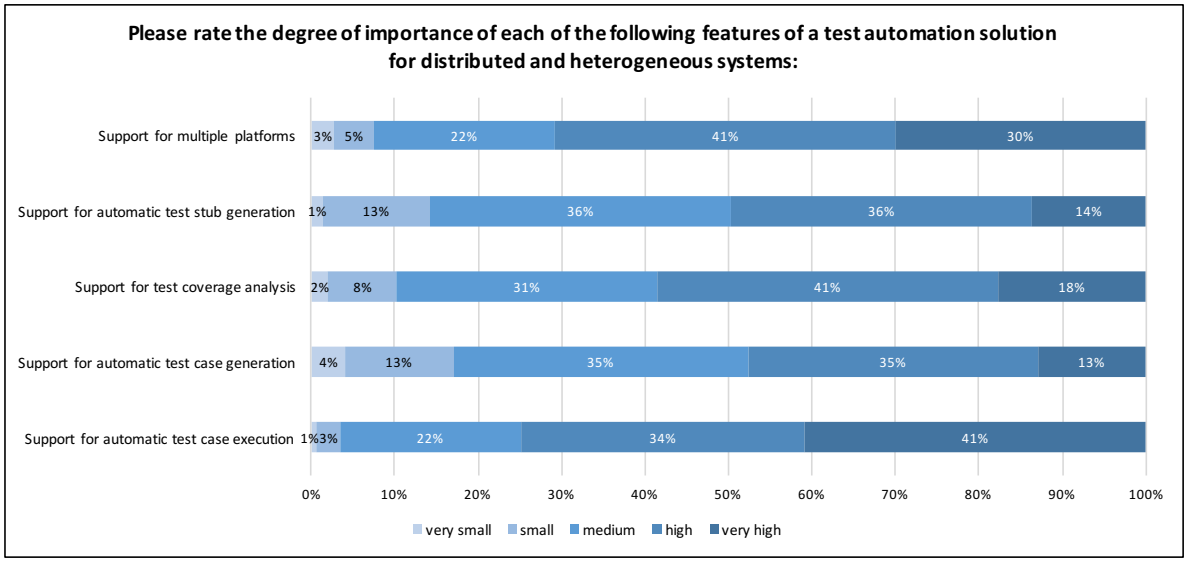

Figure 15: Tool Features.

process.

If we look for companies that have some type of automation in its testing process, we realize that the automation process is requiring a high effort in the creation / adaptation of own tools, because only 31% of companies claim to use a commercial tool to test these types of systems.

4.5 RQ4: What are the Most Desired Features in Test Automation Solutions for DHS?

Regarding the conclusions that can be drawn for future work, particularly at the level of creating tools that can reduce the effort required to test DHS, looking at Figure 15, we realize that companies identify as key aspects of a tool to test such systems the ability to automate test execution (75% of responses with high or very high importancte) and the support for multiple platforms (71%).

Nevertheless, all the features inquired were considered of medium, high or very high importance by a large majority of respondents (83% or more).

The comparison of the degree of importance attributed to automatic test case execution (96% of the responses mentioning a medium, high or very high importance in Figure 15) with the current status (78% of companies applying automatic text execution in Figure 12), show that there is a significant gap yet to be filled between the current status and the desired status of automatic test case execution.

The gap is even bigger regarding automatic test case generation, with 83% of the responses mentioning a medium, high or very high importance in Figure 15, and only 23% of the companies currently applying automatic text generation in Figure 12.

We realized even by the Figure 14, that companies are highly receptive to a test tool that has only a model of interactions as an input model for automatic test case generation and execution.

5 RELATED WORK

We only found in literature one survey (Ghazi et al., 2015) that discuss some aspects related to the testing of heterogeneous systems. The survey conducted by (Ghazi et al., 2015) explored the testing of heterogeneous systems with respect to the usage and perceived usefulness of testing techniques used for heterogeneous systems from the point of view of industry practitioners in the context of practitioners involved in heterogeneous system development reporting their experience on heterogeneous system testing. For achieving this goal the authors tried to answer two research questions:

- RQ1: Which testing techniques are used to evaluate heterogeneous systems?
- RQ2: How do practitioners perceive the identified techniques with respect to a set of outcome variables?

The authors concluded that the most frequently used technique is exploratory manual testing, followed by combinatorial and search-based testing, and that the most positively perceived technique for testing heterogeneous systems was manual exploratory

testing. Our work has a different objective of the survey conducted by Ghazi. The Ghazi main goal was to identify testing techniques, our aim is to understand how distributed systems and heterogeneous are tested in companies realizing which test levels are performed and which are the automation levels for testing these systems. The Ghazi survey also involved a much smaller number of participants (27).

