Test Tools: an illusion of usability?

Conference or Workshop Item

Test Tools: an illusion of usability?

Isabel Evans, Chris Porter, Mark Micallef
Faculty of Information and Communication Technology
University of Malta
Msida, Malta
{isabel.evans.17,chris.porter,mark.micallef}@um.edu.mt

Julian Harty
School of Computing and Communications
Open University
Milton Keynes, UK
julianharty@gmail.com

***Abstract*—Software testing is vital, yet expensive and time-consuming. This essential part of the software development process includes testers performing many repeated actions in test execution and management. Use of automation and tools could reduce costs and timescale, while providing consistency by removing human error during repetitive activities. Challenges for successful tools and automation adoption have been identified both in academic research and in industry practice, including technical, managerial, skills-related and usability issues. We set out to investigate what usability improvements would aid successful tool adoption, and discovered that usability, while a necessary attribute, is not sufficient to ensure success, and the belief in usability as a sufficient cure for automation shelfware might be an illusory phenomenon which disguises potential difficulties when using tools longer term. This *illusion of usability* includes a belief that UI attractiveness is sufficient for tool usability, a belief that testers come from a narrow group of personas, and a belief that skill levels and requirements for tools are static. This may lead to frustration for testers, and therefore reluctance to use tools and automation. We summarise our findings and outline our proposed next research steps.**

***Index Terms*—software testing, test tools, automation, usability, quality in use, user experience**

I. Introduction

Software is increasingly essential as a support to many aspects of modern life, and IT teams are challenged to deliver high quality software within reduced budgets and timescales [1]–[3]. To make decisions about software readiness, teams undertake and rely on software testing [4]. Testing is important, and also time-consuming, expensive, repetitive and difficult to do well, so software teams and their managers turn to automation and tools as a way to reduce costs and timescales. They hope to automate repetitive work, provide greater certainty about testing's outcomes, and be able to test in ways not possible without tools: *"no matter how valuable in-person testing is, effective automation is able to increase the value of overall testing by increasing its ... range"* [5].

There is within industry and academia some evidence that test automation and tooling is not always successful, and various challenges have been identified. These include: the "shelfware" phenomenon where tools are acquired but not used; problems with maintaining tests; challenges from (lack of) management support for the tools implementation project; and a shortfall in skills required to operate and maintain automation and tools [4], [6]–[10] A debate has started in the industry about whether tester skill sets need to be increased, or whether improving the usability of tools would aid adoption [11]–[13]. This discussion of tester skill sets was the initial impulse for the research described in this paper. Guided by the research question *What problems do testers experience with tools and automation?*, we examined the literature from industry and academic sources, interviewed experts, and surveyed test practitioners. We asked testers about their experiences and analysed whether they reported tool usability as a problem.

We present new data indicating that improvements in tools' usability, while aiding their initial learnability and interface aesthetics, did not always resolve difficulties when using tools longer term. Usability, while a necessary attribute, is not sufficient to ensure success: focusing just on learnability and the interface may disguise problems with other aspects of usability, or with other attributes of the tool, which cause difficulties when using tools longer term. It is not simply poor usability as a measurable attribute of the tool, it is the illusory expectation that addressing interface design and learnability is enough to ensure successful tool adoption.

The rest of the paper is organised as follows. The background to the study, including a literature review and definition of terms, is followed by a summary of the method adopted to collect and analyse data over a period of one year, then the results, discussion, conclusions and plans for future work.

II. Background

A. Literature Review Sources

We used two main sources for the literature review: academic research and industry authors. We revisited industry "classics" and read blogs and tweets by industry practitioners. We used Google Scholar to search for papers covering test automation and tools, selecting papers with industry settings.

