Using Surveys and Web-Scraping to Select Tools for Software Testing Consultancy

Päivi Raulamo-Jurvanen[1,✉], Kari Kakkonen[2], and Mika Mäntylä[1]

[1] M3S (M-Group), ITEE, University of Oulu, Finland
{paivi.raulamo-jurvanen,mika.mantyla}@oulu.fi
[2] Knowit Oy, Helsinki, Finland
kari.kakkonen@knowit.fi

Abstract. We analyzed findings from data collected utilizing surveys and Web-scraping, to support Knowit Oy, a software testing consultation company, in the process of selecting the right tools for software testing & test automation. We conducted two surveys (2013 & 2016) among (mostly Finnish) software professionals to acquire criteria and a list of tools used for software testing in industry. Considering all our data sources Selenium was the most popular pure tool, while Robot Framework was the most referenced tool (latter survey). According to the surveys Jenkins and Sikuli have the highest increase in popularity (or familiarity). Top referred criteria for selection were usability, functionality, maintainability and available support for a tool. While Knowit considers it best to utilize traditional surveys, Web-scraping is seen as cost effective support for such instruments. To get comprehensive picture and to gain knowledge of the tools in markets multiple sources should be used.

Keywords: Test automation·software testing tool·software test automation tool·tool support·selection criteria.

1 Introduction

Software test automation is tool-oriented domain and integral to frequent testing as part of continuous delivery and rapid releases. A recent online survey reported test automation to be a key factor with software quality and R&D cost saving from the viewpoint of management [19]. In another recent survey by ISTQB [5] test automation was ranked as the main area of improvement opportunities in testing activities. In addition to that, test tool/automation consultation was ranked as the service most required from external providers. The results from a survey by Capgemini, Sogeti and HP (the World Quality Report, WQR 2015-2016) [1] highlight that investing in test automation is a must to keep up with the ever increasing demand for velocity. The findings claimed that 40% of the respondents (IT leaders of mobile technologies) reported lack of right tools for their testing activities. Overall, these sources highlight the importance of test automation which cannot succeed without proper tools.

Selecting tools for software testing is a difficult practical problem as there are numerous software testing tools available. The exact number of tools is unknown as

what constitutes as a test tool is difficult to define, e.g. many people use Excel to manage test cases. The high number of tools is well reflected for example in the web-site listing tools for pair-wise testing <http://www.pairwise.org/tools.asp>, a technique to generate minimum number of test cases covering all combinations of two test inputs. The web-site has listed 41 tools for pair-wise testing, aka combinatorial testing, alone. Given the number of tools available only for this test input generation technique, we can estimate that there has to be hundreds, if not thousands, of testing tools available.

In this paper, we study the knowledge acquisition phase of the test tool selection process within a consulting company (Knowit Oy). This is our initial work on test automation tool selection and thus we provide only initial answers to following Research Questions: RQ1) *Why is selecting tools important for Knowit?* RQ2) *What is the most popular software test tool nowadays, in comparison to year 2013?* RQ3) *How does Web-scraping compare with traditional surveys?* RQ4) *What criteria people find important when selecting tools for software testing or test automation?*

2 Prior Work

Test automation consultants Graham and Fewster [2] have studied experiences of test automation with industrial cases over a long period of time. They emphasized that "there is no such thing as the perfect tool, but there are many tools that would be adequate for a given situation", it is the preferences that drive the decisions. They claim that the tool must be appropriate for a job. A tool may be inadequate in some context but suitable in another and several tools may have to be used to accomplish the goals. [2]

An online survey published in March 2016 [19] focused on the tools used in test automation, in companies of less and more than 100 employees. The findings from a total of 644 software professionals indicated organizations to use more than a single test automation solution, open source tools being popular, especially in smaller organizations. The amount of tools is thought to be high due to reasons like application complexity, multiple platforms or lack of required functionalities. The study anticipated possible disappearance of commercial tools in favor of open source tools in the near future. The most used tools were, in the order of preference, for smaller size organizations: Selenium (42%), internal tools (20%), Junit (12%), Android SKD (8%), Appium (7%), JMeter (3%), Watir (3%), Pytest (3% and Selendroid (2%). For larger organizations the tools most used were: Selenium (29%), internal tools (24%), Junit (12%), Appium (8%), Microsoft (8%, in general), QTP (7%), Selendroid (5%), TestComplete (5%) and JMeter (2%). [19].

Past work on software testing tool selection in general by Poston and Sexton [13] perceived systematic data collection method, preferably with forms or checklists, to be the secret for selecting appropriate testing tools. Although several surveys of software testing have been conducted, e.g. [3, 8, 9, 15], those typically do not cover the actual tools used. There are however studies that focus on a few specific tools e.g. comparing TestComplete and QTP on characteristics [7], acknowledging the need for

evaluation of tools [12], comparing Selenium, QTP and TestComplete (eventually concluding the best tool being QTP) [6] or comparing a few web-service tools [4].

To summarize, according to the authors' best knowledge peer-reviewed literature is missing surveys that would focus on the tools by actually naming them. Tools are essential for our trade. Academics need tools for teaching and practitioners for their business. Often tools are listed in requirements of job ads making tool knowledge essential for students graduating from universities.

