Using the 5W+1H Model in Reporting Systematic Literature Review: A Case Study on Software Testing for Cloud Computing

Changjiang Jia
Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Hong Kong
cjjia.cs@my.cityu.edu.hk

Yuen Tak Yu
Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Hong Kong
csytyu@cityu.edu.hk

***Abstract*—This paper documents a case study of using the 5W+1H model for reporting systematic literature review on software testing for cloud computing. To our knowledge, this is the first systematic literature review that applies the 5W+1H model, which is widely used in the journalism domain, to report the full picture of the research area in both software engineering and services computing. Existing guidelines on systematic literature review heavily rely on the researcher to pose the right research questions, and the review results are tightly focused on these research questions. For researchers new to a field, defining the right research questions that are effective in revealing the critical issues in the field can be challenging. Our case study demonstrates that the 5W+1H model provides an easy aid for the researcher to get over such initial challenges. As the researcher becomes more familiar with the field, he/she may then refine the research questions by adding more topic-specific contexts. In this way, the 5W+1H model serves to provide an exploratory framework to shape a systematic literature review. Applying to software testing for cloud computing, we are able to synthesize a comprehensive picture of recent researches on the field, including publication pattern, article citation immediacy, research topic diversity, research ideas for addressing testing challenges at different cloud service architectural layers. Based on the case study, we summarize the lessons learned on using the 5W+1H model in reporting systematic literature review.**

Keywords-5W+1H; cloud-based application; software testing; systematic literature review

I. INTRODUCTION

Software testing aims to expose bugs by executing programs over test cases, but it may typically consume 30–50% of the whole software development cost and leave latent bugs in the programs after test. Despite these shortcomings, the industry still widely applies it to assure software quality.

Cloud computing [5] is a recent research trend with great progress already made. Many large-scale cloud platforms, such as the Google App Engine [10], the Amazon Elastic Cloud [2], and the Microsoft Windows Azure [21] are available for commercial use. Software applications can be developed to run on top of such a platform [4]. Cloud computing promises to enable testing to be conducted with elastic resource provision, service access anywhere connected to the computing cloud, with no or little up-front hardware investment [29].

The massive computational ability of a cloud might well be useful to deal with the resource-hungry nature of software testing. But does cloud computing, on the other hand, pose new technical challenges to software testing techniques? We therefore ask: *To what extent do cloud computing and software testing techniques meet and interact with each other?* To ease our discussion, we refer to the intersection between research on *Cloud computing* and that on *Software Testing* as ***the CST-interface*** in this paper.

Considering that cloud computing and software testing are both active research areas, the CST-interface is expected to have received broad attention in the research community. As an unstructured literature review may not reveal the full picture of the CST-interface, we exploited the 5W+1H model in this paper to formulate a systematic literature review (SLR) on the CST-interface, in accordance with the authoritative review guidelines of Kitchenham and Charters [15].

The 5W+1H model is a popular model adopted in the journalism domain [22] to describe the full picture of a news story or investigative case. To our knowledge, this paper is the first in both software engineering and services computing research domains to report a SLR using the 5W+1H model. Using this model, we have proposed six conjectures on the research status of the field and verify them against the collected literature evidences. For easy reference, we have summarized the research questions, conjectures, major findings and interpretation of our SLR in Table I. Our case study provides the first set of empirical data to portray the developing nature of cloud testing research. Readers may refer to our technical report [14] for extended discussions on this topic.

In Section II, we describe the literature review protocol adopted in this case study and briefly discuss the SLR results, its related work and threats to validity. In Section III, we summarize our reflections on the lessons learned from using the 5W+1H model in reporting our SLR, which may be useful to readers who consider doing a SLR in a different area. Section IV discusses the related work and Section V concludes the paper.

II. CASE STUDY OF SLR ON THE CST-INTERFACE

SLR aims to synthesize existing work on a research area in a comprehensive and objective manner. We adopted the guidelines for performing SLR in software engineering domain in [15][17] to conduct and present our SLR.

978-0-7695-5039-8/13 $26.00 © 2013 IEEE
DOI 10.1109/QSIC.2013.13

TABLE I. SUMMARY OF RESEARCH QUESTIONS (RQS), CONJECTURES, MAJOR FINDINGS AND ASSESSMENT

RQ1. "Who": Authors and Countries

Conjecture C1: Plentiful papers have been published by diverse research groups and from different countries across the globe.

Major findings:

1. Our SLR has identified a primary set (PS) of 38 papers on the CST-interface out of an initial set (IS) of 2949 records of papers on the broader area of "cloud testing" research (which includes, for instance, hardware or network testing).
2. Twenty-two papers in PS are affiliated with China and/or USA. In each of the other 10 countries, only one research group published papers in PS.
3. Four of the 10 countries which published most cloud testing papers in IS included in Scopus contribute zero paper to the CST-interface.

Assessment: The CST-interface has not been widely researched within the review period. Moreover, the author/country distributions of publications on the CST-interface differ substantially from those on cloud testing. We cannot find adequate literature evidence to support the conjecture C1.

RQ2. "Why": Objectives of Research

Conjecture C2: Papers on multiple cloud service architectural layers (that is, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)) address the same kind of technical challenges with regard to the testing topics.

