Practitioner Evaluations on Software Testing Tools

Päivi Raulamo-Jurvanen
M3S, University of Oulu
Oulu, Finland
paivi.raulamo-jurvanen@oulu.fi

Simo Hosio
UBICOMP, University of Oulu
Oulu, Finland
simo.hosio@oulu.fi

Mika V. Mäntylä
M3S, University of Oulu
Oulu, Finland
mika.mantyla@oulu.fi

ABSTRACT

In software engineering practice, evaluating and selecting the software testing tools that best fit the project at hand is an important and challenging task. In scientific studies of software engineering, practitioner evaluations and beliefs have recently gained interest, and some studies suggest that practitioners find beliefs of peers more credible than empirical evidence. To study how software practitioners evaluate testing tools, we applied online opinion surveys (n=89). We analyzed the reliability of the opinions utilizing Krippendorff's alpha, intra-class correlation coefficient (ICC), and coefficients of variation (CV). Negative binomial regression was used to evaluate the effect of demographics. We find that opinions towards a specific tool can be conflicting. We show how increasing the number of respondents improves the reliability of the estimates measured with ICC. Our results indicate that on average, opinions from seven experts provide a moderate level of reliability. From demographics, we find that technical seniority leads to more negative evaluations. To improve the understanding, robustness, and impact of the findings, we need to conduct further studies by utilizing diverse sources and complementary methods.

KEYWORDS

opinion survey, software testing tool, tool evaluation, reliability

1 INTRODUCTION

Software projects face demands for delivering high-quality software at top speed. At the same time, there is the pressure for cost reduction. Test automation can be the solution but only after the problem of finding the right tool(s) has been solved. Therefore, selecting the correct tool(s) is important for profitable high speed and high quality testing. However, there are hundreds of commercial and open source tools available for software testing. Finding the right tool(s) even for evaluation and comparison can be challenging.

When faced with such choices practitioner often turn to fellow practitioners. In fact, in the context of software process improvement, it has been shown that practitioners prefer the opinions of their equals over empirical evidence [43]. There is no plausible evidence suggesting the situation would be any different for test tool selection. It can be questioned whether such beliefs are uniform or credible, in general.

Our goal is to study how experts evaluate quality attributes of popular software testing tools, to assess whether such expert advice can be trusted or not, and to study the effect of background (demographic) variables. We set to answer the following research questions:

RQ1 Do survey respondents agree or have consistent opinions on the criteria?

RQ2 How do background variables affect the survey evaluations (response variable)?

As a contribution, we show that increasing the number of respondents improves the reliability of the estimates measured with ICC, but the number of experts required for reliable evaluations is rather small.

2 BACKGROUND

We identified three relevant branches of prior work regarding our study: software test tool selection in Section 2.1, surveys of developers' opinions in Section 2.2 and assessment of responses in Section 2.3. In the following, we present a brief overview to these fields.

2.1 Software Test Tool Selection

Software test tool selection can be seen as a special case of software tool selection. Test automation, where tools play an integral part, can be considered as a solution to save (testing) costs and to improve quality and speed in software development [15]. Software testing tools impact the work of professionals across an organization. For a software testing tool to work in an organization, there are interconnections that need to be checked during evaluations [39].

Core capabilities of tools can be helpful in evaluation and selection of suitable tools [32]. However, challenges and obstacles in software testing are reported to be related not only to lack of time and resources, but also to lack of tools [13, 32, 41]. Costs have been reported to be among the topmost barriers to the use of automated testing tools [16, 17]. Despite the proliferation of practically free open source tools, the inevitable barrier of costs has not disappeared. Testing budgets are expected to continue to consume a big proportion of the overall budgets [6, 7].

For tool selection, there are different, more or less commercial comparison matrices available, e.g., [3, 20, 40]. Such sources may be useful for identifying tools, but the contents are neither generalizable nor validated for tool selection. There are software testing related academic studies which rely on surveys as the key methodology, e.g., [8, 11, 13, 16, 24, 32, 41, 44], but only a few report software test tools (used by the practitioners) by name (e.g., [8, 16, 17]).

In grey literature, test tool evaluations tend to propose and include tasks like live trials, proof-of-concepts and demos [45]. Such tasks require resources and competence, and are considered to bear the risk of wrong decisions [39]. Thus, investigating solutions and methodologies to help making sense of the software testing tools is topical and warranted.

2.2 Developers' Beliefs and Opinion Surveys

Passos et al. [37] and Devanbu et al. [10] conclude that people are influenced by strong beliefs obtained from personal experiences

rather than from empirical research. Similarly, Rainer et al. [43] present that for software process improvement, software practitioners find local opinion more credible knowledge than empirical evidence. Test tool automation consultation has been claimed to be the service most required from external consultants [23]. Beliefs may be the triggers for initiatives to adopt new technologies or tools, but the decisions are be based on opinions of experts [37]. Pano et al. [36] found social influence as an important factor in the process of adopting the best JavaScript framework, while prior research has little meaning to the practitioners.

Opinion survey is a common means of gauging, describing the public's collective sentiment for some defined need [14]. Online opinion surveys have emerged as a promising complementary way for understanding the collective public opinion [22]. Hosio et al. [21] have developed a light-weight decision support tool for surveying large pools of users for subjective opinions on how a given solution fairs in light of various criteria. The data can then be modeled for answers that best match a desired criteria configuration. Such a system is based on the concept of the *wisdom of the crowds* [48].

2.3 Assessment of Responses

To evaluate software testing tools, we need collective information, knowledge from people having invested time in choosing and using tools [44]. Kitchenham et al. [26] define a survey as a "*comprehensive system for collecting information to describe, compare or explain knowledge, attitudes and behavior*", and representativeness of responses can be justified by analyzing the demographics of the respondents [26].

In software engineering (SE), there are studies reporting low values for expert agreement/reliability using Kirppendorff's alpha and/or ICC, by e.g., Borg et al. [4], Anvaari et al. [1] and Kitchenham et al. [27]. Evaluations depend on the interpretation of a construct under study, i.e., include some degree of subjectivity [5, 47]. Inter-rater reliability is always specific to a given setting, i.e., respondents, instrument and time [5, 47]. Yet, Libby and Blashfield [33] claim that a small group of experts can provide as accurate evaluations as a large group.

3 METHODOLOGY

Section 3.1 explains our opinion survey. In section 3.2.1, we reason the importance of studying outlier values. Sections 3.2.2- 3.2.4 provide explanations of *Krippendorff's alpha* [30] as a measure for the agreement among observers (respondents), *intra-class correlation (ICC)* as a measure of reliability of evaluations, and *coefficient of variation (CV)* that we use to evaluate agreement. In sections 3.2.5 and 3.2.6 we describe the approaches to study the effect of the number of respondents on the accuracy of the evaluations, and the effect of demographics on tool evaluations, respectively.

3.1 Opinion Surveys

We constructed a survey questionnaire including questions about background information and 15 questions for evaluating criteria (on different selected tools of choice), see Table 1. We used the criteria for the survey from a set of characteristics considered important by practitioners in test tool selection [44, 45] and resting on the ISO/IEC 25010[1] quality model.

The criteria to be evaluated were: (1) *Applicability* (2) *Compatibility* (3) *Configurability* (4) *Cost-Effectiveness* (5) *Costs* (6) *Cross–Platform Support* (7) *Easy to Deploy* (8) *Easy to Use* (9) *Expandability* (10) *Further Development* (11) *Maintenance of Test cases & Data* (12) *Performance* (13) *Popularity* (14) *Programming Skills* and (15) *Reporting Features*.