3 Case Context and the Problem (RQ1)

In this section, we first describe the context of our work and the particular problem we are trying to solve. We use a checklist by Petersen and Wohlin [11] to describe our context in **Table 1**. Evidence-based software engineering can exploit the context description if that is done as completely and accurately as possible for the targeted object of study. Next, we describe the problem with an informal question answer format.

Table 1. Case context with the framework by [11]

Object of Study	**Tool selection and process acquiring related knowledge.**
Product	Service offering provided by the software testing consultancy company.
People	Technology consultants, Customer consultants, Tool owner in the customer organization.
Practices, Tools & Techniques	Partner discussions and information, trade fairs, cross-customer recommendation.
Processes	Software development, Software Testing, Training & Deployment. Technology, Partner, Portfolio and Project management.

- *Why is selecting tools important for Knowit?* A software development and testing project success is built on people, processes and tools. It is important to be able to recommend and help to choose a set of tools that is effective and efficient in tasks and fit the context in question.
- *Why are tool surveys conducted?* Test tools get more visibility in the industry. Surveys provide understanding about tools on the rise and tools on the decline. There are excellent newcomers to the tool scene, there are changes in product portfolios and features of existing tools sets. Identifying tools gaining market share at a given time helps to steer for the next good tool.
- *What is to be gained by surveys for tool selection?* Tool selection surveys collect and distribute the collective information from people who have invested time in choosing and using a tool. Such knowledge can make the tool comparison and selection process more efficient.

- *What does tool selection mean to the business?* The business of the customer or end user of software development process gains efficiency and effectiveness using the tool. A suitable test automation tool will impact the project velocity positively (enabling e.g. faster time to market). The efficiency provided by two different tools can be significant. For the business of a consultancy company, the tool selection is an essential part of the service offering, a must-have service although a minor one if calculated in turnover. More importantly, the consultancy company wants to provide tools bringing the best efficiency. In the end, both consultancy and customer always share the same common goal of customer business success.
- *What are the experienced difficulties?* Typical challenge is the willingness of unenlightened stakeholders to use a good tool for a purpose other than the tool was originally designed for. That may prevent achievement of the expected results for the tool adoption. Another difficulty is comparison of tools that are similar on paper, e.g. "test management" tools, some of which work on cloud and some with native client. In such case it is essential to understand the really important characteristics of the tools.

4 Research Methods

This section describes research methods used for gathering knowledge about the testing tools.

4.1 Surveys

We present results of two different surveys that collected information about test tool usage, mainly in Finnish software industry. A survey can be thought as a vehicle to harness the "wisdom of the crowds" for tool selection process. The concept embodies the idea of collective opinion (or intelligence), that under the right circumstances a group can be smarter than a single individual [14].

The first tool survey, Survey 1, was conducted in 2013 (as a thesis work for Master degree of Business Informatics at Metropolia) by Knowit employee Minna Tiitinen with Kari Kakkonen as a tutor [16]. This survey offers historical perspective on how Knowit has utilized surveys. The survey was distributed in public email list of Finnish professional testing society (TestausOSY) and also to the partners of Knowit, receiving 107 answers.

In 2016, University of Oulu and Knowit jointly conducted a tool survey, Survey 2, to find out 1) the criteria people used (or preferred) for tool selection and 2) tools that were used by software industry. The survey was targeted to software professionals and links to the questionnaires were provided in Facebook, LinkedIn and Twitter to selected groups (mainly Finnish software testing related groups) and sent to email lists of Finnish professional testing society (TestausOSY) by the partner organization, Knowit. The total of 58 answers had a clear bias in favor of Finnish respondents (51).

The questionnaires for the two surveys were rather different of nature; first, by the number of questions and second, by the design of the questions. The questionnaire for

Survey 1 included 61 questions in total (9 questions for the background information about the respondents and 52 questions for the tools). For Survey 2, the total number of questions was only 8 (5 and 3 questions for background information and tools, respectively). Survey 1 provided multiple choice questions with predefined lists of tools (and an option to add tools), the criteria for the predefined lists of tools being ISTQB tool classification and (biased) commonness of the tools. For Survey 2, the intention was not to steer respondents' opinions or tool choices by providing free text fields only. The original questionnaires are available for Survey 1 as Appendix 1 (pg. 113) in [16] and for Survey 2 from http://goo.gl/MjPFCr.

4.2 Web-scraping

Since surveys in general require effort to create, administer and respond and may suffer from low response rates and respondent bias, we utilized Web-scraping that could support or even possibly substitute surveys. Web-scraping is an approach to fetch content from the internet, a technique to access web-pages and extract a structured view of the required data [10]. However, there are both legal and ethical issues involved in this matter, e.g. the Terms of Service (ToS) for any service may clearly prohibit data scraping from the website or the usage of scraped data may violate a website owner's copyrights. Sometimes a service may provide a public API to access some data, the quality and quantity of which may be lower than (or not as up-to-date as) data acquired by disruptive web scraping. (Some services may offer free services for limited access and require a paid fee for more frequent or massive usage).

