Citation for the published Conference paper:

Title:
Heterogeneous Systems Testing Techniques: An Exploratory Survey

Author:
Ahmad Nauman Ghazi, Kai Petersen, Jürgen Börstler

Conference Name:
7th Software Quality Days (SWQD)

Conference Year:
2015

Conference Location:
Vienna

Heterogeneous Systems Testing Techniques: An Exploratory Survey

Ahmad Nauman Ghazi, Kai Petersen, and Jürgen Börstler

Blekinge Institute of Technology, Karlskrona, Sweden
nauman.ghazi@bth.se, kai.petersen@bth.se, jurgen.borstler@bth.se

Abstract Heterogeneous systems comprising sets of inherent subsystems are challenging to integrate. In particular, testing for interoperability and conformance is a challenge. Furthermore, the complexities of such systems amplify traditional testing challenges. We explore (1) which techniques are frequently discussed in literature in context of heterogeneous system testing that practitioners use to test their heterogeneous systems; (2) the perception of the practitioners on the usefulness of the techniques with respect to a defined set of outcome variables. For that, we conducted an exploratory survey. A total of 27 complete survey answers have been received. Search-based testing has been used by 14 out of 27 respondents, indicating the practical relevance of the approach for testing heterogeneous systems, which itself is relatively new and has only recently been studied extensively. The most frequently used technique is exploratory manual testing, followed by combinatorial testing. With respect to the perceived performance of the testing techniques, the practitioners were undecided regarding many of the studied variables. Manual exploratory testing received very positive ratings across outcome variables.

1 Introduction

Over the years, software has evolved from simple applications to large and complex system of systems [8]. A system of systems consists of a set of individual systems that together form a new system. The system of systems could contain hardware as well as software systems. Recently, system of systems has emerged as a highly relevant topic of interest in the software engineering research community investigating its implications for the whole development life cycle. For instance, in the context of system of systems, Lane [20] studied the impact on development effort, Ali et al. [2] investigated testing, and Lewis et al. [22] proposed a process of how to conduct requirements engineering.

Systems of systems often exhibit heterogeneity [21], for instance, in implementation, hardware, process and verification. For the purpose of this study, we define a *heterogeneous system* as a system comprised of multiple systems (system of systems) where at least one subsystem exhibits heterogeneity with respect to the other systems [13]. Heterogeneity may occur in terms of different system complexities [33], programming languages and platforms [44], types of systems [10], system supply (internal vs. third party) [19], development processes and distribution among development sites [15]. However, no frameworks are available as yet to measure these heterogeneity dimensions. The system of systems approach taken in development of heterogeneous systems give rise to various challenges due to continuous change in configurations and multiple interactions between the functionally independent subsystems. The challenges posed to testing of heterogeneous systems are mainly related to interoperability [28, 44], conformance [28] and large regression test suites [2, 6]. Furthermore, the inherent complexities of heterogeneous systems also pose challenges to the specification, selection and execution of tests.

In recent years, together with the emergence of system of systems research testing of heterogeneous systems received an increased attention from the research community. However, solutions proposed have been primarily evaluated from the academic perspective, and not the viewpoint of the practitioner.

In this study, we explored the viewpoint of practitioners with respect to testing heterogeneous systems. Two main contributions are made:

- C1: *Explore which testing techniques investigated in research are used by practitioners.* Thereby, we learn which techniques practitioners are aware of, and which ones are most accepted.
- C2: *Explore the perception of the practitioners of how well the used techniques perform with respect to a specified and frequently studied set of outcome variables.* Understanding the practitioners' perception of the techniques relative to each other allows to identify preferences from the practitioners' viewpoint. The findings will provide interesting pointers for future work to understand the reasons for the findings, and improve the techniques accordingly.

The contributions are made by using an exploratory survey to capture the opinion of practitioners.

The remainder of the paper is structured as follows: Section 2 presents the related work. Section 3 outlines the research method, followed by the results in Section 4. Section 5 presents a discussion of observations from the results. Finally, in Section 6, we conclude this study.

2 Related work

The related work focuses on testing of heterogeneous systems, first discussing testing of heterogeneous systems as such, followed by reviewing solutions of how to test them. However, no surveys could be found that discuss any aspect of testing of heterogeneous systems.

2.1 Testing in Heterogeneous Systems

Testing heterogeneous systems is primarily considered to be a challenge emanating from the problem of integration and system-level testing [9] [41]. Therefore, the current research in the area of heterogeneous systems considers it as a subsystem interaction issue [41]. It is also observed that solving the inherent complexities underlying the testing heterogeneous systems is not a priority. Therefore, most of the related research is focused on addressing the accidental complexities in testing of heterogeneous systems by tuning and optimizing different testing techniques and methods.

A number of research studies discuss system-level testing in general terms without addressing specific test objectives. For automated functional testing, Donini et al. [9] propose a test framework where functional testing is conducted in an external simulated environment based on service-oriented architectures. The researchers demonstrated that functional system testing through simulated environments can be an approach to overcome the challenge of minimizing test sets. The test sets identified were representative of the real operational usage profile for the system. Wang et al. [41] study heterogeneous systems that exhibit heterogeneity at the platform level and discussed different factors considered in system-level testing of heterogeneous systems. Other than the studies focusing on system and integration testing, a relatively small set of studies attempt to discuss the problem of testing in heterogeneous systems in other test phases. Mao et al. [23] study this problem in the unit test phase whereas Diaz [7] addresses the problem of testing heterogeneous systems in the acceptance testing phase.