The respondents were able to select one or more tools and evaluate the criteria of choice for each tool, one tool at a time. The list of tools (100) was created from a set of tools identified by practitioners for software testing [44]. The respondents could indicate the basis of their evaluations for the tool(s), i.e., whether those were based on personal experience *using* the tool, or on a *generic opinion*, e.g., from observing others using the tool. The criteria were evaluated on a scale from 0 to 10, at intervals of 0.5 (the default value being 5) and using a slider as the UI input element. The online opinion survey method used was adopted from the studies of Hosio et al. [21] and Goncalves et al. [18]. Both the questionnaire and the survey tool were validated by the authors and by an industry partner.

Survey#1 was published online August 29th, 2016. First, we promoted the survey to Finnish software testing professionals in a testing assembly in Finland, then posted a link to the survey (to selected groups) in Twitter, LinkedIn and Reddit, and sent a link to the survey to the public e-mail list of a testing association in Finland. We received 21 (of the 48 unique) responses with useful data (60 tool evaluations for 30 tools), and decided to harness survey#2.

For survey#2, we utilized the same online tool, but with clear focus to ensure fair amount of valid responses, at least for one tool. We contacted a number of practitioners from a set of Finnish collaborating companies in the EUREKA ITEA3 TESTOMAT[2] research project. The selected practitioners were known to be either familiar with *Robot Framework*[3], an open source, "*generic test automation framework for acceptance testing and acceptance test-driven development (ATDD)*" (as having used the tool and/or participated in the development of the tool), or the tool was utilized in their company. Survey#2 focused on Robot Framework, but the respondents were requested to evaluate other tools, too.

Survey#2 was published on March 1st, 2018. We promoted it by e-mail to seven professional software consultants (from six companies), asking them to distribute the link to their colleagues considered relevant for answering the questions. Similar approach, aka *snowball* or *chain sampling* [38], has been used by e.g., Ågerfalk and Fitzgerald [42]. To reach a wider audience, the survey was promoted in Robot Framework Slack and in Twitter with hashtag *robotframework*. Survey#2 was was open for a month. We received 68 (of the 80 unique) responses with useful data (101 tool evaluations for 17 tools). All collected data for both surveys are anonymous. See the study related material in Appendix A.

3.2 Methods for Analyzing the Data

3.2.1 Outliers in the Data. Tukey [49] has proposed a rule of thumb for detecting distant values, i.e., outliers on the basis of the quartiles

[1] iso25000.com/index.php/en/iso-25000-standards/iso-25010
[2] https://itea3.org/project/testomatproject.html
[3] http://robotframework.org/

Table 1: Questions in the Survey Tool

Question identifiers: ***B***=*Background*, ***C***=*Criterion* and **RFW**=Robot Framework (See the questionnaire in A)			
B1	Which option best describes your primary work area?	C4	Easy / intuitive to use.
B2	Which option best describes your role?	C5	Usage of the tool does not require programming skills.
B3	Years in your current role.	C6	Reporting features of the tool for testing results.
B4	Years in the Software Industry.	C7	Possibility to configure the tool for the needs.
B5	Country where you work.	C8	Possibility to remold or expand the tool.
B6	The domain of business you work on	C9	Cross-platform support.
B7	Describe the software or system the company or organization you work for is developing.	C10	Maintenance and re-use of test cases & test data.
B8	List the programming languages you use in your work.	C11	Active further development of the tool.
B9	My answers are based on a) personal experience (using the tool) b) personal conception (e.g., observing others use it)	C12	Popularity of the tool.
C1	Compatibility with existing tools (e.g., CI-tools).	C13	Low cost price or licensing of the tool (expected costs for acquisition and usage).
C2	Applicability of the tool to the tasks, methods & processes	C14	Performance of the tool (e.g. speed) for its purpose.
C3	Easy to deploy (initial effort to take the tool into use).	C15	Cost-effectiveness.
RFW	Question included in survey#2 only. About Robot Framework: Have you been involved in the development of the core tool and/or related libraries?		a) No b) Yes, to the development of the core tool c) Yes, to the development of related libraries d) Yes to both, development of the tool and libraries.

of the data. Tukey [49] defined an outlier as a value more than 1.5 times the interquartile range (IQR, i.e., $Q3 - Q1$)) from the quartiles, i.e., either below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$. Osborne and Overbay [35] and Chandola et al. [9] emphasize the importance of studying the outliers, as those may have real life relevance [9], and include relevant information. We intended to study outliers in the data to see if some criteria for a tool have more outliers than others. Outliers may be a sign of nuisance, error or legitimate data, but can also be *"inspiration for inquiry"* [35].

3.2.2 Krippendorff's alpha. Krippendorff's alpha (α) is a statistical measure for determining inter-rater reliability. The values for the α range from perfect disagreement (0) to perfect agreement (1). The values $\alpha \geq 0.800$ are suggested for drawing reliable conclusions while values $0.667 \leq \alpha < 0.800$ are claimed for tentative conclusions only [29].

We used the R-function kripp.alpha[4] to measure the level of agreement among the respondents (raters) on the criteria (subjects) of the top 6 most evaluated tools. We considered the level of measurement for the data to be *ratio*, since the possible values (from 0 to 10 at intervals of 0.5, i.e., 21 levels) were ordered units having the same difference and an absolute zero. As the values were limited to our scale, the α values were calculated for *ordinal* type of data, too.

3.2.3 Intra-class Correlation Coefficient (ICC). ICC is a common statistics used for measuring inter-rater reliability for ratio type of data [19]. As for Krippendorff's α, the values for ICC vary between 0 and 1, higher values indicating greater reliability. The commonly referenced ICC values are ≥ 0.90 for excellent, $0.75 \leq$ and < 0.90 for good, $0.50 \leq$ and < 0.75 for moderate and < 0.5 for poor agreement [28].

We used the R-function ICC [5] to estimate the association among the respondents for the top 6 tools. The function provides results for six different forms, presented as two numbers, i.e., ICC(x,y) or ICCxy. The first number (x) indicates the *model* (1, 2 or 3) and the second (y) the *type* of the measurement protocol (either "1" as a single rater/measurement, or "k" as the mean of k respondents/measurements) [46]. As the results may differ and lead to different interpretations, it is suggested to report both the results and the computational variant [19, 28]. To select the correct form [46], we analyzed the prerequisites suggested by Koo and Li [28]:

(1) *Do we have the same set of respondents for all criteria?* Yes, the same set of respondents evaluated all criteria.
(2) *Is the sample of respondents randomly selected from a larger population or is it a specific sample of respondents?* We had a specific sample of respondents, a convenience sample [25]. The respondents evaluated the same criteria, but the underlying contexts and constructs may vary for samples (even for respondents). Thus, there is no intention to generalize the tool related results regarding the values as such, but to analyze reliability of responses.
(3) *Are we interested in the reliability of a single respondent or the mean value of multiple respondents?* We were interested in reliability of the mean value of many respondents.
(4) *Are we concerned about consistency or agreement?* We wanted to check consistency (not absolute agreement).

Thus, the first two questions are used to guide the selection of the *model*. The third question is about the *type*, whether the measurement protocol will be conducted by applying "single respondent" or "mean of k respondents". The last question is about the difference of the purpose.

We measured ICC using a two-way mixed effects, average measures for consistency, i.e., ICC(3,k) [46] with the purpose to estimate the degree the respondents provided consistency in the evaluations across the criteria. (For ICC2 and ICC3 the difference is the consideration of respondents as random or fixed effects). In reporting the

[4] www.rdocumentation.org/packages/irr/versions/0.84/topics/kripp.alpha
[5] https://cran.r-project.org/web/packages/psych/psych.pdf

results, we followed the guidelines suggested by Hallgren [19] and Koo and Li [28]. In cases where the single measured ICC's are low (ICC2) and average measured ICC's (ICC3) are high, it is suggested to report both cases to demonstrate the discrepancy [46].