Major findings:

1. Among the total 12 testing topics identified, four (one-third) are studied in papers on more than one cloud service architectural layer.
2. Two of the latter four topics are robustness testing and testing strategy, which together account for 21.1% of the papers in PS. These papers address the same kind of challenges, namely, faulty cloud service layer and development of Testing-as-a-Service (TaaS).
3. With regard to the 12 testing topics and 3 architectural layers, only research on 16 out of the 36 combinations has been reported in PS.
4. Six out of the 38 papers in PS are survey or viewpoint papers on issues such as testing tool features and opinions of practitioners.

Assessment: The first two findings above demonstrate literature evidence that is in line with the conjecture C2. Moreover, more than half of the combinations of testing topics and architectural layers have not been studied, indicating that much opportunity on the CST-interface remains to be explored. Furthermore, the notably high proportion of survey and viewpoint papers on different aspects of the CST-interface seem to indicate that each aspect is rich and "unclear" enough to warrant a separate study.

RQ3. "What": Research Ideas

Conjecture C3: Testing research ideas for addressing the challenges in different cloud service architectural layers are different.

Major findings:

1. Papers on IaaS propose randomized testing and fault-based testing techniques to expose faults in the virtual machine implementations.
2. Papers on PaaS develop (a) techniques for testing SaaS applications to deal with failure-simulation or non-deterministic PaaS issues, (b) methodology for benchmarking, and (c) strategies to lower testing costs by exploiting the elastic property of PaaS.
3. Papers on SaaS span over 9 testing topics. A diverse set of ideas have been studied that range from performance metrics across multiple cloud service architectural layers as well as usage-specific and general models of TaaS, test parallelization, controllability and observability, test workloads, regression testing, to the detection of vulnerability faults, and so on.

Assessment: The only idea common across different layers is fault-based testing. So the conjecture C3 is consistent with our data. Research at a lower layer involves either randomized or fault-based testing. Thus, many other systematic testing approaches have not been explored.

RQ 4. "Where": Patterns of Papers at Different Cloud Service Architectural Layers and Types of Publication Venues

Conjecture C4: Every architectural layer receives good research attention. Also, consistent with the norm for computer science research, the majority of recent papers are published as research articles in conference proceedings, yet there is a good presence of journal papers.

Major findings:

1. Papers on SaaS contribute to 57.9% of PS, followed by PaaS (15.8%) and IaaS (10.5%). The rest (15.8%) are survey/viewpoint papers.
2. 92.1% of the articles are published in conference proceedings. Only a very low percentage (7.9%) of the articles is journal papers.

Assessment: Papers tend to focus on testing challenges at the upper layer (SaaS). Both the ratio (5.5 : 1) between SaaS and IaaS papers and the ratio (11.7 : 1) between conference/workshop and journal papers are notably high, indicating that the CST-interface is not yet mature. Moreover, most articles are published in conference/workshop proceedings. The conjecture C4 is only partially inconsistent with our data.

RQ 5. "When": Article Citation Immediacy

Conjecture C5: Many papers are quickly referenced by other papers.

Major findings:

1. 73.7% of the papers in PS have citation relations among themselves within the two-year time frame of this study. Within PS, 21.1% of the papers were cited by papers published in the same year. The immediacy ratio is very high.
2. The number of papers in PS published in 2010 is 16 and that in 2011 is 22. The 37% increase from 2010 to 2011 is significant.

Assessment: Publications of the CST-interface receive prompt research attention in terms of citations and have increased significantly from 2010 to 2011. The conjecture C5 is consistent with the literature evidence found.

RQ 6. "How": Article Inter-Relevance

Conjecture C6: Many papers on various cloud service architectural layers and software testing topics are inter-referenced to evolve the CST-interface.

Major findings:

1. Most of the citation relations of individual topics are within the same cloud service architectural layers. Only 7 cross-layer citation relations can be located.
2. Three (fuzzing, migration testing, and log analysis) out of the 12 testing topics have no citation relations to other topics within PS.
3. In terms of citation relations within PS, the topics robustness testing, scalability of testing, and integration testing are the three most important and actively researched ones on the CST-interface.
4. Topics on IaaS and PaaS, as well as their intersection with SaaS, have not been explored sufficiently to evolve the CST-interface research.

Assessment: Despite the generally high immediacy ratio in PS, testing topics across different cloud service architectural layers are still not within mainstream research, as seen by their few inter-citations. We do not find adequate literature evidence to support the conjecture C6.

A. Research Questions

Our SLR adopted the 5W+1H model [22] to extract a contemporary picture of the CST-interface research from our data. Specifically, in our context, we studied the CST-interface by proposing one research question (RQ) for each of the six dimensions of the (5W+1H) model as follows.

RQ1: [*Who*] Who were doing research on the testing of software deployed in the clouds?

RQ2: [*Why*] Why are the research works needed? That is, what research objectives on software testing were stated in the articles?

RQ3: [*What*] What kinds of software testing research ideas were presented in the articles?

RQ4: [*Where*] At which cloud service architectural layers were the articles focused on? Where were the articles published? Were the articles published in typical types of publication venues?

RQ5: [*When*] When did the articles start to show impact? Were articles immediately referenced among them?

RQ6: [*How*] How were the articles inter-referenced? What were the citation patterns among the articles?

For each RQ, we also formulated a conjecture (C) according to common perception. We aimed to assess to what extent the collected papers present adequate evidence to support or invalidate the conjectures (C1−C6).

C1: Plentiful papers have been published by diverse research groups and from different countries across the globe.

C2: Papers on multiple cloud service architectural layers address the same kind of technical challenges with regard to the testing topics.