Research literature related to testing of heterogeneous systems frequently discusses the interoperability as a common issue. Interoperability testing is also a key test objective in different applications and technology domains. Xia et al. [44] address the interoperability problem in the web service domain and propose a test method to automate conformance and interoperability testing for e-business specification languages. Narita et al. [28] propose a method supported by a testing framework for interoperability testing for web service domain focusing on communication in robotics domain. However, interoperability remains a challenge in other domains as well. In context of large scale component based systems, Piel et al. [35] present a virtual component testing technique and demonstrated how virtual components can be formed using three different algorithms. This technique was further implemented and evaluated in industrial settings. Furthermore, Kindrick et al. [18] propose a technique combining interoperability testing with conformance testing and conclude that combining the two techniques will reduce the cost of setting up and executing the test management processes improving the effectiveness.

2.2 Testing Techniques

We surveyed techniques on testing heterogeneous systems that have been reported to be used in literature or industry for that purpose. These have been identified through a systematic literature review and a case study [12] and represent three coarse categories of testing techniques, namely manual exploratory, combinatorial, and search-based testing.

Manual Exploratory testing: Manual exploratory testing (ET) is an approach to test software without pre-defined test cases in contrast with traditional test case based testing. The main characteristics of exploratory testing are simultaneous learning, test design and execution [16, 40]. The tester has the freedom to dynamically design, modify and execute the tests.

In past, exploratory testing was seen as an ad-hoc approach to test software. However, over the years, ET has evolved into a more manageable and structured approach without compromising the freedom of testers to explore, learn and execute the tests in parallel. An empirical study comparing the effectiveness of exploratory testing with test-case based testing was conducted by Bhatti and Ghazi [4] and further extended (cf. [1]). This empirical work concludes that ET produces more defects as compared to test case based testing where time to test is a constraint.

Combinatorial Testing: Combinatorial testing is used to test applications for different test objectives at multiple levels. A comprehensive survey and discussion is provided by Nie and Leung [29]. It has been used for both unit and system-level testing in various domains. Combinatorial testing tends to reduce the effort and cost for effective test generation [5]. There exist a number of variants of combinatorial testing, which are used in different domains to test heterogeneous systems.

The problem of testing web services is the most common area in heterogeneous systems that is addressed in literature using different test techniques as discussed in Section 2.1. Mao et al. [23] and Apilli [3] proposed different frameworks for combinatorial testing to test component based software systems in a web services domain.

Wang et al. [42] study the problem of how interaction faults can be located based on combinatorial testing rather than manual detection and propose a technique for interactive adaptive fault location. Results from this study show that the proposed technique performs better than the existing adaptive fault location techniques.

Changing configurations pose challenges to combinatorial testing techniques. To that end Cohen et al. [6] conducted an empirical study to quantify the effectiveness of test suites. The study shows that there is an exponential growth of test cases when configurations change and subsets of test suites are used, similar to what is common in regression testing.

Mirarab et al. [27] conducted an industrial case study and propose a set of techniques for requirement-based testing. The SUT was software for a range of wireless, mobile devices. They propose a technique to model requirements, a technique for automated generation of tests using combination strategies, and a technique for prioritization of existing test cases for regression testing.

Search-Based Software Testing: In search-based testing meta-heuristics are used to solve software testing problems by using, for example genetic algorithms, to search for a solution for a problem (e.g. to generate test data).

Marin et al. [24] present an integrated approach where search-based techniques are applied on top of more classical techniques to derive optimal test configurations for web applications. The authors describe state of art and future web applications as complex and distributed, exhibiting several dimensions of heterogeneity. The study describes an approach that integrates combinatorial testing, concurrency testing, oracle learning, coverage analysis, and regression testing with search-based testing to generate test cases.

Shiba et al. [38], proposed two artificial life algorithms to generate minimal test sets for t-way combinatorial testing based on a genetic algorithm (GA) and an ant colony algorithm (ACA). Experimental results show that when compared to existing algorithms including AETG (Automatic Efficient Test Generator) [5], simulated annealing-based algorithm (SA) and in-parameter order algorithm (IPO), this technique works effectively in terms of size of test set as well as time to execute.

Another study by Pan et al. [31] explores search-based techniques and defines a novel algorithm, i.e., OEPST (organizational evolutionary particle swarm technique), to generate test cases for combinatorial testing. This algorithm combines the characteristics of organizational evolutionary idea and particle swarm optimization algorithm. The experimental results of this study show that using this new algorithm can reduce the number of test cases significantly.

There are refinements of exploratory, combinatorial, and search-based testing. However, these have not been surveyed to keep the questionnaire at a manageable length in order to avoid dropouts. Manual exploratory testing is a manual testing technique where the tester simultaneously learns, designs, and executes tests. The thought process (e.g. whether a specific technique inspires the test design) is not prescribed. Both combinatorial and search-based testing are usually supported by tools and automated, while they have different approaches in solving the testing problem (see above).

3 Research method

The survey method used in this study is an exploratory survey. Thörn [39] distinguishes statistical and exploratory surveys.