3.2.4 Coefficient of Variation (CV). We measured the coefficient of variation (CV) for the criteria evaluations for the top 6 tools, to analyze the extent of variability in evaluations in relation to the mean of the population. Practically, the lower the CV the less variation there exists. As our criteria are very different of nature (e.g., some more human oriented than others like "Programming Skills" and "Costs"), CV's allow to compare the variation across different criteria having different means.

As our data was considered to be of type *ratio*, but was limited to our scale (values from 0 to 10 at intervals of 0.5, i.e., 21 levels), we calculated the CV for both *ratio* and *ordinal* type of data. For *ratio* type of data the CV was calculated as the ratio of the standard deviation to the mean (1). For calculating the CV for the *ordinal* type of data we used the formula (2) presented by Kvålseth [31].

(1) CV for *ratio* type of data

$$CV = \sigma/\overline{x} = sqrt(var(x))/mean(x) \quad (1)$$

(2) CV for *ordinal* type of data, as in Kvålseth [31].

$$\Delta = \sum_{i<j}\sum |i-j|P_iP_j$$

$$\Delta^\star = [4/(k-1)]\Delta$$

$$CV = 1 - (1-\Delta^\star)^{1/2} \quad (2)$$

3.2.5 Number of Respondents for ICC. To analyze the effect of the number of respondents to the incremental accuracy of tool evaluations, we applied the example modeled by Libby and Blashfield [33]. They empirically tested the effects of group size in decision making, and concluded that on average, having three accurate judges could improve average performance (in most cases). Employment of a small number of judges would be practical and cost efficient [33].

We generated random sets of respondents (from 2 to n respondents, n being the total number of respondents for a tool, see Table 2) for each top 6 tools. For each size of sets (from 2 to n) we run 100 iterations of ICC (each run with a new random set of respondents) with intention to compare the medians of the groups to the common ICC reference values [28]. Thus, the total number of ICC values for the tools ($(n-1)*100$) were 400 for Appium (n=5), 900 for Jenkins (n=10), 300 for Jira (n=4), 400 for JMeter (n=5), 7600 for Robot Framework (n=77) and 400 for Selenium (n=5).

3.2.6 Effect of the Demographics. For studying the effect of demographics on the evaluations, we carried out a negative binomial regression analysis (for modeling count variables) with R-function glm.nb[6]. We used an automatic method, R-function stepAIC[7] to analyze proposed variable selection. For the baseline model, we included seven variables: familiarity with Robot Framework (see the question ID *RFW* in Table 1), experience regarding the use of the tool, years in the current role and in the work area, type of role and work area, and business domain.

[6]https://stat.ethz.ch/R-manual/R-patched/library/MASS/html/glm.nb.html
[7]https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/stepAIC.html

Table 2: Top 6 Tools - Survey Data

Tool	Survey#1[1]		Survey#2[1]		Total[1]	
	Resp	Eval	Resp	Eval	Resp	Eval
Appium	2	23	3	45	5	68
Jenkins	5	75	5	75	10	150
Jira	3	45	1	15	4	60
JMeter	5	68	–	–	5	68
RFW	9	119	68	998	77	1117
Selenium	1	15	4	47	5	62

[1] Number of respondents and evaluations for a tool

4 RESULTS

The two surveys included evaluations for 2128 criteria, for 38 unique tools, in total. We filtered out any evaluations known to be test cases, duplicates or having only default values. The top 6 most evaluated tools in the surveys, namely Robot Framework, Jenkins, Appium, JMeter, Selenium and Jira, received 1525 evaluations, in total, see Table 2.

The arithmetic mean of evaluations for the criteria in the surveys for the top 6 tools, are shown in Table 4. The fact that practitioners tend to perceive Jira as a tool for software testing seemed rather reasonable, the tool being part of a whole, "*Bringing testing capabilities within Jira helps tightly integrate product management, development, and testing to streamline efficiency and productivity.*"[8].

In both surveys, Robot Framework was the most evaluated tool, see Table 2. That is expected to be a by-product of two obvious facts: 1) Robot Framework as "a local tool" among the respondents (majority working in Finland) and 2) the utilization of convenience sampling [25] for survey#2. The respondents ($n = 89$) reported the country they work in as Australia (3), Brazil (2), Canada (5), Czech Republic (1), Finland (55), India (4), Israel (1), Portugal (2), Russia (1), Spain (1), The Netherlands (4), United Kingdom (2) and USA (5). Three respondents did not provide that information. See background details in Table 3.

4.1 RQ1 - Opinions of the Criteria

To answer the RQ1 *"Do survey respondents agree or have consistent opinions on the criteria?"*, we analyzed the top 6 tools, see Table 4. We intended to identify the criteria that require focusing or investing in, and to analyze the reliability of the data.

Robot Framework had a total of 1117 evaluations (about 52% of all evaluations) by 77 respondents, see Table 2. When analyzing the boxplot for Robot Framework, the median value was ≥ 80.0 for all other criteria, except for *Popularity* and *Programming Skills*. Those criteria also had the highest variance and the lowest lower quartile values (60.0 and 40.0, respectively). The criterion having the smallest IQR for Robot Framework was *Costs* ($100 - 91.25 = 8.75$), while the largest IQR was 45 ($85 - 40$) for *Programing skills*.

The evaluations for the top 6 tools included 62 outlier values, see Table 4. There were no outliers for Appium and Jira, just one for Selenium (2%), two for JMeter (3%), four for Jenkins (3%) and 55 (5%) for Robot Framework. Those outlier values were given by 27 unique

[8]https://marketplace.atlassian.com/categories/test-management

Table 3: Respondents' Background (n = 89)

Number of Respondents in Work Areas	Software Development	19	21.3%
	Software Testing	55	61.8%
	Requirements Mgmt	1	1.1%
	Project Mgmt	4	4.5%
	Not specified, NA	10	11.2%
Number of Respondents in Roles	Individual Contributor	37	41.6%
	Specialist	25	28.1%
	Lead	16	18.0%
	Executive	6	6.7%
	Not specified, NA	5	5.6%
Experience in Years	Max (Min)	45	(0)
	Avg (Median)	12.9	(11.0)
Years in Current Role	Max (Min)	24.0	(0)
	Avg (Median)	3.8	(3.0)
Basis of Evaluations	Experience	1979	93.0%
	Opinion	149	7.0%

respondents (30% of all respondents) with years of experience, i.e., on average 14.1 years in the industry (median 12). The number of years in the current role was on average 4.0 years (median 3). The respondents were inclined to evaluate the tools critically.

The measurements for Krippendorff's α resulted in low values, see Table 6. Although the criteria being evaluated were the same, the evaluation of a criterion for a tool is a factor of some specific context, underlying construct for a tool and level of experience of an expert. Krippendorff's α was not considered as the best measure in our case as there was a wide range of possible values, and the evaluations were based on personal perceptions and experiences. Thus, we did not expect all respondents to interpret the criteria the same way. In fact, Dybå et al. [12] emphasize that "*seemingly unpatterned and disagreeing findings from quantitative studies may have underlying consistency when omnibus context is taken into account*".

