Software Testing: Survey of the Industry Practices

T. Hynninen*, J. Kasurinen**, A. Knutas*** and O. Taipale*
* Lappeenranta University of Technology, Lappeenranta, Finland
** South-Eastern Finland University of Applied Sciences (XAMK), Kotka, Finland
*** Lero, the Irish Software Research Centre, Dublin, Ireland
timo.hynninen@lut.fi, jussi.kasurinen@xamk.fi, antti.knutas@lero.ie, ossi.taipale@lut.fi

***Abstract* - The objective of this survey was to explore industry practices concerning software testing. We studied software organizations to assess how they test their products and what process models they follow. The data collection was implemented as an online implementation of the survey method. Additionally the collected data was compared to our prior survey study to understand how the industry practices have changed. According to our results, the organizations have shifted towards test automation and more sophisticated testing infrastructure, they apply more agile practices even in the mission-critical software, and they have reduced the use of formal process models.**

Keywords - software testing; survey; industry practices; quality assurance

I. Introduction

Testing can be one of the most expensive tasks for any software project. Besides causing immediate costs, problems of testing are also related to the costs of poor quality, malfunctioning programs and errors that cause large additional expenses to software producers during the maintenance [1], [2]. The costs related to testing are on the rise; the software industry has identified a need for reducing the growing cost of test environment management [3].

The objective of our study was to explore the software organizations' testing practices, tools and development process models to give an up-to-date picture of industry practices. In addition to answering these questions, this study is also a continuation study to our previous surveys (year 2009 [4] and year 2005 [5]) on the testing practices and test automation in the software industry. Comparison between earlier and current observations reveal changing practices.

The actual testing practices of the software industry were observed via an online survey, conducted in the beginning of 2017. We surveyed organizational units (OU) representing different sizes and business domains in software development. The survey questionnaire consisted of multi-choice, multi-item questions to collect quantitative data for statistical analysis and of open-ended questions for qualitative analysis. This mixed methods study [6] facilitated triangulation of the results [7]. Both the collected quantitative and qualitative data were used to assess the current practices, and compare our new results against our earlier survey results conducted seven years ago. According to the results, the applied software development models seemed to have shifted towards agile practices, causing changes in the testing infrastructure and test phases' emphasis. The number of automated tools in testing was rising, while the use of the formal process models and capability-maturity models were generally declining.

The work is structured as follows: In Section 2, related surveys and studies are introduced. Section 3 discusses the applied research method used in this work, and Section 4 presents the actual survey results with comparison to the results of our earlier survey. Discussion and conclusions are given in Sections 5 and 6, respectively.

II. Related Work

In addition to our earlier industry-wide survey of test automation and testing practices [4], software testing and test process improvement have been studied by others, for example, Ng et al. [8] in Australia and Chen et al. [9] in China. The study by Ng et al. applied the survey method to establish knowledge on such topics as testing methodologies, tools, metrics, standards, and training. Their study indicated that the most common barrier to developing testing was the lack of expertise in adopting new testing methods and the costs associated with testing tools; also in their study, only 11 organizations reported that they met the testing budget estimates. In a similar vein, Torkar and Mankefors [10] surveyed different types of communities and organizations. They found that 60% of the developers claimed that verification and validation were the first to be downgraded in cases of serious resource shortages during a project.

As for the industry studies, a similar study approach has previously been used in other areas of software engineering. For example, Ferreira and Cohen [11] completed a technically similar study in South Africa, although their study focused on the application of agile development and stakeholder satisfaction. Similarly, Li et al. [12] conducted research on the Commercial Off-The-Self (COTS) based software development process in Norway, Chen et al. [9] studied the application of open source components in software development in China, and Belt et al. [13] surveyed major Scandinavian telecom companies to identify the challenges of testing. Overall, case studies covering entire industry sectors are not particularly uncommon [14], [15].