C3: Testing research ideas for addressing the challenges in different cloud service architectural layers are different.

C4: Every architectural layer receives good research attention. Also, consistent with the norm for computer science research, the majority of recent papers are published as research articles in conference proceedings, yet there is a good presence of journal papers.

C5: Many papers are quickly referenced by other papers.

C6: Many papers on various cloud service architectural layers and software testing topics are inter-referenced to evolve the CST-interface.

We adopted the generally-accepted classification of three cloud service architectural layers discussed in [3][20] to analyze research questions and conjectures: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

IaaS provides virtualized computing resources with configurable virtual machines to users. Example testing issues include effective detection of failures and anomalies of the virtual machines and their connectivity.

PaaS provides users a virtual platform consisting of software and development tools to build their own SaaS. Example components in such a platform include a guest operating system, storage, platform-based software, and API. Example testing issues include the detection of mismatched interaction between a platform component and an application, or the anomalous behavior of a platform component that may affect an application's behavior.

SaaS provides usable software applications for users to operate or synthesize new applications on top of them. Example testing issues include test case fixing in migration testing and the optimal use of cloud resources for testing.

Under the three cloud service architectural layers, we identified 12 major testing topics in total from the PS papers as follows: log analysis [S21], concurrency testing [S7], robustness testing [S2][S11][S12][S13][S16], fuzzing [S19], performance testing [S10][S24][S28][S32], migration testing [S8], testing strategy [S27][S35][S36], context sensitivity [S4][S15], testing scalability [S3][S5][S22][S30][S34][S38], integration testing [S17][S18][S31][S33], regression testing [S6][S14], and security testing [S29][S37]. In addition, there are six PS papers that are survey in nature [S1], [S20], [S25], [S26] or share the viewpoints of the authors [S9], [S23].

This paper only presents the final result of PS paper classification under different testing topics. For the process of analysis and classification of each PS paper, readers may refer to our technical report [14] for more details.

B. Paper Identification Process

We now elaborate our paper identification process.

Databases: Our SLR used three popular databases to identify the literature: ACM Digital Library (ACM DL) [1], IEEE Xplore [12], and Scopus Citation Database (Scopus CD) [24].

Keywords: We used two keywords "Cloud" and "Testing" as the base and enumerated their popular variants to the final compound keyword: (Cloud AND (Testing OR Analysis OR Test OR Analyzing OR Analyze)).

Inclusion Criterion (IC) and Exclusion Criterion (EC): Following [15], we specified an *initial inclusion setting* (**IIS**) by matching the keywords with the *abstract* of a paper. We focused on *peer-reviewed* publications to ensure that the papers in our collection at least reach acceptable quality. In order to study the recent status on the CST-interface, we included papers published in the *recent two whole years*. The topic domain was set to *computer science*.

> *Initial Inclusion Setting* (IIS): Search the keywords in the abstract of a paper, which was published in a conference proceedings or refereed journal in 2010 and 2011, inclusively, and is within the smallest domain that includes the computer science domain.

Specifically, we set the publication venue as *journals*, *proceedings* and *transactions* for ACM DL, *conference publications* and *journals* & *magazines* for IEEE Xplore, and *journals* and *conference proceedings* for Scopus CD. We set the topic domain as *Computing & Processing (Hardware/ Software)* for IEEE Xplore (the "smallest" one that includes "computer science") and *Computer Science* for Scopus CD. We could not specify any topic domain for ACM DL as it provided no such option. We then refined the IIS by keywords into IC as follows.

IC:	Apply IIS with the keyword: ((Cloud) AND (Testing OR Analysis OR Test OR Analyzing OR Analyze)).

By performing IC, we extracted the initial set **(IS)** of papers concerned with cloud computing and testing. IS consisted of 2949 records. We then applied three exclusion criteria (EC1−EC3) in stages to further refine IS as follows.

EC1:	Exclude a paper of fewer than 4 pages [15].
EC2:	Exclude a paper that mentions no issue on cloud computing or on software testing in its abstract.
EC3:	Exclude a paper that mentions no issue on cloud computing or software testing in either introduction or conclusion of the paper. Remove duplications due to multiple records that refer to the same paper.

EC1 reduced the size of IS to 2807. EC2 eliminated a large number of papers on irrelevant topics such as storage, hardware configuration, and network, thereby reducing the number of records to 91. For EC3, we repeated what we did for EC2, but checked the paper's introduction and conclusion instead of the abstract. For duplicated records, we firstly kept the record from ACM DL if it existed, followed by IEEE Xplore and finally Scopus CD. Fig. 1 shows the number of records obtained successively by IC and then EC1−EC3.

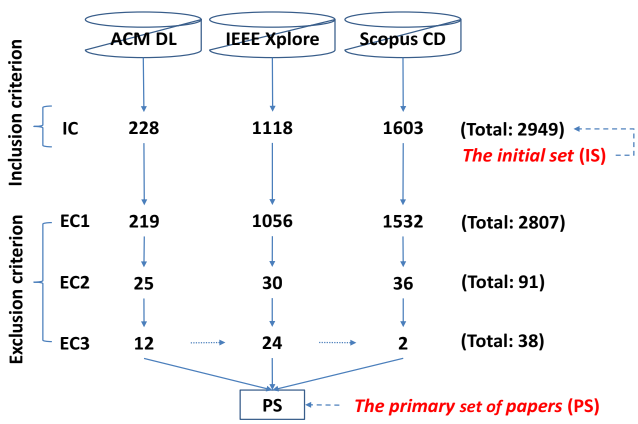

Figure 1. Paper Identification Process.