We utilized Web-scraping to collect wider views of the Top 15 tools of Survey 2. Data collected included number of Wikipedia page views, number of Google hits (using a particular search string), number of StackOverflow questions and view counts for those, and number of Twitter tweets (see **Table 3** & **Table 4** – please note, rows in both tables are sorted by column "Rank" of **Table 4**). The time period used for the searches was intentionally set on three months (January 1st - March 31st 2016) to provide some variation and recent body to the content. The data was collected on April 20th and 21st 2016.

Wikipedia is web-based encyclopedia with openly editable content, the English version of which alone contains over 5 million articles. To get the trends of (user created) page views for Wikipedia articles (available in en.wikipedia) we utilized the Pageview API[1] in RStudio/R (required R packages 'httr' and 'jsonlite').

Google Search is claimed to be the most used web search engine on the WWW. ToS of Google strictly deny any access to their services via "using a method other than the interface and the instructions that we provide"[2]. Thus, the Google hits were collected manually using Firefox browser and search string "<toolname> and 'software testing' and tool".

[1] https://wikitech.wikimedia.org/wiki/Analytics/PageviewAPI
[2] https://www.google.com/intl/en/policies/terms/?fg=1

For StackOverflow questions the data was fetched from StackExchange Data Explorer[3] (open source tool) using the provided SQL-query editor. StackExchange data explorer provides libraries of "high-quality questions and answers" and allows to fetch and download data from different sites, of which StackOverflow, language independent site for programmers, is one. The actual data fetched from StackExchange included title of the question, number of views and creation date for each question (not the full body of the posts).

Twitter is a popular, online social networking service to communicate via short 140-character messages ("tweets"). The existing Twitter API has limitations to fetching tweets (e.g. for the time period or tweets per day). Thus, an open source project GetOldTweets-java (v1.2.0), written by Henrique Jefferson[4], was utilized, allowing to get the tweets for the tools for the defined observation period. The names of the tools were used as hashtags when searching for the posted tweets.

5 Results

First, the results of Survey 1 are summarized only (as from 2013). Then the results of Survey 2 are presented and contrasted with those of Survey 1. Finally, the results from Web-scraping are presented.

5.1 Survey 1

The results of the Survey 1 (107 respondents) showed that agile processes and tools adapted to them were on the rise. Most companies seemed to use both commercial (88%) and open source tools (60%), and even proprietary tools (48%). Unsurprisingly, open source tools dominated in small companies and commercial tools in large companies. The different ways how tools were acquired in companies of different size are shown in **Table 2**.

Table 2. Types of tools & Company sizes (Survey 1)

	1-10	**11-50**	**51-100**	**101-500**	**501-1000**	**Over 1000**	**Sum**	**%**
Bought	1	5	4	22	14	46	92	40
Open Source	1	5	5	12	6	27	56	24
Proprietary Tools	2	2	2	13	7	24	50	22
Cloud Service	0	2	2	3	2	6	15	7
Rented	0	0	1	3	1	5	10	4
Other	0	1	2	2	1	1	7	3

[3] http://data.stackexchange.com/

[4] https://github.com/Jefferson-Henrique/GetOldTweets-java/

In general, in overall analysis of all tools, HP and Atlassian were the most popular commercial vendors (having tools for different testing activities) while Selenium and Robot Framework were the most popular open source tools. For different test activities, mostly tools were used for test execution, test case and defect management and reporting. Excel was widely used on the side of the more sophisticated tools. The test execution tools used in the Survey 1 are shown in **Fig. 1**. The most popular test execution tool was Selenium with 45% of the respondents using it. QTP (nowadays replaced by UTF) and Robot Framework seemed almost level while the rest of the tools were used by a small number of respondents only. (A category not shown in the figure was "Other" (21%) which included tools referenced just once by respondents to the option "Other" for the question of test execution tools).

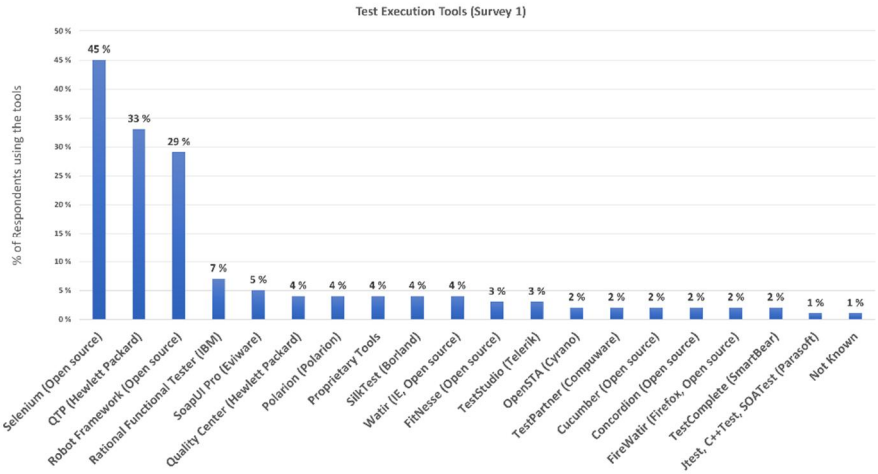

Fig. 1. Test execution tools used (Survey 1)