In exploratory surveys the goal is not to draw general conclusion about a population through statistical inference based on a representative sample. A representative sample (even for a local survey) has been considered challenging, the author [39] points out that: *"This [remark by the authors: a representative sample] would have been practically impossible, since it is not feasible to characterize all of the variables and properties of all the organizations in order to make a representative sample."* Similar observations and limitations of statistical inference have been discussed by Miller [26].

Given that the focus of this research is specific to heterogeneous systems, the population is limited. We were aware of specific companies and practitioners that work with such systems, but the characteristics of companies and their products were not available to us. Hence, an exploratory survey was conducted to answer our research questions. Though, aim was to gather data from companies with different characteristics; different domains, sizes, etc. represented; for the obtained answers, external validity is discussed in Section 3.5.

3.1 Study purpose

The goal of the survey is formulated based on the template suggested in [43] to define the goals of empirical studies. The goal for this survey is to explore *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.*

In relation to the research goal two main research questions (RQs) were asked:

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?*

3.2 Survey Distribution and Sample

We used convenience sampling to obtain the answers. Of interest were practitioners that were involved in the testing of heterogeneous systems before, thus not every software tester would be a suitable candidate for answering the survey. The sample was obtained through personal contacts as well as postings in software engineering web communities (e.g. LinkedIn and Yahoo Groups). 100 personal contacts were asked to respond, and to distribute the survey later. Furthermore, we posted the survey on 32 communities.

Overall, we obtained 42 answers, of which 27 were complete and valid. One answer was invalid as each response was given as "others", without any further specification. The remaining respondents did not complete the survey. We provide further details on the respondents and their organizations in Section 4.1.

3.3 Instrument Design

The survey instrument is structured along the following themes[1].

- *Respondents:* In this theme information about the respondent is collected. This information is comprised of: current position; duration of working in the current position in years; duration of working with software development; duration of working with testing heterogeneous systems.
- *Company, processes, and systems:* This theme focuses on the respondents' organizations and the characteristics of the products.
- *Test coverage:* Here the practitioners rate the importance of different coverage criteria on a 5-point Likert scale from *"Very Important"* to *"Unimportant"*. The coverage criteria rated were specification-based, code-based, fault-based, and usage-based.

[1] The survey can be found at https://www.surveymonkey.com/s/RP6DQKF

- *Usage of testing techniques:* We identified three categories of testing techniques through our ongoing systematic literature review that have been attributed and used in testing heterogeneous systems, namely search-based, combinatorial, and manual exploratory testing (see also Section 2). The concepts of the testing techniques were defined in the survey to avoid any confusion. Two aspects have been captured, usage and evaluation. With respect to usage we asked for the frequency of using the different techniques on a 7-point Likert scale ranking from *"Always"* to *"Never"*. We also provided the option *"Do not know the technique"*.
- *Usefulness of testing techniques:* Each technique has been rated according to its usefulness with respect to a set of outcome variables that are frequently studied in literature on quality assurance techniques. The usefulness for each technique for each variable was rated on a 5-point Likert scale from *"Strongly Agree"* to *"Strongly Disagree"*. Table 1 provides an overview of the studied variables and their definitions.
- *Contact details:* We asked the respondents for their company name and e-mail address. The answer to this question was optional in case the respondents wished to stay anonymous towards the researchers.

Table 1. Surveyed Variables

Variable	References
Ease of use	[17] [34]
Effectiveness in detecting critical defects	[1]
Number of false positives	[1]
Effectiveness in detecting various types of defects	[1]
Time and cost efficiency	[1] [34]
Effectiveness in detecting interoperability issues	[32]
Effectiveness for very large regression test sets	[14]
External product quality	[30]

The design of the survey has been pretested by three external practitioners and one researcher. The feedback led to minor reformulation and changes in the terminology used to become clear for practitioners. Furthermore, the number of response variables has been reduced to make the survey manageable in time and avoid maturation. Furthermore, the definition of heterogeneous system was revised to be more understandable. We further measured the time the respondents needed in the pretest to complete the survey. The time was between 10 and 15 minutes.

3.4 Analysis

For reflection on the data (not for inference) we utilized statistical tests to highlight differences for the techniques surveyed across the outcome variables. The Friedman test [11] (non-parametric test) has been chosen given multiple variables (treatments) were studied, the data being on ordinal scale.

3.5 Validity Threats

Internal Validity One threat to capturing truthfully is if the questions asked in the survey are misunderstood. To reduce this threat we pretested the survey and made updates based on the feedback received. Another threat is maturation where the behavior changes over time. This threat has been reduced by designing the survey so that no more than 15 minutes were necessary to answer the survey.

Construct Validity Theoretical validity is concerned with not being able to capture what we intend to capture (in this case the usefulness of different techniques across different outcome variables). To reduce this threat we defined variables based on literature, in particular focusing on variables that are frequently studied when evaluating quality assurance approaches. Given that the study is based on the subjects' experience, the lack of experience in search-based testing limits the comparability, given that eight respondents did not know the technique, and five have never used it. However, the remaining respondents had experience using it. For the other techniques (manual exploratory testing and combinatorial testing) only few respondents did not know them, or lacked experience. Given that the aim of the study is not to generalize the findings through inference, but rather identify interesting patterns and observations in an exploratory way, threats related to statistical inference were not emphasized.