We finally obtained 38 distinct papers, as listed in Appendix I. We refer to this collection as the primary set (PS). We noted that only 1.3% of the papers in IS were related to both cloud computing and software testing.

C. Quality Assurance

We browsed the official website of each paper in PS to confirm that it had stated a mandatory process of peer review for paper acceptance. We also verified that each paper indeed satisfied all the specified inclusion and exclusion criteria by manually reading its full text. For instance, we found that although ACM SIGOPS Operating System Review did not specify a paper review process, its paper entitled "Cloud9: A Software Testing Service" [S5] was a published version of [8] which had been peer-reviewed earlier in a workshop with no proceedings. Hence, we decided to include [S5] in PS. All other papers in PS were also verified to have been peer-reviewed and met all the inclusion and exclusion criteria.

Google Scholar: We did not use Google Scholar [11] as the bibliographic database in our SLR because it did not consistently index papers according to publication year nor provide any option to search on the abstract field. Searching on the paper title only would miss many important references like [S3][S4][S5][S6][S8][S10][S14][S16][S19][S21][S24][S26][S29][S30][S32]. As a triangulation check, we applied IC but replaced the search on the abstract setting by a search in the "with all of the words" field in the advanced search menu, setting "where my words occur" to "anywhere in the article" and using the "return articles dated between 2010 and 2011". We then scanned through the first 2000 returned records (out of an estimated 18800 hits) and found two more papers [6][9] that were apparently related to our SLR. To maintain our review protocol, we chose to discuss these two papers separately in subsection E of this Section.

D. Result Discussion

We now discuss the results of our case study in Table I.

The "Who" dimension considers the CST-interface from the perspective of researchers. The conjecture C1 proposes that the topic has been widely investigated by worldwide researchers. However, our case study shows that only 1.3% of papers in IS are related to the topic. C1 is not supported by this evidence, which indicates that the CST-interface topic has not received wide attention in the research community.

The dimensions of "Why", "What" and "Where" study the CST-interface from the perspective of research contents. We identified 12 testing topics from PS papers in the "Why" dimension to cluster the motivation categories. Four topics, namely robustness testing, performance testing, testing strategy and context sensitivity, had been studied in more than one cloud service architectural layer. This fact supports the conjecture C2, which proposes that some common technical challenges exist in multiple cloud service architectural layers. In terms of covered testing topics, the SaaS layer (9 topics) was more actively studied than the layers of PaaS (5 topics) and IaaS (2 topics) [14]. The papers in PS only involve 16 out of the 36 possible layer-specific testing topics. Both the unbalanced layer-topic distribution and the big proportion of un-investigated layer-specific topics indicate that a generally-accepted clear understanding of the CST-interface had not yet emerged. The result of the "What" dimension tells that research ideas on each cloud service architectural layer differed greatly. These findings support the conjecture C3. In the "Where" dimension, we found that SaaS received much more research attention than the lower layers. All data in the three dimensions consistently indicate that the CST-interface was not yet mature and the surveyed work mainly focused on the SaaS layer.

The dimensions of "When" and "How" explore the CST-interface from the perspective of research evolution. The immediacy rate of article citation in PS was high (up to 73.7%). However, most of those citations appeared as a brief summary within the sections of introduction or related work. Very few topics (e.g., testing scalability) evolved in the manner of proposing new ideas to address challenges of previous ideas. These results revealed that the CST-interface was still not well-formulated in the research community.

Next, we briefly reviewed the six papers in PS that are survey in nature [S1], [S20], [S25], [S26] or share the viewpoints of the authors [S9], [S23].

Bai et al. [S1] surveyed existing cloud testing tools and classified each work into four dimensions: simulation, service mocking, test job parallelization, and environment virtualization. Riungu et al. [S25], [S26] interviewed 11 practitioners and came up with two main conclusions. The first main conclusion was that using a computing cloud would benefit testing tasks that are loosely related (e.g., high-volume automated testing, performance testing). The second was that cloud testing management (e.g., user contexts and test process management) was an important issue for cloud testing usage. Mohammad et al. [S20] considered that the lack of test environment rendered the comparison between service composition methods difficult. They outlined the differences among traditional testing, service-oriented testing and cloud service testing, and posed several open questions for further study.

Fang et al. [S9] considered three particular aspects to be very important to cloud testing: ubiquitous test middleware to manage cloud resources and test tasks, multiple test scenario convergence, and co-simulation of different components.

E. Related work on cloud testing survey

There are several closely related papers beyond our PS set which we now discuss as follows.

Chhabra et al. [6] urged for more research effort to six software engineering aspects. Three of the six aspects were taken from the web services domain: service selection, testing criteria for cloud interactions, and availability of source code of third-party software components. From the testing perspective, these three aspects were explored by Tsai et al. [S31], Chan et al. [4], and King and co-authors [S17], [S18], respectively. The other three aspects are due to the distributed nature of nodes in a computing cloud: query oriented programming pattern (e.g., MapReduce), distributed software execution model, and the loosely-coupled design to deal with issues in cloud integration. This paper [6] is affiliated with India and has no citation relationship with [S11] or any paper reviewed in our SLR.

Gao et al. [9] classified cloud testing into seven categories, but had not explored the dimension of cloud service architectural layer as what we have presented in the present SLR case study. This work [9] was the result of collaboration between researchers from USA and China, and was cited by two papers [S1] [S10] (both published in 2011) reviewed in our SLR.