B. Definition of Usability, Quality in Use and User Experience

The Human-Computer Interaction (HCI) between a person and the technology they are using is not just about the User Interface (UI) design, or even about the usability of the technology. The UI provides the means to allow communication between a person and the technology supporting them. Whether a tool meets the needs of a user (its utility) combines with its usability to make the tool useful. Alternative definitions and ways of measuring usability are offered by different standards for example ISO 9241 and ISO 25010 [14], [15], and it is a *"difficult-to-grasp concept"* as discussed by Speicher

 DOI:10.1109/ICSTW50294.2020.00070

[16]. For the purposes of this paper we define usability as in ISO 9241 in terms of user goals, user effectiveness, user efficiency, and user satisfaction in a specified context of use, which contributes to the definition of "Quality in Use" in ISO 25010. Quality in Use includes effectiveness and efficiency, as well as other attributes such as flexibility and freedom from risk, that affect the user's experience of a software product. Quality in Use contributes to User eXperience (UX) which "encompasses all aspects of the end-user's interaction with the company, its services, and its products" [17] so that the product meets the needs of the customer, "without fuss and bother" [17], as do all the interactions between the software provider and the customer. The attributes discussed in a UX context are human-focused: for example, trust, flow and credibility. The sub-attributes of usability that we consider in this paper, derived from ISO 9241 and ISO 25010 are: user goals, freedom from risk, context of use, learnability, user error protection, operability, and satisfaction.

C. Test Tools and Test Automation

We define "test tools and automation" as "any tool used to support any testing activity", and in data collection asked open-ended questions that allowed participants to include as a "tool" or "automation" anything they felt was in scope.

D. Known challenges to test automation

The literature review identified a number of known challenges that Wiklund [9] describes as "Impediments to Test Automation". That study classified problems reported on a user discussion board for test automation, and made a survey of the users of the discussion board. The study revealed that user errors, usability, and user behaviour all appear to be significant factors, and this informed our research design (refer to section III). Several authors identify technical and organisation reasons for tools not being adopted, and problems with bias in tool selection leading to unsuitable tools being acquired and then not used (for example [4], [8], [9], [18]).

E. Usability challenges in related disciplines

We found research uncovering challenges in several related areas. Work examining programmers' use of static analysis tools [19], uncovered usability challenges, for example, one participant in that study says: *"... none of these [messages from the tool] really help me"*.

A usability analysis of visual programming environments (VPE) [20] provided insight into the challenges faced by novice and expert users of VPEs, and in particular looked at the experiences of end-user programmers. In examining the cognitive load on programmers, the authors simply say: *"The demand placed on working memory . . . is just too much"* [20]. Additionally, this study indicated that viscosity (how easy it is to change the code) is a usability problem arising from the VPE design and cognitive load on the programmer. Viscosity is a usability characteristic that affects maintainability of code; for test tools and automation, viscosity affects the maintenance of tests and scripts. A companion article [21] questions the use of visual interfaces for novice programmers, with evidence that expert programmers find visual interfaces easier than novice programmers. As the users and scripters of test tools will also range from user-testers to expert coders, it seems likely that the same range of cognitive challenges might exist for testers.

In a study of the qualitative methods and tools used in research, the authors discussed *"seamful tool support"* [22] that is, the work overhead resulting from needing multiple tools to interact in order to complete a task, noting it as a significant challenge. Like research, testing involves multiple tasks, and testers use multiple tools to aid them when executing tests, managing bug reports, or reporting results [1]. These tools and activities need to integrate with each other and with development activities and tool sets. Seamful versus seamless integration of tools is relevant for test tools and automation.

F. Usability of testing tools and automation

For a testing tool to be useful, it needs a good UI, good usability and good utility. The context of use for testers is often challenging. Frequent changes within a software development or maintenance project mean that tests also change constantly. Therefore, for a testing tool to have utility long-term, it needs to be flexible, reducing viscosity by enabling changes easily to be made to the tests [8], [23]. Quality in Use, as defined in ISO 25010 [15], includes flexibility; for test automation and tools, this is built on software attributes such as maintainability, portability and compatibility.

As discussed in a grey literature review by Raulamo-Jurvanen, [18] perceived usability and perceived popularity of a tool are two factors in choice, rather than the suitability of the tool for a particular context. At industry conference expos, we observe that some tool vendors are starting to address the perceived shortcomings in terms of tool usability, specifically in terms of learnability and interface design. Additionally, work is happening within the industry to increase skill sets, for example the Test Automation University [24].

Our own published work on the experience of testers has yet to cover the usability of test tools and automation [5], [25]–[28] although Harty [5] comments on the skill sets required, and lacking, for testers to engage with tools. The range of testers using tools, the range of tools, and what we mean by a "good" UI, usability and utility will be dependent on the person, their context, and their goals [14], which could be defined in tester personas. We have work in progress on Tester Personas, a Tools Taxonomy, and on the Lived Experiences of Testers (for example [29]), which we hope will help bring clarity to the range of requirements for tools.