External Validity The exploratory nature of the survey does not allow to statistically generalize to a population. However, as suggested by [39], interesting qualitative arguments can be made such studies. The context captured in the demographics of the survey limits the external generalizability. In particular, the majority of respondents were related to the consulting industry (35.7%), followed by computer industry (28.6%), and communications (25.0%), other industries only have very few responses and are not represented in this study (e.g. accounting, advertising, etc.). With regard to company size, all four size categories are equally well represented. With regard to development models, agile and hybrid processes have the highest representation. The data is hence not relevant for the other models. Overall, the external validity could be strengthened by a higher number of answers. Though, given that the survey was focused on heterogeneous systems the possible sample was reduced. In comparison, with a similar strategy of distributing a survey on a wider topic (automated software testing) over 100 valid responses could be obtained [36].

Conclusion Validity Interpretive validity is primarily concerned with conclusions based on statistical analysis, and researcher bias when drawing conclusions. Given that the involved researchers have no particular preference on any of the solutions surveyed based on previous research, this threat can be considered as being under control.

4 Results

We first describe the study context as this allows companies to compare their own context, and hence being able to determine to what degree the results are relevant for them. Thereafter, we characterize the heterogeneity dimensions of the systems being reported by the practitioners. Thereafter, the answers to the research questions are presented.

4.1 Context

Subjects Table 2 provides an overview of the primary roles of the subjects participating in the survey. The roles most frequently presented are directly related with either quality assurance, or the construction and design of the system. Overall, the experience in years in the current role indicates a fair to strong experience level of the respondents in their current positions.

Table 2. Roles of Subjects

Responsibility	Percent	Responses
Software developer (implementation, coding etc.)	22,2%	6
Software architect (software structure, architecture, and design)	18,5%	5
Software verification & validation (testing, inspection, reviews etc.)	18,5%	5
Software quality assurance (quality control, quality management etc.)	14,8%	4
Other	11,1%	3
System analyst (requirements elicitation, analysis, specification and validation etc.)	7,4%	2
Project manager (project planning, project measurement etc.)	3,7%	1
Product manager (planning, forecasting, and marketing software products etc.)	0,0%	0
Software process engineer (process implementation and change, process and product measurement etc.)	0,0%	0

Looking at the overall experience related to software engineering in years, the average experience is 10.55 years with a standard deviation of 7.04. This indicates that the overall experience in software development is very high.

We also asked for the experience of the practitioners in testing heterogeneous systems themselves. The average experience in testing heterogeneous systems is 4.63 years with a standard deviation of 5.22, while 8 respondents did not have experience as testers on heterogeneous systems themselves. The survey focused on practitioners involved in developing heterogeneous systems though, as those also often gain insights on the quality assurance processes (e.g. people in quality management). Hence, those responses were not excluded.

Company, processes, and systems The number of responses in relation to company size are shown in Table 3. All sizes are represented well by the respondents, hence the results are not biased towards a specific company size.

The companies surveyed worked in 24 different industry sectors (one company can work in several sectors, hence multiple answers were possible). The industries that were represented by the highest number of respondents were consulting (9 respondents), computer industry (hardware and desktop software) (7 respondents), communications (6 respondents), and business/professional services (5 respondents).

Table 3. Company Size (Number of Employees)

Size (no. of employees)	Percent	Responses
Less than 50	18.5	5
50 to 249	29.6	8
250 to 4499	29.6	8
5400 and more	22.2	6

The systems developed are characterized by different types as specified in [10]. As shown in Table 4 the clear majority of respondents was involved in data-dominant software development, though all types were represented through the surveyed practitioners.

Table 4. System Types

System type	Percent	Responses
Data-dominant software	63.0	17
Control-dominant software	25.9	7
Computation-dominant software	25.9	7
Systems software	22.2	6
Other	14.8	4

The development models used in the surveyed companies are illustrated in Table 5. The clear majority of respondents is working with agile development and hybrid processes that are dominated by agile practices.

Table 5. Development Models

Model	Percent	Responses
Agile	29,6	8
Hybrid process (dominated by agile practices, with few plan-driven practices)	29,6	8
Waterfall	11.1	3
V-Model	11.1	3
Hybrid process (dominated by plan-driven practices, with few agile practices)	11.1	3
Other	7.4	2
Spiral	3.7	1

Test coverage A key aspect of testing is the test objectives that drive the selection of test cases (cf. [25]). We captured the objectives of the participating industry practitioners in their test case selection as shown in Fig. 1. Specification-based coverage is clearly the most important criterion for the studied companies, followed by fault-based coverage. Overall, all coverage objectives are considered important by at least half of the participants.

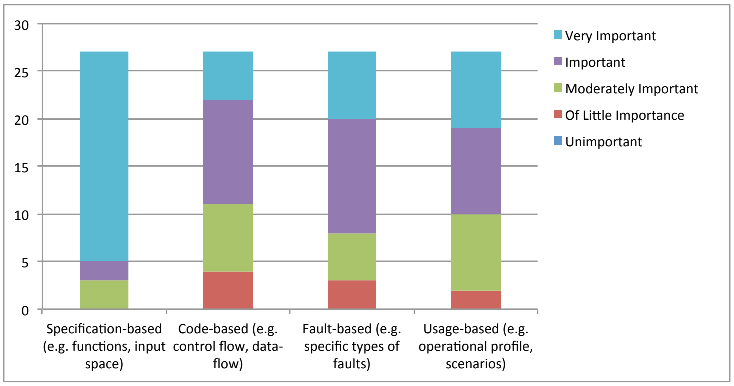

Figure 1. Importance of test objectives

4.2 Heterogeneity of Systems

There exist no agreement in literature on the definition of heterogeneity in systems. However, individual papers defined different dimensions of heterogeneity. In the survey, we asked the participants which dimensions are applicable to them, as illustrated in Table 6. The table shows that systems of different complexity, platforms and programming languages, and types of systems were the most common dimensions.