Chan et al. [4] (2009) proposed a few test data adequacy criteria (e.g., evaluating each predicate in their graph modeling to true at least once) to assure the quality of an application. This paper is affiliated with China. Their work proposed domain-specific test adequacy criteria to address test challenges at the SaaS layer, which is a testing topic not studied by any paper in PS. This work has been cited by [S6][S17] (2011), [S18] (2010) in PS and one [9] of the two papers we identified from Google Scholar as mentioned toward the end of Section II.C.

We also found an independent mapping study [16] article on cloud-based testing [23] published in a non-refereed newsletter. It focused on TaaS, though it also included papers on web service testing. Unlike our SLR, their mapping study did not explore the dimension of cloud service architectural layer and did not study the citation relations among the surveyed papers.

F. Threats to Validity

Construct Threats to Validity: We have measured the relationships among papers in PS by the intra-PS citation numbers. Using other metrics may reveal a different picture of relationships among the papers.

Internal Threats to Validity: This study has used only three databases. Recently, there are some other open-access paper archival repositories. Including them in the study may introduce many other articles and, hence, reveal different paper statistics. Moreover, there is invariably a time delay for inclusion of papers into each database. For instance, the proceedings of a conference held in December 2011 may not have appeared in the three databases at the time of data extraction. To address this threat, one may conduct several rounds of data extraction until the dataset is stabilized, but this will take an indefinite amount of time and a great deal more effort to track the evolution.

Assessing the relevance of a paper to the CST-interface is a challenging task. To ensure impartiality, the task was performed independently by more than one researcher in this case study. Our chosen keywords and their variants certainly influenced the set of articles generated for analysis. Our focus is software testing for cloud computing. We have extended the concept of testing to "analysis" in formulating the keyword string. We did not include "software" in the keyword because to our knowledge, a paper may not mention "software" or "program" in its abstract. This omission might have introduced ambiguity. To mitigate this risk, we read the included papers in detail to ensure their true relevance to software testing. We have not loosened our criterion on the term "cloud", because we could not find what other popular terms have been used with a similar meaning as "cloud" in the publication dataset. We once considered the terms "utility" or "grid", but they usually refer to non-cloud articles (such as papers on the testing of grid-based applications) that are related to some classical topics rather than the emerging topic which we are interested in. Finally, we only included papers that had been peer-reviewed. As such, some work like technical reports, white papers, or patents may have been excluded. This represents a limitation of our current case study.

External Threats to Validity: Our result may not be generalizable to other topics or the use of alternative inclusion and exclusion criteria. To generalize the 5W+1H model, one may extend the questions in each dimension. It would be interesting to compare our finding with further work that reviews the challenges of testing research in other topics.

III. LESSON LEARNED

The 5W+1H model has been widely employed in the journalism domain to completely report a story. Applying it to report our SLR, we found that the use of 5W+1H model does help new researchers to get the overall landscape of a research topic. We would like to share our experiences in this section.

Dimension Mapping from 5W+1H to SLR: Researchers who are new to a research topic can benefit greatly by being informed of the representative researchers (*Who*) and important publication venues (*Where*) of this new topic. This information enables them to quickly assess the state of the art (*When*) of this topic. It is easy to extract the meta-data of the three dimensions from published articles. The researchers will also need to understand the research issues or problems (*What*), the motivations or opportunities (*Why*), and the relationships (*How*) of these pieces of work. These tasks are relatively time-consuming and difficult to get a clear and consistent result. Multiple investigators need to cooperate to resolve the inconsistencies among individual judgments.

Reporting Completeness of SLR via 5W+1H: The 5W+1H model may completely report a news story or investigative case. However, we found it is insufficient to completely and precisely report a SLR on software testing by directly mapping this generic model to organize the review results. For example, at the beginning of our SLR, we simply studied papers in PS one by one to extract the research problems (*What*) and motivations (*Why*). However, we found each paper could not relate with other papers effectively to help us understand the whole research picture. This posed a challenge for us to synthesize the common research motivations and challenges. To address this issue, we extended the 5W+1H model with topic-specific properties. In this case study, we adopted the generally accepted three-layer cloud architecture model to refine the 5W+1H model except for the "Who" and "When" dimensions. Vigilant readers may have noticed that different extended classifications would add various effects to the generic 5W+1H model. So it may be confusing and challenging for investigators to propose and select an effective topic-specific classification scheme to concretize the generic 5W+1H model.

Traditional SLR vs **5W+1H Model**: We can distinguish two categories of SLRs: one is to address specific research questions (e.g., concurrency testing techniques, web service testing) while another is to identify the research trend of a new topic (e.g., a mapping study). For the first type of SLR, the investigators define the research questions precisely to reflect exactly what they want to know. The traditional SLR process is designed to guide the review process to be tightly centered on those defined research questions. In this situation, our experience is that following the traditional SLR rather than using the general 5W+1H model can be more effective. For the second type of SLR, investigators may have less prior knowledge on the topic and, hence, find it challenging initially to propose effective research questions to capture what they want to know about this topic. Then it is better to apply the 5W+1H model as a good starting point to define the research questions first. One may later gradually add more topic-specific contexts to refine the research questions to guide the review process. To this end, the 5W+1H model provides an exploratory framework to shape a SLR.

IV. RELATED WORK

This section reviews the literature related to SLRs and research reporting with the 5W+1H model.