On longitudinal studies in the development of testing practices, Garousi and Varma [16] conducted a series of surveys in the province of Alberta in Canada. They observed that from 2004 to 2009, the industry transitioned with a distinct elevation of codified practices: all V-model [16] levels of testing work (unit, system, and acceptance) increased along with the level of applied test automation. In addition, the amount of systematic training for the test personnel increased in all of the measured categories. Garousi and Zhi continued the work in 2013 with a nation-wide follow-up survey on the actual software testing practices, where they observed that new tools and development practices have been adopted in the Canadian industry since the prior study [17].

A study of testing practices by Lee, Kang and Lee [18] surveyed the amount of applied testing tools and test practices in South Korea. Their study reveals that even within the last ten years, some software organizations (12% of answers) have not had any meaningful test process or applied any test methods in practice. Interestingly, Lee, Kang and Lee also observed that in their survey population, application of system testing practices was more common than unit testing. One offered explanation was that unit testing is low level activity conducted by the developers, so it does not require separate tools or a process to be followed.

Khosla [19] estimated that in the near future, 80% of the staff in IT departments may be replaced by "artificial intelligence (AI) type systems." This estimate highlights, for example, automatic collection of run-time data, AI analysis of collected data together with testing and deployment automation during maintenance. Gartner report [20] also emphasizes the importance of automation. According to the Gartner report, software development phases cover 8 % and the maintenance phase, consisting of, for example, defect fixing, testing and deployment of new versions, covers 92 % of the total life cycle costs.

III. Research Method

The survey method described by Fink and Kosecoff [21] was used as the research method in this study in both of the surveys, in 2009 and again in 2017. The objective for a survey is to collect information from people about their feelings and beliefs. Further, surveys are most appropriate when information should come directly from the people [21]. Kitchenham et al. [22] divide comparable survey studies into exploratory studies from which explanations and estimates can be drawn, and confirmatory studies from which strong conclusions can be drawn. We consider this study as an exploratory, observational, and cross-sectional study that explores software testing practices and software quality approaches applied in the software industry.

The 2017 online survey questionnaire included eleven chapters containing questions of organization profile, software testing, test process maturity, applied process models and the tasks related to software development. The constructs were divided into multi-item questions based on, for example, theory, definitions or best practices of the construct. Multi-item questions are questions that are constructed by several items that measure one underlying construct. Chapters in the questionnaire were planned so that combining respondent's answers yielded holistic information of the surveyed organizational unit.

To facilitate comparison between our earlier and current survey, seven of the questionnaire chapters were taken directly from our earlier survey [4] which also observed testing and quality assurance practices. The design of the original data collection questionnaire was done by seven researchers from two different research groups. Two additional people were involved in the testing of the questionnaire with test interviews. The questionnaire for the data collection in 2017 was compiled by three researchers, and tested with representatives of our partner organization. The survey questionnaires from both 2009 and 2017 are available in the online appendix at https://doi.org/10.5281/zenodo.803995.

The 2009 and the 2017 survey both use the five point Likert scale: 1 fully disagree – 3 neutral – 5 fully agree. The 2017 survey was launched as a web survey via Webropol [23]. The sampling method was probability sampling. The survey was advertised in social media platforms such as LinkedIn, Facebook, Twitter and Researchgate, and by direct contacts to our industrial partners and open calls for participation in several public online discussion channels.

The survey results were analyzed with the R statistical language and its statistics ("stats") library [24]. In the statistical analysis, survey responses were also treated as single-item and not full constructs to see if the distribution of data between 2009 and 2017 had changed with any statistical significance. Descriptive statistics, displayed in more detail in the online appendix, were generated with the psych R library [25]. When analyzing interval data with the Mann-Whitney U statistical test, continuity correction was enabled to compensate for non-continuous data [26].

To estimate the sample size for our survey we used publically available statistics provided by the Ministry of Economic Affairs and Employment of Finland. According to the latest report of the software business sector from 2014 [27] there were 3360 companies whose main line of business was software production. The survey questionnaire was opened 930 times and it collected 33 unique responses from respondents working in different organizations within the four-week period it was available in January 2017. This gives the survey a response rate of 3.5 percent, which is fairly normal for Internet surveys according to Fink [21]. In comparison, the 2009 survey had 31 respondents from different software development organizations. This also indicates that both of the surveys had similar-sized sample of the software industry which also, while acknowledging some limitations similar to Iivari [15], were sufficient samples of the industry, and could be analyzed with quantitative approaches.