Some of the actions taken based on the results of the survey emphasized the need 1) to gain tool expertise in choosing tools (not just take the first one) and 2) to look more into the most popular tools in the survey. At the time the survey highlighted the unanticipated importance of Quality Center (HP). Today, the distribution of tools offered by Knowit to the customers has slightly changed (i.e. somewhat more Atlassian & Robot Framework (Knowit is one of the founding members of Robot Framework Foundation) and somewhat less HP). However, tool changes are quite expensive investments into learning, migrating data etc., let alone the actual tool selection and implementation. Thus, tool choices are only questioned every 3-4 years or so. One has to use a tool for some time to gain benefits of it. Also, as tool integration provides extra efficiencies, intent has been increased to integrate commercial and open source tools. The percentages of the adoption of the tools in Survey 1, as a comparison to the Top 15 tools of Survey 2, are listed in **Table 3** (column "2013 Survey, Usage %") .

5.2 Survey 2

Regarding the expertise of the respondents (58), the average years in software industry was 15.88 (median 15), while maximum years was 43 and minimum 3. Thus, the group of respondents was rather mature in years spent in software industry. The number of those having been in software industry for ten years or more was 47 (82%).

Criteria (RQ4)

The respondents were requested to describe important criteria when selecting a tool for software testing or software test automation. They were requested to describe in their own words what matters in general (e.g. regardless of technique, testing area or tool), what would be good to know, or take into account in advance. The respondents were expected to provide short, accurate descriptions of the features or characteristics they value in such tools. The question was intentionally left open: "*What are important criteria when selecting a tool for Software Testing / Test Automation? (E.g. What features or characteristics do you value in a tool? Or in your opinion, what would be good to know in advance or matters to you in general?)*". The fundamental purpose of the question was to collect data as a basis for further studies.

We assume that general requirements for software testing tools (e.g. costs, possible licensing model or developer support, to mention a few) are rather similar (even globally) despite physical location. It is notable that some criteria are always more important to some users than to others and not all criteria work for all even though we can observe general trends. Each and every software project is unique and must choose the criteria and how to apply those in their context.

The criteria were coded in NVivo, first by qualitative coding by topics appearing in the responses, i.e. open coding and axial coding. Later we mapped our codes to the ISO/IEC 25010 quality model[5]. As software testing tools are software too, ISO/IEC 25010 quality model can be used to represent the desired characteristics proposed by our respondents. However, we added "Support" and "Costs" to the categories since those were not included in the model and our respondents frequently brought up those topics. The references to the categories from the quality model are shown in **Fig. 2**.

Clearly the issue valued by the respondents was "Usability" with references to "Operability" (43), "Learnability" (13), "User Interface Aesthetics" (3), "Accessibility" (1) and "Appropriateness Recognizability" (1). "Functional Suitability" had references to "Functional Appropriateness" (24), "Functional Correctness" (17) and "Functional Completeness" (11). "Maintainability" included references to "Modifiability" (20), "Modularity" (8), "Reusability" (5) and "Analyzability" (2). None of the respondents referenced the sub-category of "Testability" which seems rather natural.

"Portability" was referenced as "Installability" (18) and "Adaptability" (10), but none of the respondents referenced "Replaceability". "Compatibility" with subcategories "Co-existence" (6) and "Interoperability" (20), however, this issue about compatibility (with tools/systems/platform/integration) is somewhat ambiguous since respondents were not always specific with their wording. "Reliability" included "Ma-

[5] http://iso25000.com/index.php/en/iso-25000-standards/iso-25010

turity" (8), "Fault Tolerance" (3) and "Availability" (1). "Performance efficiency" had reference to "Time Behavior" (4), "Resource Utilization" (3) and "Capacity" (1). For "Security", there was one reference for "Integrity".

"Support" was added with sub-categories "General Tool Support" (12), "Popularity" (9), "Future of the Tool" (6, we interpreted that the future is mainly of interest because support and new versions of the tool are needed in the future) and "Vendor Independence" (1). For "Costs" there were references for "Price" (11), "License model" (11), "Acquisition costs" (3), "Operating Costs" (3) and "Free" (3).