As regards the general software testing in the literature there are many surveys, however as the main aim of our work is to analyze the state of practice, we analyze surveys carried out in the industry by recognized standardization bodies as ISTQB (ISTQB, 2016a). The most recent survey of this organization (ISTQB, 2016b) conducted over more than 3,000 people from 89 countries, although it has a different purpose of our work because is related to the software test in general, provides results that meet the results presented in this article, namely that there are still significant improvement opportunities in test automation (was considered in this study the area with highest improvement potential).

6 CONCLUSIONS

In order to assess the current state of the practice regarding the testing of DHS and identify opportunities and priorities for research and innovation initiatives, we conducted an exploratory survey that was responded by 147 software testing professionals that attended industry-oriented software testing conferences.

The survey allowed us to confirm the high relevance of DHS in software testing practice, confirm and prioritize the relevance of testing features characteristics of DHS, confirm the existence of a significant gap between the current and the desired status of test automation for DHS, and confirm and prioritize the relevance of test automation features for DHS. We expect that the results presented in the paper are of interest to researchers, tool vendors and service providers in this field.

As future work, we intend to develop techniques and tools to support the automatic test generation and execution of test cases for DHS, based on UML sequence diagrams.

ACKNOWLEDGEMENTS

This research work was performed in scope of the project NanoSTIMA. Project “NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016” is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

REFERENCES

AAL4ALL (2015). Ambient Assisted Living For All. http://www.aal4all.org.

Boehm, B. (2011). Some Future Software Engineering Opportunities and Challenges. In Nanz, S., editor, *The Future of Software Engineering*, pages 1–32. Springer Berlin Heidelberg.

Dias Neto, A. C., Subramanyan, R., Vieira, M., and Travassos, G. H. (2007). A Survey on Model-based Testing Approaches: A Systematic Review. In *Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies: Held in Conjunction with the 22Nd IEEE/ACM International Conference on Automated Software Engineering (ASE) 2007*, WEASELTech ’07, pages 31–36, New York, NY, USA. ACM.

DoD (2008). Systems Engineering Guide for Systems of Systems. Technical report, Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering Version 1.0.

Faria, J. and Paiva, A. (2014). A toolset for conformance testing against UML sequence diagrams based on event-driven colored Petri nets. *International Journal on Software Tools for Technology Transfer*, pages 1–20.

Ghazi, A. N., Petersen, K., and Börstler, J. (2015). *Software Quality. Software and Systems Quality in Distributed and Mobile Environments: 7th International Conference, SWQD 2015, Vienna, Austria, January 20-23, 2015, Proceedings*, chapter Heterogeneous Systems Testing Techniques: An Exploratory Survey, pages 67–85. Springer International Publishing, Cham.

Hierons, R. M. (2014). Combining Centralised and Distributed Testing. *ACM Trans. Softw. Eng. Methodol.*, 24(1):5:1–5:29.

ISTQB (2016a). International Software Testing Qualifications Board. http://www.istqb.org/.

ISTQB (2016b). ISTQB Worldwide Software Testing Practices Report 2015-2016. Technical report.

Kim, B., Hwang, H. I., Park, T., Son, S. H., and Lee, I. (2014). A layered approach for testing timing in the model-based implementation. In *DATE*, pages 1–4. IEEE.

Lima, B. and Faria, J. P. (2016). *Software Technologies: 10th International Joint Conference, ICSOFT 2015, Colmar, France, July 20-22, 2015, Revised Selected Papers*, chapter Automated Testing of Distributed and Heterogeneous Systems Based on UML Sequence Diagrams, pages 380–396. Springer International Publishing, Cham.

OMG (2015). OMG Unified Modeling LanguageTM (OMG UML) Version 2.5, Superstructure. Technical report, Object Management Group.

Petrenko, A. and Yevtushenko, N. (2011). *Testing Software and Systems: 23rd IFIP WG 6.1 International Conference, ICTSS 2011, Paris, France, November 7-10, 2011. Proceedings*, chapter Adaptive Testing of Deterministic Implementations Specified by Nondeterministic FSMs, pages 162–178. Springer Berlin Heidelberg, Berlin, Heidelberg.

Tassey, G. (2002). The Economic Impacts of Inadequate Infrastructure for Software Testing. Technical report, National Institute of Standards and Technology.

Torens, C. and Ebrecht, L. (2010). RemoteTest: A Framework for Testing Distributed Systems. In *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, pages 441–446.

Utting, M. and Legeard, B. (2007). *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

West, S., Nanz, S., and Meyer, B. (2012). Demonic testing of concurrent programs. In *Proceedings of the 14th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering*, ICFEM'12, pages 478–493, Berlin, Heidelberg. Springer-Verlag.

Wohlin, C., Höst, M., and Henningsson, K. (2003). Empirical research methods in software engineering. In *Empirical methods and studies in software engineering*, pages 7–23. Springer.