III. Method

The anonymized dataset with all survey respondent quotes is available as an OSF project [2]. Figure 1 shows the outline of the research method. Following preliminary observations during conversations at conferences, interviews with testing

[1] Authors' industry experiences

[2] https://osf.io/9qsgd/?view_only=2ac1248e7fd645339df2b66553d384fb

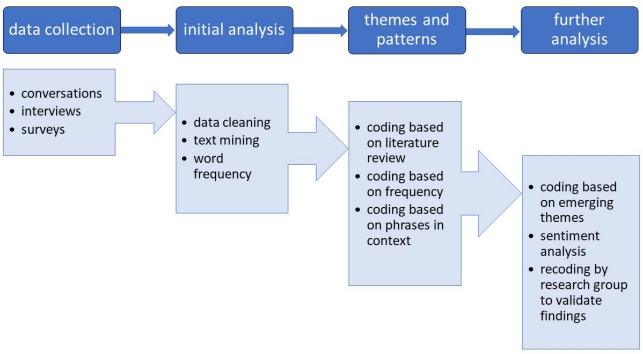

Fig. 1: Research Method Flow

and automation experts, and the literature review, which informed our research question, our work consisted of a series of studies via surveys. Four surveys were made during 2018, via industry conferences. The question sets included open ended questions about the person: *tell me a bit about yourself*, and about experiences with tools and automation: *tell me a story about an experience with automation / tools.* Two of the surveys were preceded by workshops for the survey participants. The entire corpus of data arising from these surveys consisted of 180 people's responses. Those responses which did not provide any data for the *tell me a story* question were removed, leaving a total of 111 survey responses. These per-person records showed a range of responses and were analysed as one group. Additionally, edited transcription [30] was carried out on recorded expert interviews, and key points in conversations were documented, and while these were not used in the frequency analysis, relevant quotes from both expert interviews and conference conversations are included.

Text mining on the survey data was carried out as part of the exploration phase, including a word frequency exercise to determine a high-level weighting of each overarching theme, based initially on categories identified in the literature review ("Managerial/organizational", "Technical", "Usability"). We counted positive comments, issues and challenges raised under each code. ISO 25010 software attributes were used to categorise perceived attributes of a good tool (see section IV-B, Table I, [14]). Participants for this research are unlikely to be usability experts, so express usability concerns in everyday language such as "easy to use" and "difficult". We looked for these everyday expressions when coding. We collected mentions of roles, test stages, tools and technology, as well as software attributes including usability. We performed several iterations of coding, starting with themes identified in the literature review, to identify perceptions, challenges, and commonly held ideas of what is important for successful tool adoption. As well as coding, we used MS Azure Sentiment Analysis to assess the data. As a validity check, we assessed the data to look for patterns and correlations within and between respondents, revisited the text of the answers to check, reclassify, and recode if needed. Finally, we discussed our findings with a group of ten software testing industry experts.

IV. Results

In this section we discuss the results of the data analysis, participant profiles, themes and patterns that emerged.

A. Participant profiles

Full details of the participants' profiles is in the OSF project[2]. Participants came from different countries and cultures, including UK, NZ, Australia, Thailand, Germany, China, India, Sweden, and the USA among others. There was a mix of ages, gender, lengths of work experience, and of permanent, contract, and freelance participants. A wide range of professional backgrounds was noted, from software engineering degrees to carpentry. While 56 participants described themselves as testers, and 81 as technical/automators, job titles and roles included some quite complex mixes of responsibility and specialisms. These included a *"head of testing (and) tester/automator"*, a *"project manager whilst also tester (manual and strategy)"* and another listed responsibilities including *writing UI tests, security scans and writing applications*; 74 described themselves as leaders, experts or teachers. Participants worked in a wide range of industries including financial services, healthcare, avionics, education, and retail.