As suggested by Shrout and Fleiss [46], we report ICC for single measured and average measured values for both random and fixed effects, i.e, ICC(2,1) & ICC(2,k) and ICC(3,1) & ICC(3,k), see Table 5. The resulting ICC(3,k), i.e., average consistency among fixed respondents varied between 0.60 for Selenium and 0.94 for Robot Framework. The ICC(3,k) for Robot Framework was in the "good" range (although the value 0.94 was in the "excellent" range [28]), as there is 95% chance the value will be in between 0.881 and 0.97 (in the worst case the ICC value would be considered "good"). The absolute agreement, as Krippendorff's alpha, was low for Robot Framework, 0.12 and 0.16 for single measured random and fixed effects, i.e., ICC(2,1) and ICC(3,1), respectively.

As the criteria had different means, we measured the coefficient of variation (CV) for the evaluations of top 6 tools, see Table 6. The calculated CV's indicate strong positive correlation (Pearson's correlation), although the CV's for ordinal data were slightly better, in general. For Robot Framework, *Cross-Platform Support*, *Cost-Effectiveness* and *Costs* had the lowest uncertainty (the lower the value the more precise the estimate). Thus, in the case of Robot Framework, it would be beneficial to study issues related to *Programming Skills*, *Popularity*, *Easy to Deploy* and *Maintenance of Test Cases & Data* in more detail.

For analyzing the number of experienced practitioners for improving the accuracy of tool evaluation, we run iterations, as described in Section 3.2.5. We used the values of 0.5 ("moderate"), 0.75 ("good") and 0.9 ("excellent") as threshold values for indicating the levels of reliability [28] for the group medians. For Robot Framework, *7* respondents were required to get to the "moderate" level of reliability, *16* to get to the "good" and *47* for "excellent" level of reliability, see Figure 1. For Appium, Jenkins, Jira and JMeter, the "moderate" level of reliability was reached with *3* respondents while for Selenium with *4*. For Jenkins the combination of *7* respondents reached "good" level of reliability. The medians for the other four tools (Appium, Jira, JMeter and Selenium) did not reach either "good" or "excellent" level, indicating a need for more respondents.

4.2 RQ2 - Background of the Respondents

The RQ2 covered the effect of the background of the respondents to the evaluations: *"How do background variables affect the survey evaluations (response variable)?"*. The results for evaluations for all tools (n = 2128 evaluations) are shown in Table 7.

We carried out a negative binomial regression to analyze the effect of demographics on evaluations. To select a subset of the explanatory variables, we used model simplification as described in Section 3.2.6. The proposed best model included four variables: experience regarding the use of the tool, familiarity with Robot Framework (see the question ID *RFW* in Table 1) and years in the work area and in the current role. However, we decided to keep all original seven variables, see Table 7.

The background variables were not expected to make a very accurate model, as the respondents rated their personal experiences related to a tool, and there were evaluations for different tools. In fact, the missingness information about the model indicates that there were 745 partial observations, i.e., not including all required data, and those were not used in fitting the model. The AIC measure of variance is 12710, but uninformative as we have just one model. Deviance residuals indicate our model is not biased in one direction (1Q (−0.4737), 3Q (0.5586) and median (0.0893)).

The respondents reported the basis of their evaluations, i.e., either *experience* (personal experience *using* the tool, 0) or *opinion* (*generic opinion* e.g., from observing others using the tool, 1). An opinion based evaluation is significantly associated with a decrease of 0.1349 in evaluation $n = 149, r(1358) = -0.1349, p = 0.0001$, compared to one based on experience using the tool (n = 1979). Regarding the familiarity with Robot Framework, the baseline is "NA". The factor "No", i.e., the evaluations of those respondents that had not contributed to the development of the tool, is significant $n = 629, r(1358) = 0.0788, p = 0.0014$ with respect to the baseline.

The coefficients for role implies that given all other variables were constant, an evaluation of an individual contributor would be expected to be −0.0600 less than evaluation for baseline (executive role), i.e., $n = 999, r(1358) = -0.0600, p = 0.1855$. Similarly, the categorial variables lead and specialist have impact with respect to the baseline, as for a lead the values are $n = 331, r(1358) = -0.0469, p = 0.2830$ and for a specialist $n = 617, r(1358) = -0.1018, p = 0.0195$.

Table 4: Top 6 Tools - Survey Evaluations, Number of outlier values & Rank of criteria

Rank	Criteria	Appium		Jenkins		Jira		JMeter		Robot Fw[a]		Selenium	
		Eval.[b]	O.[c]	Eval.	O.	Eval.	O.	Eval.	O.	Eval.	O.	Eval.	O.
6	Applicability	66.0^{6}	0	84.0^{5}	1	83.8^{5}	0	79.0^{6}	0	83.0^{4}	5	62.0^{7}	0
8	Compatibility	52.0^{9}	0	84.5^{4}	0	77.5^{6}	0	71.7^{12}	0	81.8^{5}	4	56.0^{9}	0
7	Configurability	70.0^{5}	0	81.0^{8}	0	77.5^{6}	0	82.0^{5}	1	80.0^{7}	2	68.3^{5}	0
4	Cost-Effectiveness	61.3^{7}	0	86.5^{2}	0	61.3^{13}	0	83.8^{3}	0	89.4^{2}	5	76.3^{3}	0
1	Costs	84.0^{1}	0	86.0^{3}	1	47.5^{15}	0	93.0^{1}	0	92.5^{1}	8	72.5^{4}	0
5	Cross-Platform Support	77.5^{3}	0	83.5^{6}	1	85.0^{4}	0	73.0^{10}	0	83.7^{3}	1	66.3^{6}	0
9	Easy To Deploy	39.0^{13}	0	70.5^{12}	0	57.5^{14}	0	84.0^{2}	0	79.4^{9}	8	51.0^{12}	1
9	Easy To Use	45.0^{11}	0	60.0^{15}	0	73.8^{9}	0	77.5^{9}	0	78.9^{10}	6	56.3^{8}	0
13	Expandability	46.3^{10}	0	78.5^{9}	0	73.8^{9}	0	59.0^{14}	1	78.4^{11}	2	38.8^{15}	0
3	Further Development	78.8^{2}	0	82.0^{7}	0	86.3^{3}	0	78.8^{8}	0	80.1^{6}	2	82.5^{1}	0
12	Maintenance of TC&D	53.8^{8}	0	72.0^{11}	0	65.0^{11}	0	72.0^{11}	0	76.8^{13}	3	53.8^{11}	0
11	Performance	36.0^{14}	0	77.0^{10}	1	65.0^{11}	0	79.0^{6}	0	78.2^{12}	3	55.0^{10}	0
1	Popularity	72.0^{4}	0	89.0^{1}	0	87.5^{1}	0	83.3^{4}	0	70.4^{14}	1	82.5^{1}	0
15	Programming Skills	27.5^{15}	0	63.5^{13}	0	87.5^{1}	0	50.0^{15}	0	62.4^{15}	5	43.8^{13}	0
13	Reporting Features	42.5^{12}	0	63.0^{14}	0	75.0^{8}	0	66.0^{13}	0	79.8^{8}	0	43.8^{13}	0
	Total # of outliers		0		4		0		2		55		1

[a] Robot Framework
[b] Arithmetic mean of the evaluations for a criterion. The superscript is the ranking of the criterion.
[c] Number of outlier values in the data for a criterion

Table 5: Intraclass Correlation Coefficients, Call: ICC(x = data, missing = TRUE, alpha = 0.05)

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.34	3.9	14	56	0.00015	0.13	0.62
ICC3	0.36	3.9	14	56	0.00015	0.14	0.65
ICC2k	0.72	3.9	14	56	0.00015	0.43	0.89
ICC3k	0.74	3.9	14	56	0.00015	0.45	0.90
Number of criteria = 15 Number of respondents = 5							