The survey was anonymous. To identify clusters and to classify answers we collected general information of the organizational unit. This information helped us to classify qualitative answers of the open-ended questions to quantitatively observed clusters. The objective of the study was not to collect data from a certain country but to reveal possible changes in the industry practices.

IV. SURVEY RESULTS

The survey questionnaire included general information of the organizational unit, a number of multi-choice, multi-item questions and open-ended questions. The multi-item questions was estimated by using the Cronbach alpha in the earlier surveys: The Cronbach coefficient alpha expresses the degree to which items in a scale are homogeneous.

Questions concerned the development practices and the available quality assurance infrastructure. In this section we present the survey results collected in January 2017. The results were compared against the 2009 results. We use mode as the primary indicator for individual items in the questionnaire, as the survey questions used an interval Likert scale. Additionally, we performed statistical analysis for the items of the multi-choice questions. In the following we only present results from the statistical analysis that were significant enough. Our anonymized survey dataset, along with the full statistical analysis, is also published in the online appendix.

General information of the organizational unit revealed that the division between the 33 organizations that took part in the survey was very even; very small, small and medium-sized organizations represented each about 21 percent of the participants, while 36.4 percent were large or very large (more than 250 employee) organizations. Approximately eighty percent of the organizations were private companies, while rest of the participants were government agencies or nonprofit organizations. Organizations focusing mainly on national operations formed 21.2 % of the respondents while 39.3 % of organizations focused mostly on international business. Out of all organizations, 30.3 % of them were in-between national and international scale. Out of all organizations, 18.2 % also considered themselves solely or primarily as open source developers. Of the people who responded to our survey, a majority (66.7 %) considered themselves primarily as software developers, while 12.1% had a management position and 15.2% worked in quality assurance. As for the mission-criticality of the organizations, 51.5 % of the organizations reported that product fault could cause remarkable economic losses. Two of these organizations indicated that a fault in their product could cause a loss of a human life. The profiles of the respondent's OUs are shown in Table 1.

The use of testing tools was measured by the question, application level of different software testing tools, and changes were observed through comparison to the earlier results. In this survey, a tool was defined as "an application, framework, web service, extra library, feature of your development environment etc. whichever supports completing the mentioned task".

Table 2 presents the number of used tools is illustrated as percentages in 2017 and 2009. As observable, the three most popular tool categories include defect reporting tools, test automation tools and unit testing tools. Defect/code tracing tools are used by over half of all surveyed organizations. When comparing the new data with the 2009 data, the overall popularity of testing tools has increased in most categories, in particular, test automation, tracing tools and defect reporting. Since 2009, the popularity of test case management (for example, ticketing systems would also fall into this category) remains high, but is no longer the most common testing-specific tool.

TABLE I. THE PROFILE OF THE 2017 SURVEY RESPONDENTS (N = 33)

Category	% of respondents
Very Small organization (1-10 employees)	21.2 %
Small (11-50 emp.)	21.2 %
Medium (51-250 emp.)	21.2 %
Large or very large (250+ emp.)	36.4 %
Private company	78.8 %
Government or non-profit organization	21.2 %
Open source developer organization	18.2 %
Primarily national business/operations	30.3 %
Primarily service business	45.5 %
Primarily product business	39.4 %
Mission-critical organization (remarkable economic losses or loss of human life)	51.5 %

TABLE II. THE PERCENTAGE OF APPLIED TESTING TOOLS IN THE INDUSTRY

Tool	% of respondents	
	2017	2009
Bug/Defect reporting	72.7 %	22.6 %
Test automation	66.7 %	29.0 %
Unit testing	57.6 %	38.7 %
Bug/Code tracing	57.6 %	3.2 %
Performance testing	48.5 %	25.8 %
Test case management	45.5 %	48.4 %
Integration testing	45.5 %	16.1 %
Virtual test environment	42.4 %	12.9 %
Quality control	36.4 %	19.4 %
Automated metrics collector	36.4 %	3.2 %
System testing	27.3 %	9.7 %
Security testing	24.2 %	3.2 %
Test completeness	24.2 %	6.5 %
Test design	15.2 %	22.6 %
Protocol/Interface conformance tool	9.1 %	6.5 %