B. Results of Analysis

Analysis included text mining, frequency analysis and qualitative analysis with coding against themes both from the literature review and emergent from the data. We coded by looking for technical attributes from ISO 25010, including both positive comments, issues and challenges, as well as within the organisational and usability themes. Table I shows the codes, example words and phrases, and their frequencies in the data. While participants did not always directly mention usability, their responses indicate usability as either a desirable characteristic of a tool, or the lack of usability as a challenge to tool usage. Fifty-three participants indicated usability as a desirable feature of a tool, for example:

> *"that I can quickly get to testing without having to waste time learning the tool, or how the tool wants me to do it"*
>
> *"supports your workflow vs forcing you to change. People [are greater than] process/tools"*

Quality in use and usability words, phrases and comments were the largest group, followed by technical points. Eighty-two respondents reported issues and challenges, some with multiple issues. Portability, performance and maintainability of the tests and the automation were the most frequently mentioned of the technical quality attributes.

If we look at the number of participants raising issues which have a usability basis, and compare that with the number of participants raising other technical issues with tools and automation, we find that 67% of the participants commented more on usability than technical attributes of the tools, and 23% commented more on technical than usability attributes. The remainder commented evenly on usability and technical points. Some participants raised multiple issues. In the data,

TABLE I: Results of analysis

Code	No. Participants	Frequency	Examples of comments
Quality-in-Use / Usability			
Operability	93	250	Easy, difficult, supportive, usable, help, learn, simple, UI, looks cool, user, suitable, friendly, complete, flexible
Learnability	54	125	
User goals	32	97	
UI aesthetics	26	33	
Context	22	34	
Satisfaction	12	24	
Total	n/a	511	
Technical			
Portability	57	138	Technical environment, installation, running out of resources, access, integration with other tools
Performance	50	102	
Maintainability	46	68	
Functionality	47	57	
Security	26	46	
Compatibility	28	39	
Reliability	12	16	
Total	n/a	466	
Management / Organisational			
Total	101	377	Motivation, value for money, time/staff/budget constraints, poor service from vendor
Issues and challenges			
Total	82	232	82 respondents raised issues and challenges, 29 did not raise issues or challenges

there is not a strong pattern about this; some participants raised both usability and technical issues, some raised one or the other, and some raised no issues at all. Some participants reported a mix of usability and technical issues.

V. Discussion

Our findings both corroborate existing findings from other authors, and also uncover new findings.

A. Corroborating existing results

In the literature review, we noted that the viscosity of interfaces in visual programming languages adversely affects the maintainability of code [20]. We speculated that the same viscosity problem could arise in test tools and automation. Some of our participants reported test maintenance as a problem, when needing to make fast and frequent changes as new tests are identified, for example:

> *"... created a library script ... then duplicated it into an execution set ... then added some more lines ... I had thought of testing ... [the tool] then decided to add some (not all) of the new lines ... my nice neat customer account data no longer corresponded ... I actually couldn't delete it either. It's been passed back to the vendor who can't understand how I did it..."*

Testers will constantly change and update tests: the inability of a tool to easily support maintenance is a viscosity problem.

Additionally, demands on working memory, and "seamful" tools [20], [22] appear to be problems testers experience with their automation and tools.

> *"I am fortunate (!) to be an administrator for one of our [tools] and received a request to add some new custom fields to one of the projects in that tool. I first had to define the fields (name, data type, etc.) Then somewhere else in the admin UI, I had to configure where this field would appear on the test case form for the project. Then somewhere else again in the admin UI, I had to define the set of possible values for the dropdown fields I'd added. [...] infuriating (and requires a re-learn [of] this ridiculousness every few months when I get such admin requests)."*

This tester experienced unnecessary steps during updates, viscosity, seamfulness, and maintenance problems, leading to a high cognitive load and frustration.

Neglecting design of the tool installation, set up and support causes testers frustration and usability issues, for example:

> *"every time I have to deal with a new tool, it's the matter of installation that is the most difficult. For some reason, everybody who develops tools prepare YouTube video[s] about how to use the tool but not how to install it."*

This is also reflected in the researchers' industry experiences, where tools designed for UI usability did not necessarily have support for easy installation, configuration, and (where applicable) integration with other tools.

Usability issues were raised by 34 people across our surveys, as well as by [9] as an *impediment* to test automation.

B. Emerging theme: Illusion of usability

Usability of testing tools appears to be a necessary but not a sufficient factor for success. It also is perhaps sometimes superficially applied. We identify three ways "usability" potentially appears to be misapplied: (1) over-focusing on attractiveness over usefulness, (2) only focusing on one user group, resulting in learnability and flexibility being in opposition, and (3) neglecting to support change and growth for the personas and their requirements.