(a) Appium

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.21	5.8	14	126	1.2e-08	0.083	0.44
ICC3	0.33	5.8	14	126	1.2e-08	0.163	0.58
ICC2k	0.72	5.8	14	126	1.2e-08	0.475	0.89
ICC3k	0.83	5.8	14	126	1.2e-08	0.661	0.93
Number of criteria = 15 & Number of respondents = 10							

(b) Jenkins

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.23	3.0	14	42	0.0031	0.034	0.53
ICC3	0.33	3.0	14	42	0.0031	0.082	0.63
ICC2k	0.55	3.0	14	42	0.0031	0.123	0.82
ICC3k	0.66	3.0	14	42	0.0031	0.263	0.87
Number of criteria = 15 & Number of respondents = 4							

(c) Jira

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.136	3.2	14	56	0.00097	0.012	0.37
ICC3	0.306	3.2	14	56	0.00097	0.094	0.60
ICC2k	0.440	3.2	14	56	0.00097	0.058	0.75
ICC3k	0.688	3.2	14	56	0.00097	0.341	0.88
Number of criteria = 15 & Number of respondents = 5							

(d) JMeter

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.12	16	14	1064	1.6e-35	0.064	0.26
ICC3	0.16	16	14	1064	1.6e-35	0.088	0.33
ICC2k	0.91	16	14	1064	1.6e-35	0.840	0.96
ICC3k	0.94	16	14	1064	1.6e-35	0.881	0.97
Number of criteria = 15 & Number of respondents = 77							

(e) Robot Framework

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.20	2.5	14	56	0.0073	0.0300	0.48
ICC3	0.23	2.5	14	56	0.0073	0.0375	0.53
ICC2k	0.55	2.5	14	56	0.0073	0.1340	0.82
ICC3k	0.60	2.5	14	56	0.0073	0.1632	0.85
Number of criteria = 15 & Number of respondents = 5							

(f) Selenium

Table 6: Top 6 Tools - Krippendorff's α & Coefficients of Variation

		Appium		Jenkins		Jira		Jmeter		Robot Fw[a]		Selenium	
		Ordin.[b]	Ratio[c]	Ordin.	Ratio	Ordin.	Ratio	Ordin.	Ratio	Ordin.	Ratio	Ordin.	Ratio
	Krippendorff's α	0.294	0.208	0.173	0.113	0.224	0.069	-0.07	0.076	0.127	0.086	0.15	0.044
Coefficients of Variation	Applicability	0.13	0.18	0.11	0.13	0.06	0.08	0.20	0.23	0.18	0.20	0.26	0.36
	Compatibility	0.20	0.34	0.11	0.12	0.12	0.16	0.35	0.46	0.18	0.19	0.14	0.24
	Configurability	0.13	0.18	0.12	0.13	0.04	0.06	0.18	0.20	0.20	0.21	0.17	0.30
	Cost-Effectiveness	0.08	0.14	0.13	0.14	0.20	0.32	0.10	0.13	0.15	0.15	0.10	0.15
	Costs	0.23	0.26	0.20	0.25	0.33	0.60	0.10	0.12	0.13	0.17	0.33	0.42
	Cross-Platform S.	0.23	0.27	0.19	0.22	0.06	0.08	0.31	0.37	0.14	0.14	0.30	0.41
	Easy To Deploy	0.24	0.55	0.21	0.26	0.35	0.57	0.17	0.19	0.22	0.24	0.18	0.32
	Easy To Use	0.22	0.43	0.27	0.38	0.17	0.26	0.24	0.30	0.19	0.22	0.13	0.23
	Expandability	0.28	0.58	0.17	0.19	0.13	0.18	0.29	0.44	0.22	0.23	0.25	0.62
	Further Devel.	0.22	0.27	0.12	0.14	0.07	0.09	0.25	0.30	0.21	0.22	0.15	0.18
	Maintenenance	0.20	0.35	0.20	0.25	0.26	0.42	0.25	0.31	0.22	0.24	0.17	0.32
	Performance	0.27	0.65	0.18	0.21	0.19	0.31	0.19	0.22	0.20	0.22	0.18	0.31
	Popularity	0.25	0.31	0.09	0.09	0.06	0.07	0.17	0.21	0.27	0.30	0.11	0.14
	Programming Skills	0.35	1.16	0.34	0.43	0.06	0.07	0.43	0.71	0.35	0.41	0.30	0.68
	Reporting Features	0.28	0.59	0.25	0.32	0.19	0.27	0.40	0.50	0.21	0.23	0.38	0.75
	Pearson's Corr.	$r(13) = 0.85$		$r(13) = 0.99$		$r(13) = 0.99$		$r(13) = 0.96$		$r(13) = 0.98$		$r(13) = 0.87$	
	P-value	$6.497e-05$		$3.33e-12$		$9.534e-14$		$6.358e-09$		$8.522e-11$		$2.314e-05$	

[a] Robot Framework
[b] *Ordinal* level of measurement, see calculation for CV in Section 3.2.4.
[C] *Ratio* level of measurement, see calculation for CV in Section 3.2.4.

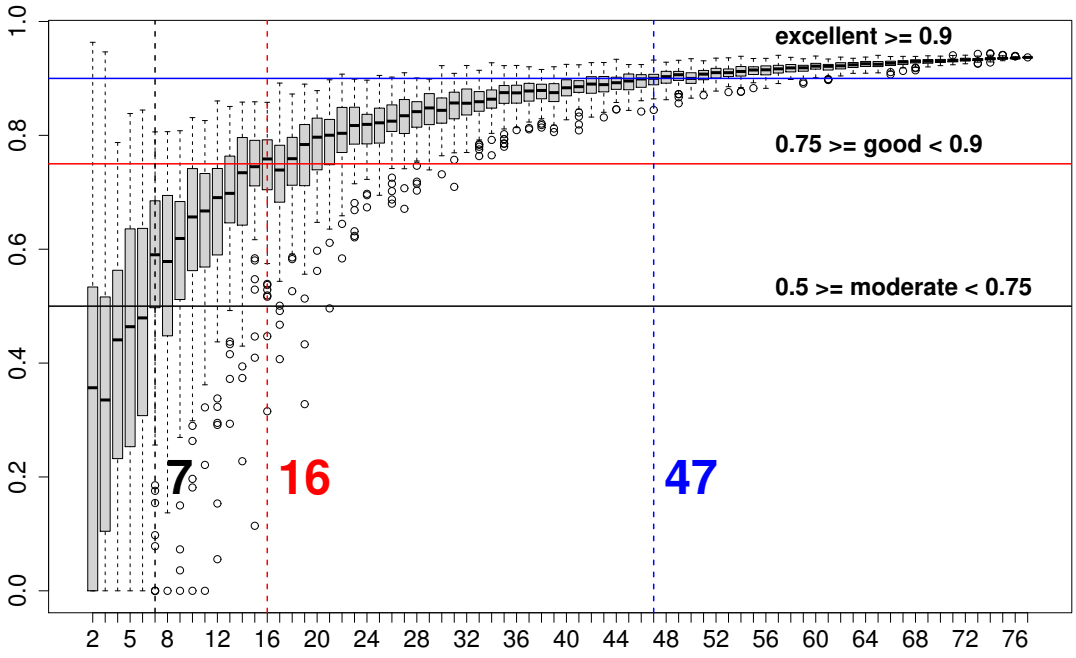

Figure 1: Robot Framework (n=77), Analysis of the number of respondents with ICC for groups of size 2-77.