To assure the rigor of the SLR process, Kitchenham et al. [18] proposed to apply evidence-based concepts, adopted from medical research, into the software engineering domain. Kitchenham and Charters [15] then defined a general framework for SLR on software engineering topics. Kitchenam and other colleagues [19] found that using a manual search process may miss some relevant papers and, further, propose to automate the search process [17].

The automated search process is dominated by the search keyword string in various databases. There are alternative suggestions to paper searching, such as reference-based search strategies [25]. Webster and Watson [26] proposed an incremental three-step paper searching approach called snowballing. This approach starts with searching papers from some generally-accepted high quality publication venues (that is, journals or proceedings) as the first step. Then in the second step, it includes papers in the reference lists of those collected in the first step. In the third step, it searches the databases (such as ISI Web of Science) to collect those papers that cite the papers collected in previous two steps. This approach has been applied to SLR of the software engineering domain [27].

To the best of our knowledge, there is no existing work applying the 5W+1H model to SLR, though it has been used for some other purposes in software engineering. Chung et al. [7] applied the 5W+1H model to re-documentation of a given legacy system with UML visual models. They map the six dimensions of the 5W+1H model to topic-specific contexts as follows: (*Why*) the benefits of doing re-documentation to software development practices, (*When*) different phases in the whole software development process, (*Who*) the role of software developers, (*Where*) different views (such as use case view) of a legacy system, (*What*) the usage of UML elements in various views by different roles, and (*How*) the process of constructing the other dimension elements and building relations.

Context-aware applications rely on captured context information to maintain their performance. However, existing context modeling techniques are specific to certain information (such as location), leading to a tight cohesion between contexts and applications. Jang and Woo [13] used the 5W+1H model to build unified user-centric contextual information to be shared among several applications. The six dimensions of the model would completely cover the complicated context.

Yang et al. [28] used the 5W+1H model to build a conceptual modeling framework to analyze the domain concepts and relations from the six aspects. However, none of the above work applies the model to report SLR in the field.

V. CONCLUSION AND FURTHER WORK

This paper has reported the first refereed SLR on software testing work for cloud computing published in 2010–2011. Our work is the first in the software engineering and service-oriented computing community to adopt the 5W+1H model to analyze and report findings in SLR. We found the model particularly useful to supplement SLR by providing an exploratory framework for the researcher to define initial research questions to study a field with which he/she is less experienced. As the researcher gains better understanding, topic-specific context can be added to refine the generic 5W+1H model, thereby building a full picture of the field by extracting views from the six dimensions and perspectives. Our case study of using the 5W+1H model in reporting SLR has revealed interesting findings as summarized in Table I. Constrained by the scope of this paper, more details of the SLR case study results are not reported here but can be found in our technical report [14].

Future work includes extending the study time period to validate and refine the conjectures and findings extracted from this case study. We are also interested in comparing SLR with and without using the 5W+1H on investigating a research topic.

ACKNOWLEDGMENT

This work is supported in part by Strategic Research Grant (project number 7002873) from City University of Hong Kong and General Research Fund (project number 111410) from Hong Kong Research Grants Council.

REFERENCES

[1] ACM Digital Library. Available at: http://dl.acm.org/. 2012.

[2] Amazon Elastic Cloud. Available at: http://aws.amazon.com/ec2/. 2012.

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical report UCB/EECS-2009-28, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.

[4] W.K. Chan, L. Mei, and Z. Zhang, "Modeling and Testing of Cloud Applications," in *Proc. of APSCC 2009*, 111−118, 2009.

[5] T.M. Chen, "The Cloud Goes Mainstream," *IEEE Network*, 25(4):2−3, 2011.

[6] B. Chhabra, D. Verma, and B. Taneja, "Software Engineering Issues from the Cloud Application Perspective," *International Journal of Information Technology and Knowledge Management*, 2(2):669−673, 2010.

[7] S. Chung, D. Won, S.-H. Baeg, and S. Park, "Service-oriented Reverse Reengineering: 5W1H Model-driven Re-documentation and Candidate Services Identification," in *Proc. of SOCA 2009*, 1−6, 2009.

[8] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, "Cloud9: A Software Testing Service," in *Proc. of LADIS 2009*, 1−6, 2009.

[9] J. Gao, X. Bai, and W.-T. Tsai, "Cloud-Testing: Issues, Challenges, Needs and Practice," *Software Engineering: An International Journal*, 1(1):9−23, 2011.

[10] Google App Engine. Available at: http://code.google.com/intl/en/appengine/. 2013.

[11] Google Scholar. Available at: http://scholar.google.com. 2013.

[12] IEEE Xplore. Available at: http://ieeexplore.ieee.org. 2013.

[13] S. Jang and W. Woo, "Unified Context Representing User-Centric Context: Who, Where, When, What, How and Why," in *Proc. of UbiComp Workshop*, 26−34, 2005.

[14] Changjiang Jia, W.K. Chan, and Y.T. Yu, "A Systematic Literature Review on Software Testing Research for Cloud Computing", Technical Report, City University of Hong Kong, 2013.

[15] B.A. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[16] B.A. Kitchenham, D. Budgen, and O.P. Brereton, "Using Mapping Studies as the Basis for Further Research: A Participant-Observer Case Study," *Information and Software Technology*, 53(6):638−651, 2011.