1) Usability focused mainly on an attractive interface: One expert interviewee raised the point that usability and a pleasing UI, together with online help for the tool, makes it marketable. Survey respondents reported that the initial adoption/learnability looks easy, but this may disguise longer term problems with maintainability, configuration, portability and integration with other tools:

> *"running the tests is quite easy ... The difficult part is maintaining the tests when it grows massively."*
>
> *" ...looks cool ... but... took time to set up, lack info online [sic], user-unfriendly UI in configuration. Not all configured things worked."*

This might result in tools that cause work-related friction.

2) Toolset designed with only one user group in mind: Ignoring the needs of multiple important stakeholders and users narrows the usability of the tool. In the survey responses, both more technically able and less technically able testers indicated usability problems:

> *"some of the early design choices were made to make it easier to use for less technically competent*

testers. For those of us with strong coding backgrounds, it can occasionally be difficult to accomplish what we want."

"It was a lot of effort learning about the tool. The tool was initially built for developers with a small element for testers."

These two participants demonstrate that multiple stakeholders may use the same tool.

3) Neglecting to support change and growth: Focusing on the UI, initial learnability, and a narrow persona definition leads to what we call a "pianola versus piano" effect: to become accomplished at using a pianola, one does not need to learn to play, but are limited by the available music rolls, however one can start to make music quickly. To play a piano well takes time and effort, and at first one's repertoire is limited. Over time, one's repertoire and the ability to learn new tunes, and improvise well increases. In the same way, more than one of the survey participants commented that the most flexible and powerful tools took longer to become accomplished at using well. Their full usefulness was not apparent to first time users. If testers are to be pianists rather than pianola users, management has to support longer learning time, more training, coaching and support, than is sometimes budgeted for (commented on by [10] and our expert interviewees). Just as not all pianists are concert pianists, and not all pianists could improvise a cadenza in a concerto, some testers require more support and guidance from their tools than others.

C. Usability is necessary but not sufficient

We note that usability is a necessary factor for long-term successful tool and automation adoption. However, the overall quality in use of the tools is affected not just by usability but by quality in use attributes such as flexibility (built from reducing viscosity and increasing maintainability of the code) as well as technical attributes such as installability, performance, and portability. For testers involved in our studies, success with tools and automation is dependent on more than just usability: 31 respondents raised multiple issues with tools and automation, including lack of support for tools from management and suppliers, lack of skills and direction changes.

The expert interviewees all discussed the need for discipline, roles and rigor in both test design and automation, and that the *craftsman's process* applies to two complementary, distinct crafts: designing tests, and coding automation. One talked about automators serving the testers: *"[their] job is to press the keys for me"* This was echoed by an expert automator:*"I was just a conduit for [the Subject Matter Expert] to run their tests"*, who also discussed the need for treating tool and automation projects as difficult software development projects. The automation *"has to dance along with the system under test ... it's incredibly complex."* Tools / automation builders need to stop, analyse and *"test [the automation]"*.

Test automation and tools projects can succeed. Successes were reported in our data, and the online community is buoyant and publicly celebratory of their successes and enjoyment of automation (see for example both blog and comments at [31]). However, the proportion of people who did raise usability as a challenge during our research indicates the need for further study, and guidelines for tools and automation designers.

D. Potential validity problems

An external validity threat exists from convenience sampling via the industry conference community. This was mitigated by sampling across countries, online, allowing greater random access to the surveys, then selecting data with rich and meaningful stories about tools. An internal threat to validity is linked with researcher bias during the coding process. To mitigate this, the emerging codes were iteratively reviewed by the team, and wherever necessary, codes were modified, merged or further refined.

VI. Recommendation, Future Work and Conclusion

Our recommendations to tools and automation designers and developers are to use usability review and testing methods. A simple way for tools and automation providers to overcome some of these issues would be to apply Nielsen's Usability Heuristics [32] to assess how the tool might be experienced by the tester. Developing persona lifecycles would clarify who will use or be affected by their work. We encourage looking beyond dermal-level UI efforts, by considering core work processes and flows (related to primary tasks) afforded by tools, taking into account evolving configurations, testers' expertise, project sophistication and workflows – over time.