Table 6. Heterogeneity Dimensions in the Studied Systems

Heterogeneity Dimensions	Percent	Responses
System complexity	70.4	19
Programming language and platforms	59.2	16
Type of systems	55.56	15
System supply (internally developed and third party)	48.1	13
Development processes	40.7	11
Distribution of development systems in different locations	25.9	7

The respondents could select multiple heterogeneity items, as several may apply to their development context. For at least half of the systems the respondents selected three or more heterogeneity dimensions that apply to them. A quarter of all systems surveyed is characterized by four or more dimensions (see Fig. 2). Only few systems are only characterized by one of the dimensions.

4.3 RQ1: Usage of Testing Techniques

We captured the frequency of usage for the three different techniques introduced earlier (search-based, manual exploratory, and combinatorial testing). The frequencies are illustrated in Fig. 3.

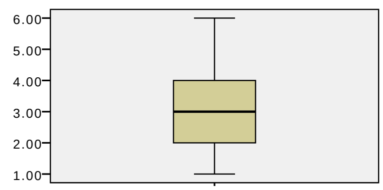

Figure 2. Number of Heterogeneity Dimensions selected

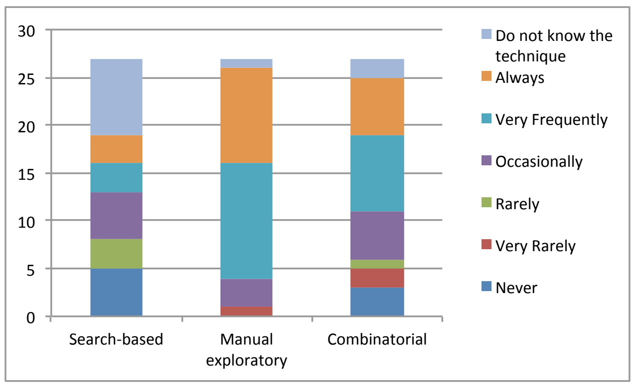

Figure 3. Usage of Techniques in Heterogeneous Systems

Looking at the overall distribution of usage, it is clearly visible that manual exploratory testing is the most frequently used technique, followed by combinatorial testing and search-based testing. There was not a single respondent indicating of never having used manual exploratory testing.

Search-based testing is the least-used technique, as well as the technique that is least-known. However, 3 respondents who mentioned that they always use search-based testing are all test consultants. Another consultant mentioned frequent usage of the technique along with 2 more respondents who are in education and professional services industries, respectively. Only very few respondents are not aware of manual exploratory and combinatorial testing, while the usage appears to depend on the role of the respondent.

4.4 RQ2: Perceived Usefulness

Fig. 4 provides the rating of the variables for the three different techniques studied. To highlight patterns in the data, we also used statistical testing as discussed in Section 3.4. The results of the test are shown in Table 7.

The highest undecided rates are observed for search-based testing. This can be explained by the observation that people were not aware of the technique, or never used it (see Fig. 3). Also, a relatively high undecided rate can be seen for combinatorial testing, however, this cannot be attributed to the lack of knowledge about the technique, or that practitioners never used it, as the numbers

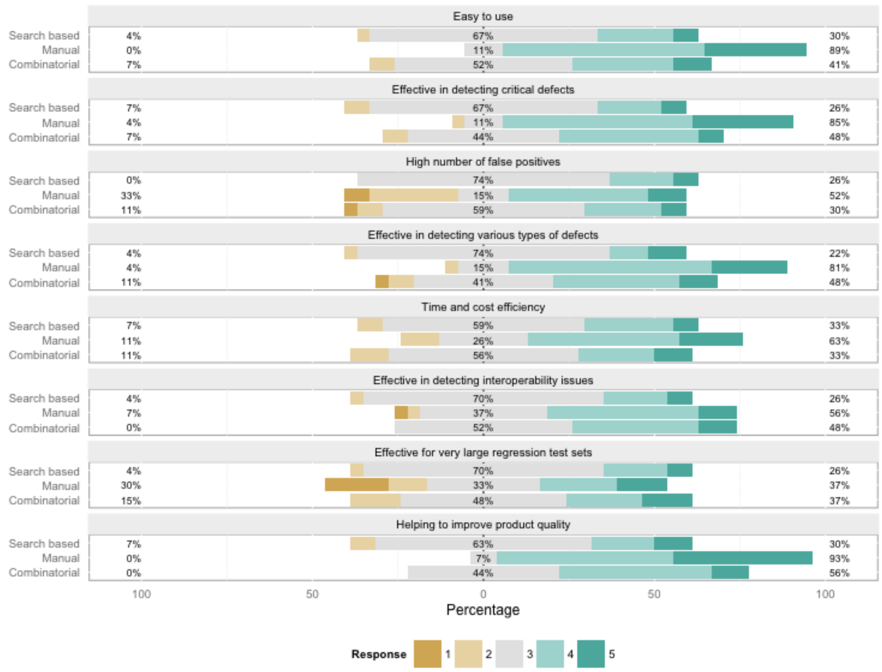

Figure 4. Practitioners' Perceptions of Testing Techniques for Heterogeneous Systems (1 = Strongly Disagree, 2 = Disagree, 3 = Uncertain, 4 = Agree, 5 = Strongly Agree)

on both items were relatively low. The opposite is true for manual exploratory testing, where only very few practitioners were undecided.