[17] B.A. Kitchenham, R. Pretorisu, D. Budgen, P. Brereton, M. Turner, M. Niazi, and S.G. Linkman, "Systematic Literature Reviews in Software Engineering: A Tertiary Study," *Information and Software Technology*, 52(8):792−805, 2010.

[18] B.A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-Based Software Engineering," in *Proc. of ICSE 2004*, 273−281, 2004.

[19] B. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic Literature Reviews in Software Engineering: a Systematic Literature Review," *Information and Software Technology*, 51(1):7–15, 2009.

[20] A. Lenk, M. Klem, J. Nimis, S. Tai, and T. Sandholm, "What's Inside the Cloud – An Architectural Map of the Cloud Landscape," in *Proc. of ICSE Workshop on Software Engineering Challenges of Cloud Computing*, 23−31, 2009.

[21] Microsoft Windows Azure. Available at: http://www.microsoft.com/windowsazure/. 2013.

[22] Z. Pan and G.M. Kosicki, "Framing Analysis: An Approach to News Discourse," *Political Communication*, 10(3):55−73, 1993.

[23] S. Priyanka, I. Chana, and A. Rana, "Empirical Evaluation of Cloud-based Testing Techniques: A Systematic Review," *ACM SIGSOFT Software Engineering Notes*, 37(3):1−9, 2012.

[24] Scopus Citation Database. Available at: http://www.scopus.com. 2013.

[25] M. Skoglund and P. Runeson, "Reference-based Search Strategies in Systematic Reviews," in *Proc. of EASE 2009*, 31–40, 2009.

[26] J. Webster and R.T. Watson, "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Quarterly*, 26(2):xiii–xxiii, 2002.

[27] C. Wohlin and R. Prikladniki, "Systematic Literature Reviews in Software Engineering," *Information and Software Technology*, 55(6):919–920, 2013.

[28] L. Yang, Z. Hu, L. Jun, and T. Guo, "5W1H-based Conceptual Modeling Framework for Domain Ontology and Its Application on STPO," in *Proc. of SKG 2011*, 203–206, 2011.

[29] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-of-the-Art and Research Challenges," *Journal of Internet Services and Applications*, 1(1):7−18, 2010.

APPENDIX I: PAPERS INCLUDED IN OUR SLR

[S1] X. Bai, M. Li, B. Chen, W.T. Tsai, and J. Gao, "Cloud Testing Tools," in *Proc. of International Symposium on Service-Oriented System Engineering*, 1−12, 2011.

[S2] T. Banzai, H. Koizumi, R. Kanbayashi, T. Imada, T. Hanawa, and M. Sato, "D-Cloud: Design of a Software Testing Environment for Reliable Distributed Systems using Cloud Computing Technology," in *Proc. of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 631−636, 2010.

[S3] S. Bucur, V. Ureche, C. Zamfir, and G. Candea, "Parallel Symbolic Execution for Automated Real-world Software Testing," in *Proc. of 6th European Conference on Computer Systems*, 183−198, 2011.

[S4] G. Candea, S. Bucur, and C. Zamfir, "Automated Software Testing as a Service," in *Proc. of 1st ACM Symposium on Cloud Computing*, 155−160, 2010.

[S5] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, "Cloud9: A Software Testing Service," *ACM SIGOPS Operating Systems Review*, 43(4):5−10, 2010.

[S6] M.B. Cooray, J.H. Hamlyn-Haris, and R.G. Merkel, "Test Reconfiguration for Service Oriented Applications," in *Proc. of 4th International Conference on Utility and Cloud Computing*, 300−305, 2011.

[S7] C. Csallner, L. Fegaras, and C. Li, "Testing MapReduce-style Programs," in *Proc. of 19th ACM SIGSOFT Symposium and 13th European Conference on Foundations of Software Engineering*, 504−507, 2011.

[S8] X. Ding, H. Huang, Y. Ruan, A. Shaikh, B. Peterson, and X. Zhang, "Splitter: A Proxy-based Approach for Post-migration Testing of Web Applications," in *Proc. of 5th European Conference on Computer Systems*, 97−110, 2010.

[S9] W. Fang and Y. Xiong, "Cloud Testing: the Next Generation Test Technology," in *Proc. of 10th International Conference on Electronic Measurement and Instruments*, 291−295, 2011.

[S10] J. Gao, P. Pattabhiraman, X. Bai, and W.-T. Tsai, "SaaS Performance and Scalability Evaluation in Clouds," in *Proc. of 6th International Symposium on Service-Oriented System Engineering*, 61−71, 2011.

[S11] H.S. Gunawi, T. Do, P. Joshi, P. Alvaro, J.M. Hellerstein, A.C. Arpaci-Dusseau, R.H. Arpaci-Dusseau, K. Sen, and D. Borthakur, "FATE and DESTINI: A Framework for Cloud Recovery Testing," in *Proc. of 8th USENIX Conference on Networked Systems Design and Implementation*, 1−14, 2011.

[S12] T. Hanawa, H. Koizumi, T. Banzai, M. Sato, S. Miura, T. Ishii, and H. Takamizawa, "Customizing Virtual Machine with Fault Injector by Integrating with SpecC Device Model for a Software Testing Environment D-Cloud," in *Proc. of 16th Pacific Rim International Symposium on Dependable Computing*, 47−54, 2010.

[S13] T. Hanawa, T. Banzai, H. Koizumi, R. Kanbayashi, T. Imada, and M. Sato, "Large-scale Software Testing Environment Using Cloud Computing Technology for Dependable Parallel and Distributed Systems," in *Proc. of 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 428−433, 2010.