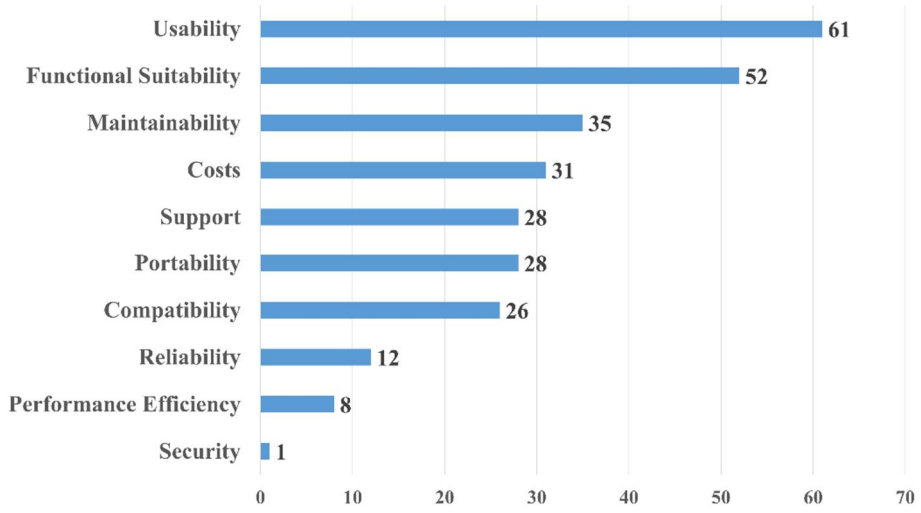

Fig. 2. Survey 2 Criteria categorized using ISO25010 and with "Support" and "Cost" categories

Open Source software becoming more common may have an impact on the concern for price or costs in general. However, when respondents were referring to "Open Source software" (or "open source interfaces") we interpreted those to refer to licensing model although we were not certain whether they were referring to importance of related costs, licensing model or modifiability of the software, or all of those. Adoption of a software testing or software test automation tool is expected to be a sustainable investment. Since adoption of free tools brings on additional costs, mostly resource related, there are expectations for a lifetime and development of those tools.

"Usability" and "Functional Suitability" of a tool were of high importance. The respondents valued maintainability, i.e. ability to configure or modify the tool according to their needs. "Support" was a topic of its own in the responses. The respondents seemed to prefer tools that were mature, i.e. had been adopted by fellow professionals already for some period of time and had support available in various software forums, e.g. StackOverflow.

Tools Used by the Respondents

The question querying about the tools used resulted in 164 different tools. The question for tools was intentionally open, not to give bias in favor of any tool: "*List tools for Software Testing / Software Test Automation you have been using yourself or tools which have been utilized in your organization. (For non-public, self-made tools you may write "Inhouse tool for doing X*")". The number of tools, considered as identifiable software testing or test automation tools, was 133 (excluding tools e.g. like Excel, Word, WinSCP, PuTTY or Cygwin).

To support the goals of the partner organization of having up-to-date sales offering of the most popular testing tools, we analyzed the top 15 tools based on the number of responses. The list of Top 15 tools includes different types and range of tools. The top four referenced tools are all open source tools and only five of the Top 15 tools were commercial. Open source tools seemed popular among the respondents of this survey, too. Nearly 70% of the respondents (column "2016 Survey, Usage %", **Table 3**) had some experience with the most used tool, Robot Framework. The respondents were mainly from Finland, thus the popularity (of familiarity) of Robot Framework in comparison to other tools in understandable. Interestingly, the Top 5 tools within included tools for different purposes, e.g. for acceptance testing (Robot Framework), web application testing (Selenium), continuous integration (Jenkins) and cross-platform functional testing (UFT/QTP & SoapUI). **Table 3** & **Table 4** show the details of the Top 15 tools and related data.

Contrasting Survey 2 with Survey 1 (RQ2)

Looking at the new results, (although two surveys were somewhat different of nature) it seems that the market has partially changed and partially stayed the same over the three years. The same tools seem to dominate, with just slight changes in market share (e.g. Robot Framework & Selenium). One could see somewhat more mobile-suitable tools, e.g. Appium, in the results as an expected development. In particular, Jenkins (continuous build and test management tool) and Sikuli (visual GUI testing tool) have appeared as new tools in comparison to Survey 1. Both surveys indicated a strong preference for using supportive tools (e.g. Office tools). According to Survey 1 those tools were mainly used for reporting and documentation purposes. Thus, it is unsurprising that reporting features were also listed as one of the important supportive features for test automation in Survey 2.

In both surveys, we also requested the respondents to describe important criteria when selecting a tool (for software testing or test automation). Qualitative coding of the criteria of Survey 2 indicated two clearly important categories, the possibility to modify the tool (to the needs of the organization) and usability of the tool. The respondents seemed to value features such as (in the order of preference) ease and intuitiveness of use, compatibility of the tool with the existing system, applicability (to tasks, methods and processes), reporting features as well as price. Cost related issues were not the topmost interest for the respondents of Survey 2 (although related to the concept of open source tools).

In Survey 1 the functionalities reported to affect the acquisition of software testing tools the most were (in the order of preference) price, ease of use, functionality, man-

agement and compatibility with other applications. The differences in conducting the surveys may affect the results (e.g. possibly having options to choose from vs. free text questions as in Survey 1 & 2, respectively).