Our recommendations for future work by academia are to provide guidelines to tools designers to support their adoption of UX methods, their testing of tools multiple software attributes, and overcoming organisational challenges. It would also be interesting to understand how much academic research into techniques and tools has penetrated industrial practice, and what would aid industrial practice improvement, whether in test design, metrics, test execution or other activities. Another question is whether usability problems for testers' tools are different to those of other tool sets. This is an area for investigation; it may be that by resolving problems in our own toolsets we are better able to resolve usability problems for other people.

We propose taking these findings into industry, via workshops, presentations, and our industry social media network. This includes feeding back to participants in the data collection. In our next research stages, we aim to produce a taxonomy of testers, leading to persona definitions, as well as a taxonomy of tools to help us understand the range of tools used by testers. By linking the two taxonomies together we hope to understand who uses which tools, to support what testing activities and techniques.

In this paper, we have explored what perceptions and problems testers encounter when using test automation and test tools, specifically the perceived importance of usability of these tools. During our exploration of the problems, our data corroborated other authors, and led us to consider quality in

use and user (tester) experience of the tool. Test automation and tools projects can succeed. Yet, the proportion of people who did indicate usability as a challenge during our research indicates the need for further study, and perhaps guidelines on usability and quality in use for tools designers, vendors and test automators. We contribute new findings, namely that stakeholders for automation and test tools, whether tools designers, automation specialists, commissioning/purchasing managers, or testers, may hold an *illusion of usability* that could work against long term successful test automation. Focusing more on UX methods including personas and testers' journeys and work flows, contributes to understanding both what is required to enhance testers' skill sets and to improving tools adoption.

Our findings on the illusion of usability as an impediment to tool adoption, together with the call in Wiklund [9] for *"investigations into how to systematically prevent commonly occurring impediments [and package these solutions in a way] that is attractive, available, and useful to practitioners"* leads us to recommend that tools and automation designers and builders take more account of usability and quality in use. This is not just a superficial attention to the UI of the tool. Applying usability and quality in use attributes to understanding test tools and automation is not simply a cosmetic way to make the tools look more marketable, but also a way to enhance the use and life span of the tools to the benefit of individual testers, their teams, their customers and their organisations.

REFERENCES

[1] R. Hubig and I. Morschel, *Quality and Productivity Improvement in Object-Oriented Software Development*. Wiesbaden: Deutscher Universitätsverlag, 1997, pp. 3–13. [Online]. Available: https://doi.org/10.1007/978-3-322-99929-0_1

[2] H. D. Mills and R. C. Linger, "Cleanroom software engineering: Developing software under statistical quality control," *Encyclopedia of Software Engineering*, 2002.

[3] R. J. Leach, *Introduction to software engineering*. Chapman and Hall/CRC, 2018.

[4] ISTQB, "Downloads - ISTQB® international software testing qualifications board," 2018. [Online]. Available: https://www.istqb.org/downloads/send/51-ctfl2018/208-ctfl-2018-syllabus.html

[5] J. Harty, "Finding usability bugs with automated tests," *Communications of the ACM*, vol. 54, no. 2, pp. 44–49, 2011.

[6] P. Havelock. (2000) Testware or shelfware - the reality. [Online]. Available: https://www.bcs.org/membership/member-communities/software-testing-specialist-group/the-tester/newsletter-archive/testware-or-shelfware-the-reality/

[7] C. Kaner, "Avoiding shelfware: A managers' view of automated gui testing," in *STAR'98 Conference, Orlando, FL.*, 1998. [Online]. Available: http://www.kaner.com/pdfs/shelfwar.pdf

[8] D. Graham and M. Fewster, *Experiences of test automation: case studies of software test automation*. Addison-Wesley Professional, 2012.

[9] K. Wiklund, "Impediments for automated software test execution," Ph.D. dissertation, Mälardalen University, 2015.