The second chapter of the questionnaire discussed the observed test and quality assurance process problems, identified originally in 2009 [4] supplemented with new questions related to maintenance issues. New maintenance and support questions were added because maintenance and support activities have continued growing and are responsible for a large amount of the total lifecycle costs [20]. The observations, especially when comparing the 2009 data with 2017, implied that the configurability of the testing tools has become an issue, and that the support for different software platforms might become an issue, when observing the trend of the changes. Additionally, feature development during late development phases shorten testing schedule and it has become an increasingly pressing issue. The detailed results containing the self-assessment figures for both 2017 and 2009 are presented in Table 3.

The third chapter of the survey was software processes and the amount of agile practices in the organizations. In the survey of 2009, the industry was observed to be interested in the introduction of agile and, in general, more informal practices. Based on our responses, the results of

TABLE III. SOFTWARE TEST PROCESS PROBLEMS, AS IDENTIFIED IN OUR 2017 SURVEY AND IN 2009 [9]. RESPONSES ARE ON A SCALE OF 1 TO 5 (1 FULLY DISAGREE - 3 NEUTRAL - 5 FULLY AGREE)

	2017 mode	2009 mode
Complicated testing tools cause test configuration errors.	4	1
Commercial testing tools do not offer enough support for our development platforms.	3	1
It is difficult to automate testing because of low reuse and high price.	4	5
Insufficient communication slows the bug-fixing and causes misunderstanding between testers and developers.	4	2
Feature development in the late phases of the product development shortens testing schedule.	4	4
Testing personnel do not have expertise in certain testing applications.	4	4
Existing testing environments restrict testing.	3	4

this chapter are very in-line with the earlier results giving emphasis on the agility of the industry-applied processes.

The industry drive towards agile practices can also be observed from another chapter in our survey where we asked about the use of formal process models such as SPICE (software process assessment, ISO/IEC 15504, currently part of the ISO/IEC 33000 series) [28] or software testing standard (ISO/IEC 29119) [29]. The question covered also the utilization of capability and maturity models, such as TMMi - test maturity model integrated [30] or CMMi – capability maturity model integrated [31]. Based on our survey results, the use of formal models have decreased within the last eight years. Some form of process model (formal or self-defined) was applied by 21.2 percent of organizations (62.5 percent in 2009), while none of the organizations in 2017 applied capability or maturity certificates in their organization (it was 43.8 percent in 2009). In 2017, V-model, acceptance criteria for tickets and "generic agile" were mentioned, all based on best practices collected from various sources and "self-defined". Detailed division of answers is presented in Table 4.

The final chapter in the survey included questions concerning the software testing and quality assurance practices. In general, the results do not indicate any major shifts in the applied testing and quality assurance practices between the two surveys. Organizational units are confident that they are building the product right, and at the same time, building the right product. Survey responses detailed in Table 5 highlights some differences between the surveys: Testing schedules may not be kept (2009 mode 4, partially agree, 2017 mode 2, partially disagree) and time is not necessarily allocated enough for testing (2009 mode 4, partially agree, 2017 mode 2, partially disagree). Respondents are less confident in their function testing practices (3.8 vs. 2.9 in average between 2009 and 2017. 2009 mode 4, partially agree, 2017 mode 3, neutral). Statistical significance in the difference of distributions between the years for the single question "our functional testing is excellent" can be established with the Mann–Whitney U test, $U=613$ at significance level $p=0.005$. Formal inspections are the testing practices on which the surveyed organizations have become more confident (2009 mode 2, partially disagree, 2017 mode 4, partially agree), while code review practices have become more varied between different organizations (2009 mode 4, partially agree, 2017 mode 1, fully disagree).