Table 3. Top 15 tools of Survey 2 (#1): Usage % (figures for Survey 1, 2013) included for reference), Wikipedia page views & Google Hits (rankings for columns included)

		2016 Survey	2013 Survey	Wikipedia	Google
Tool	**Vendor**	**Usage %**	**Usage %**	**PageViews**	**GoogleHits**
Python	Open Source SW	9 [8]	---	547245 [1]	453000 [1]
Selenium	Open Source SW	47 [2]	45	112851 [3]	428000 [2]
Jenkins	Open Source SW	26 [3]	---	118621 [2]	134000 [12]
TFS	Microsoft	5 [10]	---	80620 [5]	164000 [11]
Junit	Open Source SW	12 [7]	27	28872 [7]	257000 [8]
UFT & QTP **	Hewlett-Packard	19 [4]	32	34640 [6]	414000 [4]
Jira	Atlassian	17 [5]	54	97178 [4]	340000 [5]
Quality Center	Hewlett-Packard	17 [5]	54	27089 [8]	274000 [6]
SoapUI	Open Source SW	19 [4]	5	24264 [10]	187000 [10]
jMeter	Open Source SW	14 [6]	24	15932 [11]	260000 [7]
Appium	Open Source SW	9 [8]	---	3 [14]	64500 [13]
ALM	Hewlett-Packard	7 [9]	---	24656 [9]	212000 [9]
Robot Framework	Open Source SW	69 [1]	29	11186 [12]	14500 [15]
FitNesse	Open Source SW	5 [10]	3	7670 [13]	424000 [3]
Sikuli	Open Source SW	14 [6]	---	0 [15]	32900 [14]

Table 4. Top 15 tools of Survey 2 (#2): StackOverflow questions & view counts, average views per question, Twitter tweets, total sum of rankings and ranking

	StackOverflow			Twitter	Total	
Tool	**Questions**	**ViewCount**	**Views / Qs**	**Tweets**	**Sum**	**Rank**
Python	42149 [1]	2132602 [1]	50,6 [4]	63664 [1]	17	**1**
Selenium	2765 [2]	144305 [2]	52,2 [3]	2290 [4]	18	**2**
Jenkins	1504 [3]	67632 [3]	45,0 [12]	2049 [5]	40	**3**
TFS	738 [5]	41143 [4]	55,7 [2]	1972 [6]	43	**4**
Junit	753 [4]	36381 [5]	48,3 [8]	419 [9]	48	**5**
UFT & QTP **	44 [11]	2091 [11]	47,5 [9]	1123 [8]	53	**6**
Jira	195 [8]	7449 [10]	38,2 [14]	1388 [7]	53	**6**
Quality Center	15 [15]	733 [15]	48,9 [7]	8924 [2]	58	**7**
SoapUI	191 [9]	9483 [8]	49,6 [6]	105 [12]	59	**8**
jMeter	496 [6]	19379 [7]	39,1 [13]	385 [10]	60	**9**
Appium	356 [7]	20315 [6]	57,1 [1]	296 [11]	60	**9**
ALM	32 [12]	1453 [12]	45,4 [11]	2298 [3]	65	**10**
Robot Framework	154 [10]	7703 [9]	50,0 [5]	40 [14]	66	**11**
FitNesse	27 [14]	824 [14]	30,5 [15]	56 [13]	82	**12**
Sikuli	30 [13]	1367 [13]	45,6 [10]	13 [15]	86	**13**

The Top 15 list of tools from Survey 2 has not yet affected the preferences of Knowit, but it supports their strategy by substantiating the general evidence of the rising trend of open source tools in the market. For example, the company has already adopted Jenkins as part of their offering despite the fact that it did not appear in the 2013 survey. From a viewpoint of a consulting company, tailored sets of tools are required to serve different types of needs of customers. The results from both surveys, in general, seem to correlate with the tools used by the wide customer-base (mainly Finnish companies) of Knowit.

5.3 Web-scraping of Top 15 Tools

From the data, see **Table 3** & **Table 4**, it seems that Python was clearly the tool being most "popular" considering any indicator. This is understandable: Python appeared about 25 years ago, thus the tool is expected to have far more software enthusiasts than any other tool in the list. We are aware Python is not a pure software testing tool, but mainly a programming language, supporting many testing related activities.

Surprisingly, the differences in the number of hits for the search terms were not as diverse among the tools as the case with Wikipedia page views. (Four tools having most hits shared the same level in hundreds of thousands). As a disclaimer, the hits were not analyzed any further to reveal whether those links were actually truly relevant to the original search terms or not. In fact, a hit is a request to a webserver for a file. Since Web pages may contain several files and images, loading a web page does not always equal to one [18]. Thus, it is claimed that a hit is an inaccurate measure of popularity or traffic of a website, page views providing a more accurate measure [17]. All Top 15 tools but two, Sikuli & Appium, had an article (in English) in Wikipedia. UFT shared an article with QTP (QTP being deprecated and replaced by UFT).

The commercial tools seem to have a trend of having less questions in StackOverflow than open source (free) tools. That may be due to a fact that commercial software vendors often provide dedicated customer services for their paying customers in need of help. With open source tools, people rely on the help of their fellow colleagues or forums of software professionals (or software enthusiasts). Also, the user manuals (e.g. online services) for commercial tools may be of higher quality and richer content than those for open source tools (if manuals exist). Interestingly, the differences in the average numbers of view counts per question for tools are not that big. Also, there seems to be a difference between posted tweets for commercial and open source tools compared with the number of StackOverflow questions. The five commercial tools are among the top eight tools for the number of tweets. However, the tweets may, due to the nature of the Twitter – list of short messages, possibly connected with hashtag(s), contain totally irrelevant content. Furthermore, since the length of a tweet is limited the tweets with "valid" content are not expected to include comprehensive discussions or descriptions but rather opinions or short comments like user tips, promotions, job advertisements or release notes.