Variables that are more unique and emphasized for heterogeneous systems (effectiveness in detecting interoperability issues and effectiveness for very large regression test sets) have higher undecided rates for all the techniques. That is, there is a high level of uncertainty across techniques. In the case of regression tests manual exploratory testing was perceived as the most ineffective. For interoperability testing no major difference between the ratings can be observed, which is also indicated by the statistical tests shown in Table 7.

Of all techniques, manual exploratory testing is rated exceptionally high in comparison to other techniques for ease of use, effectiveness in detecting critical defects, detecting various types of defects, and in improving product quality. The high rating is also highlighted through the statistical tests, which detected this as a difference in the data sets (see Table 7). At the same time, it also received the strongest negative ratings, which was the case for false positives and effectiveness for very large regression test suites.

Table 7. Friedman test statistics

Item	N	χ^2	df	p-value
Easy to use	27	22.522	2	0.000
Effective in detecting critical defects	27	19.500	2	0.000
High number of false positives	27	0.090	2	0.956
Effective in detecting various types of defects	27	17.848	2	0.000
Time and cost efficiency	27	3.797	2	0.150
Effective in detecting interoperability issues	27	7.000	2	0.030
Effective for very large regression test sets	27	1.509	2	0.470
Helping to improve product quality	27	25.400	2	0.000

5 Discussion

Based on the data collected, we highlight interesting observations, and present their implications.

Observation 1: Interestingly, search-based testing was applied by several practitioners in the scoped application of testing heterogeneous systems (in total 14 of 27 used it at least very rarely), even though in comparison it was the least frequently applied technique. Literature surveying research on search-based testing reported acknowledges that testing is primarily a manual process [25]. Also, in heterogeneous systems we only identified few studies in our search for literature that used search-based testing. Hence, it is an interesting observation that companies are using search-based testing. At the same time, many practitioners were not aware of it at all. This leads to the following lessons learned:

Lessons learned: First, given the presence of search-based testing in industry, there exist opportunities for researchers to study it in real industrial environments and to collect experiences made by practitioners; Second, practical relevance of search-based testing in heterogeneous testing is indicated by the adoption of the technique, which is encouraging for this relatively new field.

Observation 2: Although, the survey was targeted towards a specific group of practitioners that have experience with developing and testing heterogeneous systems, the practitioners were largely undecided on whether the techniques used are suitable for detecting interoperability issues. Fig. 4 shows that search-based testing has comparatively high undecided rates for all the variables.

Lessons learned: Practitioners require further decision support and comparisons to be able to make informed decisions about the techniques given the high level of uncertainty. In particular, further comparative studies (which were lacking) are needed in general, and for heterogeneous systems in particular. If people are undecided, adoption is also hindered; hence one should aim to reduce the uncertainty on outcomes for the variables studied.

Observation 3: Manual exploratory testing is perceived as very positive by practitioners for the variables “Ease of use”, “Effective in detecting critical defects”, “Effective in detecting various types of defects”, "Time and cost effective” and “Helping to improve product quality”. On the other hand, it has been perceived poorly in comparison to other techniques for the variables “High number of false positives” and “Effective for very large regression-test suites”. Given the context of testing heterogeneous systems, these observations are interesting to compare with findings of studies investigating exploratory testing. Shah et al. [37] investigated exploratory testing and contrasted the benefits and advantages of exploratory and scripted testing through the application of a systematic review combined with expert interviews. Their review is hence used as a basis for the comparison with literature.

The finding with respect to ease of use was understandable, but could also be seen as a paradox. On the one hand there are no perceived barriers as one does not have to learn testing techniques; however, the quality of tests is not known because there is such a high dependency on the skills of the testers (cf. Shah et al. [37]), which could potentially lead to a wrong perception. Shah et al. identified multiple studies indicating time and cost efficiency, and also confirmed that the exploratory testing is good at identifying the most critical defects. Overall, this appears to be well in-line with the findings for heterogeneous systems. With respect to false positives, the practitioners were in disagreement on whether manual exploratory testing leads to a high number of false positives. Literature on the other hand suggests that fewer false positives are found. With respect to regression testing, the findings indicate the potential for better regression testing in case that sessions are properly recorded, but it was also recognized that it is difficult to prioritize and reevaluate the tests.

Lessons learned: Even though not representative, the data indicates a gap between industry focus and research focus. Therefore, research should focus on investigating exploratory testing, how it should be applied, and how efficient it is in capturing interoperability issues to support companies in improving their exploratory testing practices.