Table 7: Regression Coefficients (all data, $n = 2128$)

Coefficients	Est.	Std.Error	z	Pr(>|z|)
(Intercept)	4.4283	0.0831	53.26	0.0000
Familiarity with Robot Framework,[a] see ID *RFW* in Table 1				
No[n=629]	0.0788	0.0247	3.19	0.0014
Yes-Both[n=60]	0.0996	0.0540	1.85	0.0649
Yes-Core[n=57]	0.0322	0.0581	0.55	0.5802
Yes-Lib[n=207]	0.0807	0.0311	2.59	0.0096
familiarity[b]	-0.1349	0.0339	-3.98	0.0001
YearsInRole	0.0137	0.0028	4.88	0.0000
YearsInWA	-0.0036	0.0015	-2.32	0.0201
Role[a, c]				
Individual Contrib.[n=999]	-0.0600	0.0453	-1.32	0.1855
Lead[n=331]	-0.0469	0.0437	-1.07	0.2830
Specialist[n=617]	-0.1018	0.0436	-2.34	0.0195
Work Area[a, d]				
Requirements Mgmt[n=2]	0.3918	0.2280	1.72	0.0858
SW Development[n=472]	0.0803	0.0457	1.76	0.0791
SW Testing[n=1212]	0.0814	0.0408	1.99	0.0463
Business Domain[a, e]				
Consulting[n=665]	-0.0423	0.0519	-0.82	0.4148
Consumer Products[n=40]	0.0783	0.0773	1.01	0.3114
Energy & Utilities[n=30]	0.0748	0.0770	0.97	0.3309
Financial Services[n=266]	-0.0743	0.0547	-1.36	0.1743
HealthCare & LifeSci.[n=75]	0.0275	0.0665	0.41	0.6792
Manufacturing[n=60]	0.2141	0.0671	3.19	0.0014
Media & Entertain.[n=58]	-0.0205	0.0664	-0.31	0.7582
Public Sector[n=43]	0.0123	0.0741	0.17	0.8681
Science & HighTech[n=134]	-0.0882	0.0582	-1.52	0.1294
Telecommunication[n=137]	-0.0216	0.0588	-0.37	0.7138
Transportation[n=60]	-0.1383	0.0651	-2.12	0.0338

[a] n = Number of entries in the data
Familiarity with RFW: NA (default), No, Yes-Both, Yes-Core, Yes-Lib
[b] familiarity: Experience=0 (n=1979), Opinion=1 (n=149)
[c] Role: NA (missing), Executive, Individual Contributor (default), Lead, Specialist
[d] Work Area: NA (default), Project Mgmt, Requirements Mgmt, SW Development, SW Testing
[e] For Domains Automotive (default) is not shown

(Dispersion parameter for Negative Binomial(11.5396) family taken to be 1)
Null deviance: 1706.2 on 1382 degrees of freedom
Residual deviance: 1535.2 on 1358 degrees of freedom
(745 observations deleted due to missingness)
AIC: 12710, Number of Fisher Scoring iterations: 1
Theta: 11.540, Std. Err.: 0.537
2 x log-likelihood: -12657.612

Years in the current role, $r(1358) = 0.0137, p = 0.0000$, is a more significant factor than years in the working area, $r(1358) = -0.0036, p = 0.0201$. The coefficients of the business domains factors "Manufacturing" ($n = 60, r(1358) = 0.2141, p = 0.0014$) and "Transportation" ($n = 60, r(1358) = -0.1383, p = 0.0338$) seem to signify the most positive and the most negative evaluations with some significance with respect to the baseline ("Automotive").

5 DISCUSSION

Regarding the RQ1, "*Do survey respondents agree or have consistent opinions on the criteria?*", we acknowledge that tool evaluations are context-specific, practitioner-related and conducted in retrospect to experiences [51]. As our surveys were anonymous, we could not ask the respondents to reason their evaluations, but just analyze the variability of the values for the criteria. As there were 21 options for each criterion and agreement requires absolute consistency, the low results for Krippendorff's alpha were not surprising. However, when analyzing the relative ordering of the ratings, deviations from the mean with ICC(3,k), the average consistency among the respondents for the top 6 tools was in the "moderate" or "good" level of reliability, in general, see Table 5.

Costs are considered as barriers to the use of automated testing tools [16, 17]. The top 6 tools were considered as low cost and cost-effective, in general, see Table 4. Wagenaar et al. [50] reported Scrum teams to prefer perceived usefulness over perceived ease of use after using a tool. Our findings from the surveys seem to support that observation as the rankings for e.g., *Applicability* (#6) or *Cost-Effectiveness* (#4) are higher than for *Easy to Use* (#9). Azizyan et al. [2] observed diversity of opinions in the form of conflicting comments for simple versus more comprehensive tools. Our findings from the surveys for the criterion of *Programming Skills* (high variability among respondents, see Table 6) support the former when considering programming skills as prerequisite for the use of a more comprehensive tool.

The CV's tend to be higher for *Programming Skills* than for the other criteria. Agreements on the criteria are important for confirming assumptions, but the disagreements (i.e., low values and outliers) are valuable for identifying possibly problematic issues. For example, low evaluations for criteria considered important in tool selection [44] are worth studying, in more detail. Our findings suggest that collective opinion can be used to point out issues, worth focusing on or investigating, in more detail.

Mannes et al. [34] consider expert knowledge as "*accurate, robust and appealing as a mechanism for helping individuals tap collective wisdom*". Our findings support the suggestion by Libby and Blashfield [33] that performance of a group as a function of the number of raters improves with a few accurate raters only. However, more raters may be required for minimizing the probability of making poor decisions [33]. Our findings from running ICC by pooling different combinations of respondents (raters) suggest that seven experienced respondents are enough for "moderate" level of reliability, but considerably more experienced respondents are required for "good" or "excellent" level of reliability. Thus, we find that trusting an opinion of just one or opinions of a few practitioners may be questionable or misleading, and can lead to wrong decisions.

To study the RQ2, "*How do background variables affect the survey evaluations (response variable)?*", we carried out a negative binomial regression, see Table 7. We observed the *opinion* based evaluations to be significantly lower than those based on *experience of using* the tool. Years of experience in the working area seems to be a factor having negative effect on the evaluations. The years in the current role, in turn, seems significant. The tools have been available only

for some time, e.g., JMeter[9] since 1998 but Appium[10] only since 2012. Nowadays, popular open source tools have active development communities. Technical seniority (e.g., having a specialist role), was a significant factor, specialists providing slightly more critical evaluations. Thus, the role of a respondent is predicted useful for similar types of surveys.

Earlier, expert tool users were not considered reliable for evaluating software testing tools, as they were not expected to have the experience or knowledge to make distinctions between various aspects of tools usage [39]. Nowadays, expert tool users can be active in the development of some open source tool(s) and thus, have in depth understanding of the functionality and possible special characteristics of such tool(s). Practitioners seem to value perceptions of local crowds as credible empirical evidence [18, 43]. Software testing tools are used in various business domains. However, studying tool evaluations in a single company or within a single domain could provide a limited view on the criteria. Anvaari et al. [1] reported that neither long experience in the area of interest nor the same domain of expertise provided agreement among raters.

6 THREATS TO VALIDITY

We followed the guidelines presented by Wohlin et al. [52] for evaluating the validity of the study. Regarding internal validity, we acknowledge the bias of the sampling techniques for the surveys (to reach experts from several organizations, to get a rich set of data, at least on one case tool). Threats to external validity are related to the small ($n = 89$) sample size.

As tool evaluations are construct and context specific, bound to time and experiences, the results are not generalizable as such. There is no single truth to confirm, but the results provide a basis for analyzing possible problematic perceptions. To address construct validity, the survey was piloted in advance. Based on the results (e.g., variance for evaluations of some criteria), the questionnaire would need to be refined for further studies. Thus, our results may be due to confounding variables not taken into account.