In addition to multi-choice questions the survey contained open-ended questions, where we asked the respondents to explain how their organization manages the increasing testing and maintenance effort. The following themes were highlighted from the responses:

- Moving from proprietary software to open source
- Increasing the coverage of automated tests
- Focusing on service scalability in design
- Re-implementing legacy applications
- Setting up dedicated testing and development environments
- Offshoring testing work
- Establishing pre-planned maintenance time for projects, during which last defects are fixed
- Forming dedicated maintenance teams
- Emphasizing the responsibility of current developers
- Employing a risk-based testing approach to cover the most critical components rather than trying to get perfect coverage.

TABLE IV. THE USE OF FORMAL PROCESS MODELS AND CAPABILITY OR MATURITY CERTIFICATES IN ORGANIZATIONS

Category	2017	2009
Process model - Yes, formal	9.1 %	25.0 %
Process model - Yes, informal	12.1 %	37.5 %
Process model - No	63.6 %	37.5 %
Capability certificate - Yes, formal	37.0 %	0.0 %
Capability certificate - Yes, informal	6.3 %	0.0 %
Capability certificate - No	56.3 %	81.8 %

TABLE V. THE SELF-ASSESSMENT OF THE QUALITY OF THE DIFFERENT TESTING AND QUALITY ASSURANCE PRACTICES (1 FULLY DISAGREE – 3 NEUTRAL – 5 FULLY AGREE)

	2017 mode	**2009 mode**
Our software correctly implements a specific function. We are building the product right.	4	5
Our software is built traceable to customer requirements. We are building the right product.	5	4
Our formal inspections are OK.	4	2
We go through checklists.	2	3
We keep code reviews.	1	4
Our unit testing (modules or procedures) is excellent.	4	2
Our integration testing (multiple components together) is excellent.	3	3
Our usability testing (adapt software to users' work styles) is excellent.	3	2
Our function testing (detect discrepancies between a program's functional specification and its actual behavior) is excellent.	3	4
Our system testing (system does not meet requirements specification) is excellent.	3	4
Our acceptance testing (users run the system in production) is excellent.	4	4
We keep our testing schedules.	2	4
Last testing phases are kept regardless of the project deadline.	4	4
We allocate enough testing time.	2	4

V. DISCUSSION AND IMPLICATIONS

The objective of this study was to explore the testing practices of software companies, compare the results with earlier survey result from the year 2009 and thereby outline changes in software industry. The collected data is publicly available in the online appendix should other researchers want to validate, replicate or build upon our findings.

Overall, the availability and application level of testing- and quality assurance-dedicated tools has increased across the industry, in almost all measured categories. Especially tools related to automated testing (e.g. test automation, automated metrics collection, performance testing, tracing tools) have increased significantly. The respondents of the survey refer to testing and automated testing almost synonymously. The available testing tools in 2017 are more sophisticated than in the 2009, imposing less restrictions but causing more configuration problems.

The use of different formal standards, certifications and process models has decreased, while the amount of agile practices has increased moderately. The mission-criticality of the software no longer limits the organization from using agile practices or other informal approaches. In 2017, the last product features are introduced later during the development process than in 2009. This leads to increased shortages of testing resources (time) and puts more emphasis on the acceptance phase testing. Test design and documentation work in general have declined while the confidence in functional testing practices has declined. Issues in testing and maintenance are more related to software development processes and practices, the quality and coverage of testing, and test schedule rather than the cost of quality assurance.

The survey results indicate increase in test automation, a shift towards agile practices, and that the formal software process models are less popular among industry practitioners. Results are in line with the observations of, for example, Khosla [19]: the rise of automation in testing, deployment and maintenance. Growing test automation also fits well to the observations of the Gartner report [20]. Explanatory factors to the growing test automation include, for example, agile methods with regression testing [32], continuous deployment and integration to shorten the timespan between product versions [33], DevOps to lower the threshold between development and use [34], and the general requirements for automation in IT-departments, server rooms and data centers to reduce the costs [19].

In comparison to other industry surveys in software testing, our results suggest similar trends as, for example, Canadian software industry report by Garousi and Zhi [17]. The most important testing tools in our study include defect tracking, unit testing and test automation, and Canadian organizations see functional and unit testing as the most common testing work. Likewise, Canadian organizations perform testing activities mostly during a dedicated testing phase in development (test-last approach). Our respondents did not suggest any other approach than test-last, and our results indicate that test phases may even be skipped in some circumstances.