6 Conclusion

In this study we explored the testing of heterogeneous systems. In particular, we studied the usage and perceived usefulness of testing techniques for heterogeneous systems. The techniques were identified based on an ongoing systematic literature review. The practitioners surveyed were involved in the development of heterogeneous systems. Two main research questions were answered:

RQ1: Which testing techniques are used to assess heterogeneous systems? The most frequently used technique is exploratory manual testing, followed by combinatorial and search-based testing. As discussed earlier, it is encouraging for the field of search-based testing that a high number of practitioners have made experiences with search-based testing. This may provide opportunities to study the technique from the practitioners' perspective more in the future. Looking at the awareness, the practitioners were well aware of manual exploratory and combinatorial testing, however, a relatively high number was not aware of what search-based testing is.

RQ2: How do practitioners perceive the identified techniques with respect to a set of outcome variables? The most positively perceived technique for testing heterogeneous systems was manual exploratory testing, which was the highest rated in five (ease of use, effectiveness in detecting critical defects, effective in detecting various types of defects, time and cost efficiency, helping to improve product quality) out of eight studied variables. While manual exploratory testing was the most used technique in the studied companies, it is the least investigated technique in the literature on testing heterogeneous systems.

In future work, based on the results of the study, several important directions of research were made explicit:

- Given there are no frameworks available that can help identify, to what degrees one system is heterogeneous in comparison to other systems. Therefore, a framework will be provided to measure different dimensions of heterogeneity.
- In order to reduce the uncertainty with respect to the performance of the techniques comparative studies are needed. In particular, in the context of heterogeneous systems variables more relevant to that context should be studied (interoperability, large regression test suits). However, in general more comparative studies may be needed, for instance by comparing their performance on heterogeneous open source systems (e.g. Linux).
- Given the positive indications of the adoption of search-based in the industry, the focus should also be on understanding how and with what success search-based is used in the industry for heterogeneous and other systems.
- Interesting patterns identified and highlighted in the discussion should be investigated in further depth, two examples should be highlighted: First, does (and if so how) heterogeneity affect the performance of exploratory testing in terms of false positives reported? Second, how could it be explained that manual exploratory testing is so positively perceived? Possible propositions are there is a low perceived entry level of using the technique, while it is at the same time very hard to master given its dependence on the testers' skills. Furthermore, interestingly it was perceived as being time- and cost efficient, which should be understood further. Overall, large and complex systems have many interactions that could require automation to be able to achieve a satisfactory level of coverage.

References

[1] W. Afzal, A. N. Ghazi, J. Itkonen, R. Torkar, A. Andrews, and K. Bhatti. An experiment on the effectiveness and efficiency of exploratory testing. *Empirical Software Engineering*, pages 1–35, 2014.

[2] N. B. Ali, K. Petersen, and M. Mäntylä. Testing highly complex system of systems: An industrial case study. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM 2012)*, pages 211–220. ACM, 2012.

[3] B. S. Apilli. Fault-Based Combinatorial Testing of Web Services. *Companion to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2009, October 25-29, 2009, Orlando, Florida, USA*, pages 731–732, 2009.

[4] K. Bhatti and A. N. Ghazi. Effectiveness of exploratory testing: An empirical scrutiny of the challenges and factors affecting the defect detection efficiency. Master's thesis, Blekinge Institute of Technology, 2010.

[5] D. Cohen, S. Dalal, M. Fredman, and G. Patton. The aetg system: an approach to testing based on combinatorial design. *Software Engineering, IEEE Transactions on*, 23(7):437 –444, jul 1997.

[6] M. B. Cohen, J. Snyder, and G. Rothermel. Testing Across Configurations : Implications for Combinatorial Testing. *Software Engineering Notes*, 31(6):1–9, 2006.

[7] J. Diaz, A. Yague, P. P. Alarcon, and J. Garbajosa. A Generic Gateway for Testing Heterogeneous Components in Acceptance Testing Tools. *Seventh International Conference on Composition-Based Software Systems (ICCBSS 2008)*, pages 110–119, Feb. 2008.

[8] DoD. Systems and software engineering. systems engineering guide for systems of systems, version 1.0. Technical Report ODUSD(A&T)SSE, Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Washington, DC, USA, 2008.

[9] R. Donini, S. Marrone, N. Mazzocca, A. Orazzo, D. Papa, and S. Venticinque. Testing Complex Safety-Critical Systems in SOA Context. *2008 International Conference on Complex, Intelligent and Software Intensive Systems*, pages 87–93, 2008.

[10] A. Forward and T. C. Lethbridge. A taxonomy of software types to facilitate search and evidence-based software engineering. In *Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds*, page 14. ACM, 2008.

[11] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[12] A. N. Ghazi. Testing of heterogeneous systems. *Blekinge Institute of Technology Licentiate Dissertion Series*, 2014(03):1–153, 2014.

[13] A. N. Ghazi, J. Andersson, R. Torkar, K. Petersen, and J. Börstler. Information sources and their importance to prioritize test cases in heterogeneous systems context. In *Proceedings of the 21st European Conference on Systems, Software and Services Process Improvement (EuroSPI)*. Springer, 2014.

[14] T. L. Graves, M. J. Harrold, J.-M. Kim, A. Porter, and G. Rothermel. An empirical study of regression test selection techniques. In *Proceedings of the 20th International Conference on Software Engineering*, ICSE '98, pages 188–197, Washington, DC, USA, 1998. IEEE Computer Society.

[15] J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering*, pages 188–198. IEEE Computer Society, 2007.

[16] C. Kaner, J. Bach, and B. Pettichord. *Lessons learned in software testing.* John Wiley & Sons, 2008.

