Stuck in Limbo with Magical Solutions: The Testers' Lived Experiences of Tools and Automation

Isabel Evans[1][a], Chris Porter[1][b], Mark Micallef[2][c] and Julian Harty[3][d]

[1] *Department of Computer Information Systems, University of Malta, Malta*

[2] *Department of Computer Science, University of Malta, Malta*

[3] *School of Computing and Communications, Open University, Milton Keynes, U.K.*

Keywords: Software Testing, Automation, Tools, HCI, User Experience, UX, Lived Experience, Usability, Industry.

Abstract: The automation of people's roles at work brings changes to their lives and work, bringing advantages of increased effectiveness and efficiency, yet potentially life-changing effects, including redundancy. The software industry's purpose is to automate people's tasks and activities, and this applies also to jobs within the software industry, including teams who specialise in testing software. Test automation projects are not always successful, and our research initially set out to discover whether the challenges were usability-related, and whether HCI methods could help improve tools. We discovered a much richer story, which told of emotional stresses and life experiences within the software testing community. We discuss how automation, with all its benefits, affects motivation, causing disassociation of testers from their roles, and affecting their job-task mix. We show reasons why software test automation affects testers. Finally, we set out our position for our research about the lived experience of software testers using automation, which we are calling *TX: The Testers' Lived Experiences of Tools and Automation*, and argue that the effect of automation and tooling on testers' lived experience and its effect on their motivation is an area of study worthy of research.

1 INTRODUCTION

The automation of people's roles at work brings changes to their lives (Groover, 2019). These include increased effectiveness and efficiency in their work, potentially saving money and time for the employing organisation, as well as providing gains in productivity, and consistency of output. Should we additionally (even instead) focus on the effect on the person's job, how it changes that, and the experience of the automation as it affects people's lives? When those jobs are within the software industry, where the purpose of the industry is to automate people's tasks and activities, the effects of automation on the individual may still be the same as for any other worker. One group of workers within the software industry are the teams who specialise in testing software. Aspects of their roles are subject to automation projects.

In this paper we set out our position for our research about the lived experience of software testers of automation, which we are calling *TX: The Testers' Lived Experiences of Tools and Automation*, and argue that the effect of automation and tooling on testers' lived experience and its effect on their motivation is an area of study worthy of research.

2 BACKGROUND

Software is ubiquitous and relied upon. It is essential that software is tested adequately before it is used. This testing is done by developers and by users of the software, however testing is also done by specialist individuals and teams who focus on testing.

2.1 Software Testing

Software testing is an activity intended to give stakeholders information about the quality of software or services, and includes the planning for, design of, execution of, and analysis of the results of investigation into the behaviour and characteristics of a

[a] https://orcid.org/0000-0003-3913-4463

[b] https://orcid.org/0000-0003-3813-2941

[c] https://orcid.org/0000-0003-4159-6923

[d] https://orcid.org/0000-0003-4052-0054

Evans, I., Porter, C., Micallef, M. and Harty, J.
Stuck in Limbo with Magical Solutions: The Testers' Lived Experiences of Tools and Automation.
DOI: 10.5220/0009091801950202
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)*, pages 195-202
ISBN: 978-989-758-402-2

product or part of a product (ISTQB, 2018; Reckless, 2017a). Testing is hard to do well, expensive and time-consuming but takes place in the context of projects where quality must be maintained in the face of reduced timescales and budgets (F. Lehner and Abran, 2013; Jones, 2015; Tassey, 2002). Testing is hard because it includes challenging cognitive activities that require creativity such as threat modelling within complex domain and system spaces (Bach and Bolton, 2016; Bolton, 2016; Toledo, 2017).

2.2 Software Test Automation

Aspects of testing are repetitious or difficult for people to do well and these are candidates for automation (ISTQB, 2018). Test automation is the use of tools to aid parts of the test activity. That can be for example, repetitious execution of tests, checking the results of testing, aspects of test planning and test reporting (ISTQB, 2018).

Test automation projects are not always successful, causing both practitioners and academics to debate the reasons and potential solutions for automation project failures (Fewster and Graham, 1999; Graham and Fewster, 2012; Kaner, 1988; Wiklund, 2015). A debate within the software testing community asks whether the causes of failure include testers' lack of adequate knowledge and technical skills, or whether the tools are inadequately designed, requiring unreasonable cognitive effort from the testers and slowing them carrying out their challenging tasks (Hendrikson, 2010; Lambert, 2014; Reckless, 2017b).