Formal process models are more common in large and very large organizations. According to the study of Hardgrave and Armstrong [35], small and medium-sized organizations are able to apply the principles and best practices of the formal models in their work. Therefore, the reason for the decreasing use of the process models cannot be directly explained, and has to be assessed in more detail in the future works.

Concerning the validity of the study, even though the survey constructs and questions between the rounds were almost the same, there were differences in the data collection procedures: in 2009 the data was collected by interviewing representatives of software organizations whereas the 2017 dataset was collected online. The number of interviews in the 2009 dataset was 32 and the number of filled on-line questionnaires in the 2017 survey was 33. The response rate of 2017 was in line with the estimates given for on-line surveys [21]. The sample is small but comparable with the sample of 2009 and the observations are presented as explorative and not as strong conclusions. Overall, the metrics presented in this paper are accumulation data from the survey, so the researcher bias on the results should be minimal. The 2017 results were largely similar to the 2009 results, which adds to the rigor of the results, and helps highlight differences between the years.

VI. CONCLUSION

The results of the survey presented in this paper indicate that the software testing practices have undergone some changes in the industry within the last eight years. First, automation in testing has continued its growth. Within testing trends, automation has become more common on all levels of testing. Second, the application of formal software process models and capability maturity models seems to have decreased, while the testing tools have become increasingly common and more sophisticated.

This change is also reflected by the organizational considerations over the testing tools: the tools no longer restrict the organizational unit as much as they did in 2009 but in exchange, configuration problems and lack of platform support have become increasingly common. Also testing done during the design phase is decreasing. Since the last features are introduced later in the software development process, the emphasis on the late testing and, especially, acceptance testing has increased, while, at the same time, available time for testing work has decreased. Overall, the changes are not dramatic but the industry practices evolve as we can observe from the comparison of the surveys.

In our future work, the focus is on the expenses of testing and quality assurance. Based on our observations, the reduced use of formal processes, and the need to push new features into the product, mean that the products need better support for acceptance testing, regression testing and in general quality assurance for the features added after the initial launch. This study area is interesting, since the reduction of the costs of the maintenance cycle and automated regression testing would probably have a meaningful impact on the overall costs of quality assurance work.

REFERENCES

[1] E. Kit, *Software testing in the real world*. Addison-wesley, 1995.

[2] S. Planning, "The economic impacts of inadequate infrastructure for software testing," 2002.

[3] "World Quality Report 2016-17," Capgemini Group, 2017.

[4] J. Kasurinen, O. Taipale, and K. Smolander, "Software test automation in practice: empirical observations," *Adv. Softw. Eng.*, vol. 2010, 2010.

[5] O. Taipale, K. Smolander, and H. Kälviäinen, "Finding and Ranking Research Directions for Software Testing," in *Software Process Improvement: 12th European Conference, EuroSPI 2005, Budapest, Hungary, November 9-11, 2005. Proceedings*, I. Richardson, P. Abrahamsson, and R. Messnarz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 39–48.

[6] G. Paré and J. J. Elam, "Using case study research to build theories of IT implementation," in *Information systems and qualitative research*, Springer, 1997, pp. 542–568.

[7] N. K. Denzin, *The research act: A theoretical introduction to sociological methods*. Transaction publishers, 1973.

[8] S. P. Ng, T. Murnane, K. Reed, D. Grant, and T. Y. Chen, "A preliminary survey on software testing practices in Australia," in *Software Engineering Conference, 2004. Proceedings. 2004 Australian*, 2004, pp. 116–125.

[9] W. Chen, J. Li, J. Ma, R. Conradi, J. Ji, and C. Liu, "An empirical study on software development with open source components in the chinese software industry," *Softw. Process Improv. Pract.*, vol. 13, no. 1, pp. 89–100, 2008.

[10] R. Torkar and S. Mankefors, "A survey on testing and reuse," in *Software: Science, Technology and Engineering, 2003. SwSTE'03. Proceedings. IEEE International Conference on*, 2003, pp. 164–173.