[10] S. Gamba and D. Graham, "Test automation patterns," 2018. [Online]. Available: http://testautomationpatterns.org

[11] E. Hendrikson, "Do testers have to write code?" 2010. [Online]. Available: http://testobsessed.com/2010/10/testers-code

[12] R. Lambert, "Why testers really should learn to code," 2014. [Online]. Available: http://thesocialtester.co.uk/why-testers-really-should-learn-to-code/

[13] M. Gill, "Testing skills that you must have: An overview." [Online]. Available: https://club.ministryoftesting.com/profiles/blogs/software-testing-skills-that-you-must-have-an-overview

[14] ISO, "ISO 9241-11:2018 Ergonomics of human-system interaction - part 11: Usability: Definitions and concepts," 2018. [Online]. Available: https://www.iso.org/obp/ui/iso:std:iso:9241:-11:ed-2:v1:en

[15] ——, "ISO/IEC 25010:2011 systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models," 2011. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

[16] M. Speicher, "What is usability? a characterization based on ISO 9241-11 and ISO/IEC 25010," *arXiv preprint arXiv:1502.06792*, 2015.

[17] D. Norman and J. Nielsen, "The definition of user experience (ux)," 2020. [Online]. Available: https://www.nngroup.com/articles/definition-user-experience/

[18] P. Raulamo-Jurvanen, M. Mäntylä, and V. Garousi, "Choosing the right test automation tool: a grey literature review of practitioner sources," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2017, pp. 21–30.

[19] B. Johnson, Y. Song, E. Murphy-Hill, and R. Bowdidge, "Why don't software developers use static analysis tools to find bugs?" in *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013, pp. 672–681.

[20] T. R. G. Green and M. Petre, "Usability analysis of visual programming environments: a 'cognitive dimensions' framework," *Journal of Visual Languages & Computing*, vol. 7, no. 2, pp. 131–174, 1996.

[21] M. Petre, "Why looking isn't always seeing: readership skills and graphical programming," *Communications of the ACM*, vol. 38, no. 6, pp. 33–44, 1995.

[22] A. M'manga, S. Faily, J. McAlaney, C. Williams, Y. Kadobayashi, D. Miyamoto, U. Poole, and U. Porton Down, "Qualitative adaptation: informing design for risk-based decision making," in *Proceedings of the 32nd International BCS Human Computer Interaction Conference*. BCS Learning & Development Ltd., 2018, p. 216.

[23] B. Dijkstra, "Three things everybody should know about test automation," 2016. [Online]. Available: https://www.linkedin.com/pulse/three-things-everybody-should-know-test-automation-bas-dijkstra

[24] Applitools, "Test automation university." [Online]. Available: https://testautomationu.applitools.com

[25] A. Borg, C. Porter, and M. Micallef, "Poster: Is Carmen better than George? testing the exploratory tester using HCI techniques," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2. IEEE, 2015, pp. 815–816.

[26] M. Micallef and C. Porter, "HCI - the tester's new sidekick?" 2016, the Tester Magazine – BCS 03/2016 (pp 12 -15). [Online]. Available: https://cdn.bcs.org/bcs-org-media/2742/tester-2016.pdf

[27] M. Micallef, C. Porter, and A. Borg, "Do exploratory testers need formal training? an investigation using HCI techniques," in *2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2016, pp. 305–314.

[28] M. Micallef and C. Porter, "Help! I'm only human!" 2017, keynote at the British Computer Society SIGIST Testing Conference, March 14, 2017, London, UK. [Online]. Available: https://cdn.bcs.org/bcs-org-media/2743/tester-2017.pdf

[29] I. Evans, J. Harty, C. Porter, and M. Micallef, "Stuck in limbo with magical solutions: The testers' lived experiences of tools and automation," in *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. HUCAPP, 2019.

[30] S. Team, "Types of transcription," 2019. [Online]. Available: https://www.speechpad.com/blog/types-of-transcription

[31] B. Dijkstra, "Why test automation is like bubble wrap," 2019. [Online]. Available: https://www.ontestautomation.com/why-test-automation-is-a-lot-like-bubble-wrap

[32] J. Nielsen, "Ten Usability Heuristics," 2019. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics

ACKNOWLEDGMENT

The authors thank the TAICPART paper reviewers for their useful and stimulating comments. We thank the expert testers and automators we interviewed, and our survey and workshop participants. We thank the following for supporting data collection and discussing results: UKSTAR, EuroSTAR, ODIN-

DAYS, STARWEST, STAREAST, STARCanada, Women Who Test, ANZTB, UK Software Testing Retreat.