7 CONCLUSIONS AND FUTURE WORK

Tool evaluations are construct and context specific, and bound to time and experiences. Thus, opinions on software testing tools can be diverging or conflicting. Recollection of personal experiences is error-prone, but beliefs should be given attention in research to help to provide and to disseminate verified evidence to the practitioners [10]. Trusting on beliefs or perceptions of a small group of practitioners can be inaccurate or misleading. Therefore, perceptions and beliefs of practitioners should be analyzed with caution. We find it possible to harness realistic personal insights of the subject area into crowd-based insights.

We find that collective opinion, in the context of interest, is important in pointing out the criteria of importance or with polarized opinions worth investigating in more detail. According to our findings, experience based evaluations (on using a tool) seem to be more positive than those based on pure opinion (not having used a tool), and expert respondents tend to provide consistent evaluations for some criteria. However, some specific roles (with technical seniority like specialists) are highly significant providing negative evaluations. Practitioners with different background may not have consensus about evaluations but the differences how they apply the given scale may be predicted.

Our findings suggest that more than just three expert respondents are required to gain reliable evidence for testing tool evaluations. We conclude that on average, opinions from seven experts can provide reliable evidence for moderate level of accuracy. There is a need for practical and efficient ways for conducting tool evaluations that provide reliable empirical evidence for software practitioners. Considerably more work needs to be conducted for better understanding and for establishing more definitive, tool specific evidence.

[9] https://web.archive.org/web/20150419065701/http://projects.apache.org/projects/jmeter.html

[10] http://appium.io/history.html

ACKNOWLEDGMENTS

The work was partially supported by two research grants: No.: 3192/31/2017 for the EUREKA ITEA3 TESTOMAT project (16032) from Business Finland and No.: 286386 for the CPDSS project from the Academy of Finland.

REFERENCES

[1] Mohsen Anvaari, Carl-Fredrik Sørensen, and Olaf Zimmermann. 2016. Associating Architectural Issues with Quality Attributes: A Survey on Expert Agreement. In *Proccedings of the 10th European Conference on Software Architecture Workshops (ECSAW '16)*. ACM, New York, NY, USA, Article 11, 7 pages. https://doi.org/10.1145/2993412.3004847

[2] Gayane Azizyan, Miganoush K. Magarian, and Mira Kajko-Matsson. 2011. Survey of Agile Tool Usage and Needs. In *2011 Agile Conference*. IEEE, Salt Lake City, UT, USA, 29–38. https://doi.org/10.1109/AGILE.2011.30

[3] Bajaj Harsh, Infosys. 2015. Choosing the right automation tool. https://www.infosys.com/it-services/validation-solutions/white-papers/documents/choosing-right-automation-tool.pdf.

[4] Markus Borg, Iben Lennerstad, Rasmus Ros, and Elizabeth Bjarnason. 2017. On Using Active Learning and Self-training when Mining Performance Discussions on Stack Overflow. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE'17*. ACM, New York, NY, USA, 308–313. https://doi.org/10.1145/3084226.3084273

[5] Anne Bruton, Joy H Conway, and Stephen T Holgate. 2000. Reliability: What is it, and how is it measured. *Physiotherapy* 86, 2 (2000), 94–99. https://doi.org/10.1016/S0031-9406(05)61211-4

[6] Capgemini Consulting. 2015. World Quality Report 2014-2015, Sixth Edition, 2014. https://www.sogeti.com/explore/reports/world-quality-report-2014-2015/.

[7] Capgemini Consulting. 2016. World Quality Report 2015-2016, 2015. https://www.sogeti.com/explore/reports/world-quality-report-2015-2016/.

[8] Adnan Causevic, Daniel Sundmark, and Sasikumar Punnekkat. 2010. An Industrial Survey on Contemporary Aspects of Software Testing. In *2010 3rd International Conference on Software Testing, Verification and Validation*, Vol. 1. IEEE, USA, 393–401. https://doi.org/10.1109/ICST.2010.52

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

[10] Prem Devanbu, Thomas Zimmermann, and Christian Bird. 2016. Belief & Evidence in Empirical Software Engineering. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*. ACM, New York, NY, USA, 108–119. https://doi.org/10.1145/2884781.2884812

[11] Arilo Claudio Dias-Neto, Santiago Matalonga, Martín Solari, Gabriela Robiolo, and Guilherme Horta Travassos. 2017. Toward the characterization of software testing practices in South America: looking at Brazil and Uruguay. *Software Quality Journal* 25, 4 (01 Dec 2017), 1145–1183. https://doi.org/10.1007/s11219-016-9329-3

[12] Tore Dybå, Dag I.K. Sjøberg, and Daniela S. Cruzes. 2012. What Works for Whom, Where, when, and Why?: On the Role of Context in Empirical Software Engineering. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '12)*. ACM, New York, NY, USA, 19–28. https://doi.org/10.1145/2372251.2372256

[13] Emelie Engström and Per Runeson. 2010. A qualitative survey of regression testing practices. In *Product-Focused Software Process Improvement/Lecture Notes in Computer Science*, M. A. Babar, M. Vierimaa, and M. Oivo (Eds.) (Eds.), Vol. 6156. Springer, Berlin, Heidelberg, 3–16. https://doi.org/10.1007/978-3-642-13792-1_3

[14] Arlene Fink. 1995. *The Survey Handbook*. Sage publications, CA, USA.

[15] Vahid Garousi and Mika V. Mäntylä". 2016. When and what to automate in software testing? A multi-vocal literature review. *Information and Software Technology* 76 (2016), 92–117. https://doi.org/10.1016/j.infsof.2016.04.015

[16] Vahid Garousi and Junji Zhi. 2013. A survey of software testing practices in Canada. *Journal of Systems and Software* 86, 5 (2013), 1354–1376. https://doi.org/10.1016/j.jss.2012.12.051

[17] Adam M. Geras, Michael R. Smith, and James Miller. 2004. A survey of software testing practices in alberta. *Canadian Journal of Electrical and Computer Engineering* 29, 3 (July 2004), 183–191. https://doi.org/10.1109/CJECE.2004.1532522

[18] Jorge Goncalves, Simo Hosio, and Vassilis Kostakos. 2017. Eliciting Structured Knowledge from Situated Crowd Markets. *ACM Trans. Internet Technol.* 17, 2, Article 14 (mar 2017), 21 pages. https://doi.org/10.1145/3007900

[19] Kevin A. Hallgren. 2012. Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial. *Tutorials in quantitative methods for psychology* 8, 1 (2012), 23–34. http://europepmc.org/articles/PMC3402032

[20] Helppi, Ville-Veikko. Bitbar Technologies. 2015. The Basics Of Test Automation For Apps, Games, Mobile Web. https://www.smashingmagazine.com/2015/01/basic-test-automation-for-apps-games-and-mobile-web/.

[21] Simo Hosio, Jorge Goncalves, Theodoros Anagnostopoulos, and Vassilis Kostakos. 2016. Leveraging Wisdom of the Crowd for Decision Support. In *Proceedings of the 30th International BCS Human Computer Interaction Conference: Fusion! (HCI '16)*. BCS Learning & Development Ltd., Swindon, UK, Article 38, 12 pages. https://doi.org/10.14236/ewic/HCI2016.38

[22] Simo Hosio, Jorge Goncalves, Vassilis Kostakos, and Jukka Riekki. 2015. Crowdsourcing Public Opinion Using Urban Pervasive Technologies: Lessons From Real-Life Experiments in Oulu. *Policy & Internet* 7, 2 (2015), 203–222. https://doi.org/10.1002/poi3.90

[23] ISTQB (International Software Testing Qualifications Board). 2016. Worldwide Software Testing Practices Report 2015–2016. https://www.istqb.org/documents/ISTQB_Worldwide_Software_Testing_Practices_Report.pdf.