Contrasting Web-scarping method with Surveys (RQ3)

Utilizing information in the web by different searches has for long been an important method for finding information about tools for Knowit. For a few decades there have been many websites collecting "the most used tools" or "the best tools" for the help of others looking for such tools. However, these are often rather biased geographically (e.g. US-based perspective only) and do not really show how common some tool is.

Web-scraping provides a rather quick way to acquire large amounts of data in comparison to surveys. However, selection of relevant sources for data can be difficult as well as finding the suitable methods to process and analyze the data, in order to provide useful or meaningful information. As experienced in this study, the Top 15 tools are quite different by characteristics and purpose. Thus, the popularity or familiarity of the tools, based on data from Survey 2, are not expected to be comparable, as such. Utilizing Web-scraping as a big-data style approach, as presented in this paper, gives some of the power of surveys into utilizing the web as a source for information. This makes the popularity of tools more evident. Still, surveys are irreplaceable in giving voice to the people themselves, especially in specific market sectors and contexts – and Web-scraping should be considered only as a good add-on source to surveys.

Contrasting Web-scraping results with Survey (RQ3)

Robot Framework was the most popular tool referenced by the respondents of Survey 2, however, just third but last (among the Top 15 tools) according to the ranking based on the results from Web-scraping (see **Table 3** & **Table 4**). Please note, the total ranking is counted from all ranked columns, taking account the number of references in Survey 2, Wikipedia page views, Google hits, StackOverflow questions, view count of those and average number of views per question as well as Twitter tweets. Popularity of Selenium and Jenkins was evident in the results of both Web-scraping and Survey 2 (although the latter was not referenced in Survey 1). Surprisingly, TFS seems to be widely adopted (based on the number of Wikipedia page views, StackOverflow questions, view count of those and even tweets) although that was not the case in Survey 2 (or Survey 1).

Popularity, as being widely adopted, is difficult to generalize from data from a survey since such results are always biased by the size and the origin of the sample, like in the case of Robot Framework. Web-scraping provides a wider, quantitative perspective to the tool scene. However, in our surveys the background information about respondents serves as anchor for positioning the results to more concrete contexts.

6 Discussion

Our data could not confirm the growth of the number of the internal tools in relation to open source frameworks, as reported by [19]. Only about 7% of the respondents of Survey 2 reported having been using an inhouse tool for some specific purpose. (It may also be that those tools have not been reported accurately or have not been con-

sidered as tools related to software testing). Interestingly, the most popular tool, Selenium, reported by [19] was the second most popular in our survey, after Robot Framework (particularly popular in Finland due to its origin). Four and five out of nine different tools reported for small and large size organizations (respectively) were also in the list of Top 15 tools in our Survey 2 [19]. However, the order of the tools was otherwise differing.

When considering popularity or familiarity of software testing or test automation tools, sources like StackOverflow or social media like Twitter are actually rather good sources of information. On the other hand, comparing commercial and open source tools that way does not seem very appropriate since most likely majority of questions or problems faced with commercial tools may have been handled via official (private) customer support channels, not via developer forums. Also, groups of software professionals may have formed self-sufficient support networks within their organizations only.

As with criteria provided by the respondents, ease of use, compatibility and applicability of the tools are important. Some software professionals may have been using some tool for years, may be well familiar with functionalities, pros and cons of such tools, but not aware of existing or new tools that could actually be more suitable (or supportive) for their purposes (e.g. considering costs or effectiveness). Some commercial tools provide a period of free trial but companies may not have the resources to share for trying out different choices or combinations of compatible tools. This is where the wisdom of the crowds could be applied to.

6.1 Limitations

There are several limitations affecting this study. Firstly, the surveys were targeted to (mainly Finnish) selected groups of software professionals, thus the two sets of respondents were rather small and expected to be biased (although experienced based on work history). Secondly, the list of tools analyzed is based on experiences of those small groups of respondents. The tools were not only related to software testing or software test automation but also more general and supportive for the process. Thirdly, comparison of popularity or familiarity of such tools brought up issues like commercialism and concept of open source software, different characteristics and purposes for the tools, different contexts for utilizing those tools and compatibility, just to mention a few. Furthermore, the surveys were not identical as the first survey used a list of preselected tool options while the latter survey was implemented as open text-fields. The question that remains is whether some less used tools are left out if only the main tools are mentioned (e.g. a tool set of a typical project may include 10-20 tools).

7 Conclusions and Future Work

In this paper, we make three contributions. First, we describe the test tool selection problem in a software testing consultancy. Having the right tools is critical for testing

consultants when they offer suitable services for their clients (RQ1). Second, we presented results of two surveys conducted in 2016 and 2013 (the responses mainly from Finland). Surveys are a good source of professional knowledge that can be applied, within the limits of a known context, as such. We found that among our respondents' the tools which have gained popularity are Robot Framework, Jenkins, and Sikuli while Selenium has maintained its high popularity (RQ2). Third, we present Web-scraping as a method that may provide additional quantitative (or qualitative) support for the tool selection process (RQ3).