[11] C. Ferreira and J. Cohen, "Agile systems development and stakeholder satisfaction: a South African empirical study," in *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, 2008, pp. 48–55.

[12] J. Li, F. O. Bjørnson, R. Conradi, and V. B. Kampenes, "An empirical study of variations in COTS-based software development processes in the Norwegian IT industry," *Empir. Softw. Eng.*, vol. 11, no. 3, pp. 433–461, 2006.

[13] P. Belt, J. Harkonen, M. Mottonen, P. Kess, and H. Haapasalo, "Improving the efficiency of verification and validation," *Int. J. Serv. Stand.*, vol. 4, no. 2, pp. 150–166, 2008.

[14] K. Y. Wong, "An exploratory study on knowledge management adoption in the Malaysian industry," *Int. J. Bus. Inf. Syst.*, vol. 3, no. 3, pp. 272–283, 2008.

[15] J. Iivari, "Why are CASE tools not used?," *Commun. ACM*, vol. 39, no. 10, pp. 94–103, 1996.

[16] V. Garousi and T. Varma, "A replicated survey of software testing practices in the Canadian province of Alberta: What has changed from 2004 to 2009?," *J. Syst. Softw.*, vol. 83, no. 11, pp. 2251–2262, 2010.

[17] V. Garousi and J. Zhi, "A survey of software testing practices in Canada," *J. Syst. Softw.*, vol. 86, no. 5, pp. 1354–1376, May 2013.

[18] J. Lee, S. Kang, and D. Lee, "Survey on software testing practices," *IET Softw.*, vol. 6, no. 3, pp. 275–282, 2012.

[19] V. Khosla, "Keynote speech," presented at the Structure Conference, San Francisco, 2016.

[20] "The Four Laws of Application, Total Cost of Ownership." Gartner, 2012.

[21] A. Fink, *How to Conduct Surveys: A Step-by-Step Guide*. Sage Publications, 2012.

[22] B. A. Kitchenham *et al.*, "Preliminary guidelines for empirical research in software engineering," *IEEE Trans. Softw. Eng.*, vol. 28, no. 8, pp. 721–734, 2002.

[23] "Webropol." [Online]. Available: https://www.webropolsurveys.com/. [Accessed: 22-Jun-2017].

[24] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2017.

[25] W. Revelle, *psych: Procedures for Psychological, Psychometric, and Personality Research*. Evanston, Illinois: Northwestern University, 2017.

[26] R. Bergmann, J. Ludbrook, and W. P. Spooren, "Different outcomes of the Wilcoxon—Mann—Whitney test from different statistics packages," *Am. Stat.*, vol. 54, no. 1, pp. 72–77, 2000.

[27] T. Metsä-Tokila, "Ohjelmistoala," Työ- ja elinkeinoministeriö, 2014.

[28] "ISO/IEC 15504-1: Information technology — Process assessment — Part 1: Concepts and vocabulary." International Organization for Standardization, 2004.

[29] "ISO/IEC 29119-2: Test Processes." International Organization for Standardization, 2013.

[30] E. van Veenendaal and B. Wells, *Test Maturity Model Integration TMMi*. The Netherlands: Uitgeverij Tutein Nolthenius, 2012.

[31] R. Kneuper, *CMMI: Improving Software and Systems Development Processes Using Capability Maturity Model Integration*. Rocky Nook, 2008.

[32] K. Schwaber and M. Beedle, *Agile software development with Scrum*, vol. 1. Prentice Hall Upper Saddle River, 2002.

[33] M. Fowler, "Continuous Integration," *martinfowler.com*. [Online]. Available: https://martinfowler.com/articles/continuousIntegration.html. [Accessed: 22-Jun-2017].

[34] S. K. Bang, S. Chung, Y. Choh, and M. Dupuis, "A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps," in *Proceedings of the 2nd annual conference on Research in information technology*, 2013, pp. 61–62.

[35] B. C. Hardgrave and D. J. Armstrong, "Software process improvement: it's a journey, not a destination," *Commun. ACM*, vol. 48, no. 11, pp. 93–96, 2005.