Practitioner Evaluations on Software Testing Tools

Päivi Raulamo-Jurvanen  
M3S, University of Oulu  
Oulu, Finland  
paiivi.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 demographic variables. We set to answer the following research questions: (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...
To evaluate software testing tools, we need collective information and explanations of Krippendorff’s alpha \(^1\) for the survey from a set of characteristics considered important (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\(^2\) 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 Gonçalves 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 ITENA TESTOMAT\(^3\) research project. The selected practitioners were known to be either familiar with Robot Framework\(^4\), an open source, “generic test automation framework for acceptance testing and acceptance test-driven development (ATDD)” \((15)\) (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 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 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) \((30)\) as a measure of reliability of evaluations, and coefficient of variation (CV) \((5, 47)\) 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\(^2\) quality model.

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

---

\(^1\) Krippendorff’s alpha

\(^2\) ISO/IEC 25010

\(^3\) ITENA TESTOMAT

\(^4\) Robot Framework
of the data. Tukey [49] defined an outlier as a value more than 1.5 times the interquartile range (IQR, i.e., $Q_3 - Q_1$) from the quartiles, i.e., either below $Q_1 - 1.5 \times IQR$ or above $Q_3 + 1.5 \times IQR$. 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 ($\alpha$) is a statistical measure for determining inter-rater reliability. The values for the $\alpha$ 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 $\alpha$ 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]. For as Krippendorff’s $\alpha$, the values for ICC vary between 0 and 1, higher values indicating greater reliability. The commonly referenced ICC values are $\geq 0.90$ for excellent, $0.75 \leq \alpha < 0.90$ for good, $0.50 \leq \alpha$ for moderate and $< 0.50$ 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

---

Table 1: Questions in the Survey Tool

| Question identifiers | Question included in survey#2 only |
|----------------------|-----------------------------------|
| B1                   | C4 Easy / intuitive to use         |
| B2                   | C5 Usage of the tool does not require programming skills. |
| B3                   | C6 Reporting features of the tool for testing results. |
| B4                   | C7 Possibility to configure the tool for the needs. |
| B5                   | C8 Possibility to remodel or expand the tool. |
| B6                   | C9 Cross-platform support.         |
| B7                   | C10 Maintenance and re-use of test cases & test data. |
| B8                   | C11 Active further development of the tool. |
| B9                   | C12 Popularity of the tool.        |
| C1                   | C13 Low cost price or licensing of the tool (expected costs for acquisition and usage). |
| C2                   | C14 Performance of the tool (e.g. speed) for its purpose. |
| C3                   | C15 Cost-effectiveness.            |
| RFW                  |                                   |

---

\(^4\)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
\[
CV = \frac{\sigma}{\bar{x}} = \frac{\text{sqrt}(\text{var}(x))}{\text{mean}(x)}
\]

(2) CV for ordinal type of data, as in Kvålseth [31],
\[
\Delta = \sum_{i<j} |i-j|P_iP_j
\]
\[
\Delta^* = \left[\frac{4}{(k-1)}\right] \Delta
\]
\[
CV = 1 - \left[1 - \frac{\Delta^*}{\Delta}\right]^{1/2}
\]

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 \((n - 1) \ast 100\) 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.nb⁶. We used an automatic method, R-function stepAIC⁷ 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.

### Table 2: Top 6 Tools - Survey Data

| Tool       | Survey#1 | Survey#2 | Total |
|------------|----------|----------|-------|
|            | 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 Programming 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

---

⁶https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/glm.nb.html
⁷https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/stepAIC.html
⁸https://marketplace.atlassian.com/categories/test-management
In the worst case the ICC value would be considered "good". Thus, we did not expect all respondents to interpret the criteria in general. For Robot Framework, correlation), although the CV's for ordinal data were slightly better, of variation (CV) for the evaluations of top 6 tools, see Table 6.

The ICC(3,k) for Robot Framework was in the "good" range (although the value 0.94 was in the "excellent" range [28]), while for Selenium with the "moderate" level of reliability was reached with 3 respondents while for Jenkins 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 using the tool, familiarity with Robot Framework (see the question ID RWF 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 categorical 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 3: Respondents' Background (n = 89)

| Basis of Evaluations | Software Development | Software Testing | Requirements Mgmt | Project Mgmt | Not specified, NA |
|----------------------|----------------------|------------------|------------------|--------------|------------------|
| Number of Respondents in Work Areas | 19 | 21.3% | 55 | 61.8% | 1 | 1.1% | 4 | 4.5% | 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 | Max (Min) | 45 | 0 | in years | Avg (Median) | 12.9 | (11.0) |
| Current Role | Max (Min) | 24.0 | 0 | Basis of Evaluations | Opinion | 149 | 7.0% |
Table 4: Top 6 Tools - Survey Evaluations, Number of outlier values & Rank of criteria