According to our results, from both surveys and Web-scraping, the tools that ranked the most popular (based on the references of 2016 and metrics from Web-scraping, **Table 3** & **Table 4**) were Python, Selenium, Jenkins, TFS and Junit (in that particular order). Python was included in the list of Top 15 tools since many of the respondents of Survey 2 had listed it as an important instrument for test automation and thus is clearly of interest in global scale, too.

Results provided by the respondents regarding important criteria for tool selection were surprisingly alike (RQ4). However, when mapping our open coded criteria results to the ISO/IEC 25010 quality model, there were two evident categories that could not be directly mapped to the model, namely "Support" and "Costs". "Support" (including "General Tool Support", "Popularity", "Future of the Tool" and "Vendor Independence") or "Costs" (including "Price", "License model", "Acquisition costs", "Operating costs" or "Free") are in fact characteristics that may be critical in providing value to stated and implied needs of stakeholders.

The results highlight that local preferences may differ from global preferences (considering e.g. Robot Framework), but some tools like Selenium and Junit stand out as popular tools based on our surveys and Web-scraping as well as [19]. Web-scraping is seen as cost effective support for traditional surveys. Utilizing multiple sources enables getting a comprehensive picture of the tools in markets.

As future work, based on this study, another survey is planned to be conducted applying acquired criteria for related tools. The idea is to support the process of selecting the right tools by acquiring and comparing software professionals' knowledge or perceptions of characteristics of selected tools.

8 Acknowledgments

Our thanks to all respondents for their contribution to this survey.

9 References

1. Capgemini Consulting: World Quality Report 2015-2016. (2015), https://www.capgemini.com/thought-leadership/world-quality-report-2015-16
2. Fewster, M., Graham, D.: Software test automation: effective use of test execution tools. ACM Press/Addison-Wesley Publishing Co (1999)

3. Garousi, V., Zhi, J.: A survey of software testing practices in Canada. J. Syst. Software, 86, 5, 1354-1376 (2013)
4. Hussain, S., Wang, Z., Toure, I. K., Diop, A.: Web service testing tools: A comparative study. arXiv preprint arXiv:1306.4063 (2013)
5. ISTQB (International Software Testing Qualifications Board): ISTQB® Worldwide Software Testing Practices Report 2015 - 2016. (2016), http://www.istqb.org/references/surveys/istqb-worldwide-software-testing-practices-report.html
6. Kaur, H., Gupta, G.: Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete. Int. Journal of Engineering Research and Applications, 3, 5, 1739-1743 (2013)
7. Kaur, M., Kumari, R.: Comparative study of automated testing tools: Testcomplete and quicktest pro. Int. Journal of Computer Applications, 24, 1, 1-7 (2011)
8. Michael, J. B., Bossuyt, B. J., Snyder, B. B.: Metrics for measuring the effectiveness of software-testing tools. In: 13th International Symposium on Software Reliability Engineering, pp. 117-128. IEEE (2003)
9. Ng, S., Murnane, T., Reed, K., Grant, D., Chen, T.: A preliminary survey on software testing practices in Australia. In: 2004 Australian Software Engineering Conference, pp. 116-125. IEEE (2004)
10. Pan, A., Raposo, J., Álvarez, M., Hidalgo, J., Viña, Á.: Semi-automatic wrapper generation for commercial web sources. In: Rolland, C., Brinkkemper, S., Saeki, M. (eds.) IFIP TC8 / WG8.1 Working Conference on Engineering Information Systems in the Internet Context, vol. 103, pp. 265-283. Springer, New York (2002)
11. Petersen, K., Wohlin, C.: Context in industrial software engineering research. In: 3rd International Symposium on Empirical Software Engineering and Measurement, pp. 401-404. IEEE (2009)
12. Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M., Beecham, S.: Tools used in Global Software Engineering: A systematic mapping review. Information and Software Technology, 54, 7, 663-685 (2012)
13. Poston, R. M., Sexton, M. P.: Evaluating and selecting testing tools. Software, 9, 3, 33-42, IEEE (1992)
14. Surowiecki, J.: The wisdom of crowds. Anchor (2005)
15. Tassey, G.: The economic impacts of inadequate infrastructure for software testing. National Institute of Standards and Technology, RTI Project, 7007.011 (2002)
16. Tiitinen, M., Kakkonen, K.: Software Testing Tools Research in Finland, Knowit Oy (2013), http://www.knowit.fi/Documents/Testausty%C3%B6kalututkimus_Posteri_A3_Tiitinen_V11_ENG.pdf
17. Wikipedia Page view, https://en.wikipedia.org/wiki/Page_view
18. Wikipedia article traffic, https://en.wikipedia.org/wiki/Wikipedia_article_traffic
19. Yehezkel, S.: Test Automation Survey 2016, http://blog.testproject.io/2016/03/16/test-automation-survey-2016/