[S14] S. Huang, Z. Li, Y. Liu, and J. Zhu, "Regression Testing as a Service," in *Proc. of Annual SRII Global Conference*, 315−324, 2011.

[S15] W. Jenkins, S. Vilkomir, P. Sharma, and G. Pirocanac, "Framework for Testing Cloud Platforms and Infrastructures," in *Proc. of International Conference on Cloud and Service Computing*, 134−140, 2011.

[S16] P. Joshi, H.S. Gunawi, and K. Sen, "PREFAIL: A Programmable Tool for Multiple-failure Injection," in *Proc. of ACM International Conference on Object Oriented Programming Systems Languages and Applications*, 171−188, 2011.

[S17] T.M. King, A.S. Ganti, and D. Froslie, "Enabling Automated Integration Testing of Cloud Application Services in Virtualized Environments," in *Proc. of Conference of the Center for Advanced Studies on Collaborative Research*, 120−132, 2011.

[S18] T.M. King and A.S. Ganti, "Migrating Autonomic Self-testing to the Cloud," in *Proc. of 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 438−443, 2010.

[S19] L. Martignoni, R. Paleari, G.F. Roglia, and D. Bruschi, "Testing System Virtual Machines," in *Proc. of 19th International Symposium on Software Testing and Analysis*, 171−182, 2010.

[S20] A.F. Mohammad and H. Mcheick, "Cloud Services Testing: An Understanding," *Procedia Computer Science*, 5:513−520, 2011.

[S21] M. Nagappan, "Analysis of Execution Log Files," in *Proc. of 32nd ACM/IEEE International Conference on Software Engineering* — vol. 2 (*Doctorial Symposium*), 409−412, 2010.

[S22] M. Oriol and F. Ullah, "YETI on the Cloud," in *Proc. of 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 434−437, 2010.

[S23] T. Parveen and S. Tilley, "When to Migrate Software Testing to the Cloud?" in *Proc. of 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 424−427, 2010.

[S24] S. Patil, M. Polte, K. Ren, W. Tantisiriroj, L. Xiao, J. López, G. Gibson, A. Fuchs, and B. Rinaldi, "YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores," in *Proc. of 1st ACM Symposium on Cloud Computing*, 1−14, 2011.

[S25] L.M. Riungu, O. Taipale, and K. Smolander, "Research Issues for Software Testing in the Cloud," in *Proc. of IEEE 2nd International Conference on Cloud Computing Technology and Science*, 557−564, 2010.

[S26] L.M. Riungu, O. Taipale, and K. Smolander, "Software Testing as an Online Service: Observations from Practice," in *Proc. of 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 418−423, 2010.

[S27] P. Robinson and C. Ragusa, "Taxonomy and Requirements Rationalization for Infrastructure in Cloud-based Software Testing," in *Proc. of IEEE 2nd International Conference on Cloud Computing Technology and Science*, 454−461, 2011.

[S28] N. Snellman, A. Ashraf, and I. Porres, "Towards Automatic Performance and Scalability Testing of Rich Internet Applications in the Cloud," in *Proc. of 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, 161−169, 2011.

[S29] V. Srivastava, M.D. Bond, K.S. McKinley, and V. Shmatikov, "A Security Policy Oracle: Detecting Security Holes Using Multiple API Implementations," in *Proc. of 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation*, 343−354, 2011.

[S30] M. Staats and C. Pasareanu, "Parallel Symbolic Execution for Structural Test Generation," in *Proc. of 19th International Symposium on Software Testing and Analysis*, 183−194, 2010.

[S31] W.-T. Tsai, P. Zhong, J. Balasooriya, Y. Chen, X. Bai, and J. Elston, "An Approach for Service Composition and Testing for Cloud Computing," in *Proc. of 10th International Symposium on Autonomous Decentralized Systems*, 631−636, 2011.

[S32] W.-T. Tsai, Y. Huang, and Q. Shao, "Testing the Scalability of SaaS Applications," in *Proc. of International Conference on Service-Oriented Computing and Applications*, 1−4, 2011.

[S33] T. Vengattaraman, P. Dhavachelvan, and R. Baskaran, "A Model of Cloud Based Application Environment for Software Testing," *International Journal of Computer Science and Information Security*, 7(3):257−260, 2010.

[S34] J. Wu, C. Wang, Y. Liu, and L. Zhang, "Agaric — A Hybrid Cloud Based Testing Platform," in *Proc. of International Conference on Cloud and Service Computing*, 87−94, 2011.

[S35] L. Yu, W.-T. Tsai, X. Chen, L. Liu, Y. Zhao, L. Tang, and W. Zhao, "Testing as a Service over Cloud," in *Proc. of 6th International Symposium on Service-Oriented System Engineering*, 181−188, 2010.

[S36] L. Yu, X. Li, and Z. Li, "Testing Tasks Management in Testing Cloud Environment," in *Proc. of 35th IEEE Annual International Computer Software and Applications Conference*, 76−85, 2011.

[S37] P. Zech, "Risk-based Security Testing in Cloud Computing Environments," in *Proc. of 4th IEEE International Conference on Software Testing, Verification and Validation*, 411−414, 2011.

[S38] L. Zhang, Y. Chen, F. Tang, and X. Ao, "Design and Implementation of Cloud-based Performance Testing System for Web Services," in *Proc. of 6th International ICST Conference on Communications and Networking in China*, 875−880, 2011.