[17] E. Karahanna and D. W. Straub. The psychological origins of perceived usefulness and ease-of-use. *Information & Management*, 35(4):237 – 250, 1999.

[18] J. D. Kindrick, J. A. Sauter, and R. S. Matthews. Interoperability Testing. *StandardView*, 4(1):61–68, 1996.

[19] J. Kontio. A case study in applying a systematic method for cots selection. In *Software Engineering, 1996., Proceedings of the 18th International Conference on*, pages 201–209. IEEE, 1996.

[20] J. A. Lane. Sos management strategy impacts on sos engineering effort. In *New Modeling Concepts for TodayâĂŹs Software Processes*, pages 74–87. Springer, 2010.

[21] G. Lewis, E. Morris, P. Place, S. Simanta, D. Smith, and L. Wrage. Engineering systems of systems. In *Systems Conference, 2008 2nd Annual IEEE*, pages 1–6. IEEE, 2008.

[22] G. A. Lewis, E. Morris, P. Place, S. Simanta, and D. B. Smith. Requirements engineering for systems of systems. In *Systems Conference, 2009 3rd Annual IEEE*, pages 247–252. IEEE, 2009.

[23] C. Mao. Towards a Hierarchical Testing and Evaluation Strategy for Web Services System. *2009 Seventh ACIS International Conference on Software Engineering Research, Management and Applications*, pages 245–252, 2009.

[24] B. Marin, T. Vos, G. Giachetti, A. Baars, and P. Tonella. Towards testing future Web applications. *Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on*, pages 1 –12, may 2011.

[25] P. McMinn. Search-based software test data generation: a survey. *Software Testing, Verification and Reliability*, 14(2):105–156, 2004.

[26] J. Miller. Statistical significance testing–a panacea for software technology experiments? *Journal of Systems and Software*, 73:183–192, 2004.

[27] S. Mirarab, A. Ganjali, L. Tahvildari, S. Li, W. Liu, and M. Morrissey. A Requirement-Based Software Testing Framework : An Industrial Practice. *Test*, pages 452–455, 2008.

[28] M. Narita, M. Shimamura, K. Iwasa, and T. Yamaguchi. Interoperability verification for Web Service based robot communication platforms. *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 1029 –1034, dec. 2007.

[29] C. Nie and H. Leung. A survey of combinatorial testing. *ACM Computing Surveys*, 43(2):1–29, Jan. 2011.

[30] M. Ortega, M. Pérez, and T. Rojas. Construction of a systemic quality model for evaluating a software product. *Software Quality Journal*, 11(3):219–242, 2003.

[31] X. Pan and H. Chen. Using Organizational Evolutionary Particle Swarm Techniques to Generate Test Cases for Combinatorial Testing. *2011 Seventh International Conference on Computational Intelligence and Security*, pages 1580–1583, Dec. 2011.

[32] T. Perumal, A. R. Ramli, C. Y. Leong, S. Mansor, and K. Samsudin. Interoperability among heterogeneous systems in smart home environment. In *Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on*, pages 177–186, 2008.

[33] K. Petersen, M. Khurum, and L. Angelis. Reasons for bottlenecks in very large-scale system of systems development. *Information and Software Technology*, 2014.

[34] K. Petersen, K. Rönkkö, and C. Wohlin. The impact of time controlled reading on software inspection effectiveness and efficiency: A controlled experiment. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical*

Software Engineering and Measurement, ESEM '08, pages 139–148, New York, NY, USA, 2008. ACM.

[35] E. Piel, A. Gonzalez-sanchez, and H. G. Gross. Built-In Data-Flow Integration Testing in Large-Scale Component-Based Systems. *Ifip International Federation For Information Processing*, pages 79–94, 2010.

[36] D. M. Rafi, K. R. K. Moses, K. Petersen, and M. Mäntylä. Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In *Automation of Software Test (AST), 2012 7th International Workshop on*, pages 36–42. IEEE, 2012.

[37] S. M. A. Shah, C. Gencel, U. S. Alvi, and K. Petersen. Towards a hybrid testing process unifying exploratory testing and scripted testing. *Journal of Software: Evolution and Process*, 2013.

[38] T. Shiba. Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing. *Computer*, 2004.

[39] C. Thörn. Current state and potential of variability management practices in software-intensive smes: Results from a regional industrial survey. *Information & Software Technology*, 52(4):411–421, 2010.

[40] E. Van Veenendaal et al. *The testing practitioner*. UTN Publishers Den Bosch, 2002.

[41] D. Wang, B. Barnwell, and M. B. Witt. A Cross Platform Test Management System for the SUDAAN Statistical Software Package. *2009 Seventh ACIS International Conference on Software Engineering Research, Management and Applications*, pages 237–244, 2009.

[42] Z. Wang, B. Xu, L. Chen, and L. Xu. Adaptive Interaction Fault Location Based on Combinatorial Testing. *2010 10th International Conference on Quality Software*, pages 495–502, July 2010.

[43] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, and B. Regnell. *Experimentation in Software Engineering*. Springer, 2012.

[44] Q.-M. Xia, T. Peng, B. Li, and Z. wen Feng. Study on Automatic Interoperability Testing for E-Business. *CiSE 2009. International Conference on Computational Intelligence and Software Engineering, 2009.*, pages 1 –4, dec. 2009.