[24] Tanjila Kanij, Robert Merkel, and John Grundy. 2013. Lessons learned from conducting industry surveys in software testing. In *2013 1st International Workshop on Conducting Empirical Studies in Industry (CESI)*. IEEE, NJ, USA, 63–66. https://doi.org/10.1109/CESI.2013.6618474

[25] Barbara A. Kitchenham and Shari L. Pfleeger. 2008. *Personal Opinion Surveys*. Springer London, London, 63–92. https://doi.org/10.1007/978-1-84800-044-5_3

[26] Barbara A. Kitchenham, Shari L. Pfleeger, Lesley M. Pickard, W. Jones, Peter, David C. Hoaglin, Khaled El Emam, and Jarrett Rosenberg. 2002. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering* 28, 8 (Aug 2002), 721–734. https://doi.org/10.1109/TSE.2002.1027796

[27] Barbara Ann Kitchenham, Dag I.K. Sjøberg, Tore Dybå, Dietmar Pfahl, Pearl Brereton, David Budgen, Martin Höst, and Per Runeson. 2012. Three empirical studies on the agreement of reviewers about the quality of software engineering experiments. *Information and Software Technology* 54, 8 (2012), 804–819. https://doi.org/10.1016/j.infsof.2011.11.008 Special Issue: Voice of the Editorial Board.

[28] Terry K. Koo and Mae Y. Li. 2016. A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *Journal of Chiropractic Medicine* 15, 2 (2016), 155 – 163. https://doi.org/10.1016/j.jcm.2016.02.012

[29] Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage, Thousand Oaks, CA, USA.

[30] Klaus Krippendorff. 2011. Computing Krippendorff's Alpha Reliability. Retrieved May 18, 2018 from https://repository.upenn.edu/asc_papers/43

[31] Tarald O. Kvålseth. 1995. Coefficients of Variation for Nominal and Ordinal Categorical Data. *Perceptual and Motor Skills* 80, 3 (1995), 843–847. https://doi.org/10.2466/pms.1995.80.3.843

[32] J. Lee, S. Kang, and D. Lee. 2012. Survey on software testing practices. *IET Software* 6, 3 (2012), 275–282. https://doi.org/10.1073/pnas.0307752101

[33] Robert Libby and Roger K. Blashfield. 1978. Performance of a composite as a function of the number of judges. *Organizational Behavior and Human Performance* 21, 2 (1978), 121–129. https://doi.org/10.1016/0030-5073(78)90044-2

[34] A.E. Mannes, Jack B. Soll, and Richard P. Larrick. 2014. The wisdom of select crowds. *Personality and Social Psychology* 107, 2 (2014), 276–299. https://doi.org/10.1037/a0036677

[35] Jason W. Osborne and Amy Overbay. 2004. The power of outliers (and why researchers should ALWAYS check for them). *Practical Assessment, Research & Evaluation* 9, 6 (2004), 1–8. https://www.ingentaconnect.com/content/doaj/15317714/2004/00000009/00000006/art00001

[36] Amantia Pano, Daniel Graziotin, and Pekka Abrahamsson. 2018. Factors and actors leading to the adoption of a JavaScript framework. 10.1007/s10664-018-9613-x. "Article in Press".

[37] Carol Passos, Ana P. Braun, Daniela S. Cruzes, and Manoel Mendonca. 2011. Analyzing the Impact of Beliefs in Software Project Practices. In *2011 International Symposium on Empirical Software Engineering and Measurement*. IEEE, Banff, AB, Canada, 444–452. https://doi.org/10.1109/ESEM.2011.63

[38] Michael Quinn Patton. 2002. *Qualitative Research and Evaluation Methods, 3rd. ed.* Sage publications, Thousand Oaks, CA, USA.

[39] Robert M. Poston and Michael P. Sexton. 1992. Evaluating and selecting testing tools. In *[1992] Proceedings of the Second Symposium on Assessment of Quality Software Development Tools*. IEEE, New Orleans, LA, USA, 55–64. https://doi.org/10.1109/AQSDT.1992.205836

[40] QASource. 2016. A Guide to Selecting The Best Test Automation Tool. https://drive.google.com/open?id=0B6dKdxaNjBENY3pxMkgwb3BneVk.

[41] Dudekula Mohammad Rafi, Katam Reddy Kiran Moses, Kai Petersen, and Mika V. Mäntylä. 2012. Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey. In *Proceedings of the 7th International Workshop on Automation of Software Test (AST '12)*. IEEE Press, Piscataway, NJ, USA, Article 6, 7 pages. http://dl.acm.org/citation.cfm?id=2663608.2663616

[42] Pär J. Ågerfalk and Brian Fitzgerald. 2008. Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy. *MIS Quarterly* 32, 2 (2008), 385–409. http://www.jstor.org/stable/25148845

[43] Austen Rainer, Timothy Hall, and Nathan Baddoo. 2003. Persuading developers to "buy into" software process improvement: a local opinion and empirical evidence. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings*. IEEE, Rome, Italy, 326–335. https://doi.org/10.1109/ISESE.2003.1237993

[44] Päivi Raulamo-Jurvanen, Kari Kakkonen, and Mika Mäntylä. 2016. *Using Surveys and Web-Scraping to Select Tools for Software Testing Consultancy*. Springer International Publishing, Cham, 285–300. https://doi.org/10.1007/978-3-319-49094-6_18

[45] Päivi Raulamo-Jurvanen, Mika Mäntylä, and Vahid Garousi. 2017. Choosing the Right Test Automation Tool: A Grey Literature Review of Practitioner Sources. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (EASE'17)*. ACM, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/3084226.3084252

[46] Patrick E. Shrout and Joseph L. Fleiss. 1979. Intraclass Correlations: Uses in Assessing Rater Reliability. *Psychological Bulletin* 86, 2 (July 1979), 420–428.

[47] Steven E. Stemler. 2004. A Comparison of Consensus, Consistency, and Measurement Approaches to Estimating Interrater Reliability. *Practical Assessment, Research & Evaluation* 9, 4 (2004), 1–11. https://www.ingentaconnect.com/content/doaj/15317714/2004/00000009/00000004/art00001

[48] James Surowiecki. 2005. *The Wisdom of Crowds*. Anchor, New York, USA.

[49] John W. Tukey. 1977. *Exploratory data analysis*. Addison-Wesley, Reading, Mass.

[50] Gerard Wagenaar, Sietse Overbeek, and Remko Helms. 2017. Describing Criteria for Selecting a Scrum Tool Using the Technology Acceptance Model. In *Intelligent Information and Database Systems*, Ngoc Thanh Nguyen, Satoshi Tojo, Le Minh Nguyen, and Bogdan Trawiński (Eds.). Springer International Publishing, Cham, 811–821. https://doi.org/10.1007/978-3-319-54430-4_77

[51] Claes Wohlin, Martin Höst, and Kennet Henningsson. 2003. *Empirical Research Methods in Software Engineering*. Springer, Berlin, Heidelberg, 7–23. https://doi.org/10.1007/978-3-540-45143-3_2

[52] Claes Wohlin, Martin Höst, and Kennet Henningsson. 2006. Empirical Research Methods in Web and Software Engineering. In *Web Engineering*, Emilia Mendes and Nile Mosley (Eds.). Springer, Berlin, Heidelberg, 409–430. https://doi.org/10.1007/3-540-28218-1_13

A RELATED RESEARCH MATERIAL

See related material in https://drive.google.com/open?id=1yg7FPY8wNiFOMmwz6-BLiN7ileIBVNKd