| Rank | Criteria                  | Appium         | Jenkins        | Jira           | JMeter         | Robot Fw<sup>a</sup> | Selenium |
|------|---------------------------|----------------|----------------|----------------|----------------|----------------------|-----------|
| 6    | Applicability             | 66.0<sup>6</sup> | 0              | 84.0<sup>5</sup> | 1              | 83.8<sup>5</sup>    | 0         |
| 8    | Compatibility             | 52.0<sup>9</sup> | 0              | 84.5<sup>4</sup> | 0              | 77.5<sup>6</sup>    | 1         |
| 7    | Configurability           | 70.0<sup>5</sup> | 0              | 81.0<sup>8</sup> | 0              | 71.7<sup>12</sup>   | 5         |
| 4    | Cost-Effectiveness        | 61.3<sup>7</sup> | 0              | 86.5<sup>2</sup> | 0              | 82.0<sup>3</sup>    | 2         |
| 1    | Costs                     | 84.0<sup>10</sup>| 0              | 86.3<sup>1</sup> | 1              | 93.0<sup>1</sup>    | 5         |
| 5    | Cross-Platform Support    | 77.5<sup>3</sup> | 0              | 83.5<sup>6</sup> | 0              | 83.8<sup>3</sup>    | 5         |
| 9    | Easy To Deploy            | 39.0<sup>13</sup> | 0              | 70.5<sup>12</sup> | 0              | 57.5<sup>4</sup>    | 5         |
| 9    | Easy To Use               | 45.0<sup>11</sup> | 0              | 60.0<sup>15</sup> | 0              | 73.8<sup>6</sup>    | 5         |
| 13   | Expandability             | 46.3<sup>10</sup> | 0              | 78.5<sup>9</sup> | 0              | 73.8<sup>6</sup>    | 5         |
| 3    | Further Development       | 78.8<sup>2</sup> | 0              | 82.0<sup>7</sup> | 0              | 78.8<sup>8</sup>    | 5         |
| 12   | Maintenance of TC&D       | 53.8<sup>8</sup> | 0              | 72.0<sup>11</sup> | 0              | 65.0<sup>11</sup>   | 5         |
| 11   | Performance               | 36.0<sup>14</sup> | 0              | 77.0<sup>10</sup> | 1              | 65.0<sup>11</sup>   | 5         |
| 1    | Popularity                | 72.0<sup>3</sup> | 0              | 89.0<sup>1</sup> | 0              | 87.5<sup>1</sup>    | 5         |
| 15   | Programming Skills        | 27.5<sup>15</sup> | 0              | 63.5<sup>3</sup> | 0              | 87.5<sup>1</sup>    | 5         |
| 13   | Reporting Features        | 42.5<sup>12</sup> | 0              | 63.0<sup>14</sup> | 0              | 75.0<sup>8</sup>    | 5         |

Total # of outliers: 0 4 0 2 5 1

<sup>a</sup> Robot Framework
<sup>b</sup> Arithmetic mean of the evaluations for a criterion. The superscript is the ranking of the criterion.
<sup>c</sup> 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  | 0.0015 | 0.13  | 0.62  |
| ICC3  | 0.36 | 3.9  | 14  | 56  | 0.0015 | 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

| 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 | 0.90  |
| ICC3k | 0.83 | 5.8  | 14  | 126 | 1.2e-08 | 0.661 | 0.93  |

Number of criteria = 15 & Number of respondents = 10

| 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 | 0.55 | 16   | 14  | 1064| 1.6e-35 | 0.840 | 0.96  |
| ICC3k | 0.91 | 16   | 14  | 1064| 1.6e-35 | 0.881 | 0.97  |

Number of criteria = 15 & Number of respondents = 77

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

|               | Appium | Jenkins | Jira | Jmeter | Robot Fw<sup>a</sup> | Selenium |
|---------------|--------|---------|------|--------|----------------------|----------|
| Krippendorff’s α | 0.294  | 0.173   | 0.224| 0.069  | -0.07                | 0.127    |
|               | Ordin. | Ratio   | Ordin.| Ratio  | Ordin.                | Ratio    |
| Applicability | 0.13   | 0.11    | 0.06 | 0.20   | 0.18                 | 0.15     |
|               | Ordin. | Ratio   | Ratio | Ordin. | Ratio                | Ordin.   |
| Compatibility | 0.20   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Configurability| 0.13  | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Cost-Effectiveness | 0.08  | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
|               | Ordin. | Ratio   | Ratio | Ordin. | Ratio                | Ordin.   |
| Costs | 0.23   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Cross-Platform S. | 0.23  | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Easy To Deploy | 0.24   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Easy To Use | 0.22   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Expandability | 0.28   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Further Devel. | 0.22   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Maintenence | 0.20   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Performance | 0.27   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Popularity | 0.25   | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Programming Skills | 0.35  | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Reporting Features | 0.28  | 0.04    | 0.35 | 0.12   | 0.18                 | 0.36     |
| Pearson’s Corr. | r(13)  | 0.85    | r(13) | 0.99  | r(13)               | 0.96     |
| P-value | 6.497e−05 | 3.33e−12 | 9.534e−14 | 6.358e−09 | 8.522e−11 | 2.314e−05 |

<sup>a</sup> Robot Framework

<sup>b</sup> Ordinal level of measurement, see calculation for CV in Section 3.2.4.

<sup>c</sup> 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.
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”).
for some time, e.g., JMeter9 since 1998 but Appium10 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.

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. Assessing Architectural Issues with Quality Attributes: A Survey on Expert Agreement. In Proceedings 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 Arzuyan, Miganoush K. Magarian, and Mira Kajo-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, 6 (2000), 94–99. https://doi.org/10.1016/S0031-9406(05)61211-4
[6] Cappgemini Consulting. 2015. World Quality Report 2014-2015, Sixth Edition, 2014. https://www.sogei.com/explore/reports/world-quality-report-2014-2015/
[7] Cappgemini Consulting. 2016. World Quality Report 2015-2016, 2015. https://www.sogei.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 Diaz-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 Dyba, 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/10.1145/2372521.2372526
[13] Annelie Engstroem and Per Ranesson. 2010. A qualitative survey of regression testing practices. In Product-Focused Software Process Improvement/Lecture Notes in Computer Science, M. A. Bahar, M. Vierimaa, and M. Oivo (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.

9 https://web.archive.org/web/20150419065701/http://projects.apache.org/projects/jmeter.html
10 http://appium.io/history.html