2.3 Usability, User Experience and Lived Experience

The interaction between a person and the technology she is using is not just about the design of the user interface (UI), or even about the usability of the technology. The UI provides the necessary affordances which support interaction between a person and the technology. The UI and interaction design contribute to usability. Whether a tool meets the needs of a user (its utility) combines with its usability to make the tool useful.

For the purposes of this paper we define usability as in ISO 9241 (ISO, 2018) in terms of user goals, user effectiveness, user efficiency, and user satisfaction in a specified context of use. Usability contributes to user experience, which also includes the quality of all the interactions between the software provider and the customer: "*[User experience] encompasses all aspects of the end-user's interaction with the company, its services, and its products [so that the product meets the needs of the customer] without fuss and bother*" (Norman and Nielsen, 2019). User experience attributes are human-focused, including trust, flow and credibility, and these contribute to the lived experience of the user. In considering 'lived experience', we look at "*repercussions extending into the users' daily lives and away from the technology itself*" (Porter, 2015), and this includes the emotional impact of the technology. Thus, the UI design contributes to usability, and usability of software contributes to but does not encompass user experience. User experience in turn contributes to and is part of the lived experience. The emotional content of the experience can be supported by a great interface design, but all the circumstances of a person's interactions with the software, its provider and the people around them will also contribute to their emotional response.

2.4 Our Research to Date

Based on the research question: "What are the experiences of testers with automation?", we are exploring the interactions of testers and their automation tools, seeking to understand what problems hinder successful tool adoption.

Following a literature review and industry experience, our hypothesis at the start of the study was that the challenges would fall into technical, organisational or usability themes (Fewster and Graham, 1999; Graham and Fewster, 2012; Wiklund, 2015). Examples of technical challenges include issues with the IT environment, IT security, the tool's performance, and execution of tests. Examples of organisational challenges include process issues, management support, staff availability, and testers' skill sets (Gamba and Graham, 2018b). Examples of usability challenges include the automation not supporting the testing workflow, poor learnability and poor operability for the end user (in this case, the end user is the tester using the automation).

2.4.1 Mixed Methods Approach

A mixed-method approach was adopted to collect primary data from practitioners in the field. Semi-structured interviews and surveys were used to learn more about testers, their skill-sets and experience with test tools and automation. Qualitative methods were adopted to deconstruct personal experiences or stories with test tools and automation, allowing for a richer understanding of the phenomena being investigated. We asked testers to "tell us a story about your experiences with test tools and automation." We expected in our study to uncover issues with skill sets and with usability to help us examine the skills/usability debate, provide evidence for which was the most important contributory factor to automation success, and to allow us to suggest potential solutions. Responses collected so far from from 117 people via surveys and interviews were used to feed thematic analysis initially based on the findings from our literature review, while additional themes were added as they emerged from data analysis. The data we have gathered so far indicates patterns and themes that confirm findings from other industry writing and academic research on technical, organisational and usability impediments to test automation (Graham and Fewster, 2012) and (Wiklund, 2015) but also our data indicates that the relationship that testers have with their tools is more diverse, richer, and affords more emotional experiences than previously reported. This has led us to think about the purposes and effects of automation, not just in terms of productivity gains, but in terms of the lived experience of software testers of their tool set and automation as a significant challenge to successful test automation projects, and ultimately to the sustainable well-being of those testers.

3 DISCUSSION

The purpose of the software industry is to provide benefit to its stakeholders by automating activities of all kinds. When we start to apply tools and automation to activities within the software development sphere, then the focus of the automation is the change or removal of tasks carried out by software professionals. This includes software testing; when testing is automated, then the people who test software are subject to the same benefits and threats as anyone else whose role is subject to an automation (software) project. We argue that automation, as it affects the people whose tasks are being automated, is worthy of discussion, and that this includes the roles within the software industry. It is reasonable to ask questions about the lived experience of staff within organisations, and that executives, managers, owners and shareholders be concerned with the emotional welfare of their staff, as well as with profit and productivity. In examining the automation and tooling used by software testers, concern for their motivation and happiness, as well as their efficiency and effectiveness, is humane as well as contributing to a healthy organisation. Motivation studies (Reid, 2015; Warden and Nicholson, 1996) show that testers are more productive when motivated.

The data we have collected so far based on the question "*Tell me a story about an experience with test automation*" indicates a richer and more nuanced relationship between testers and their tools than we had expected. We received responses that were more emotional, more about the user experience of the tools, and more about the lived experience than we had expected. In this paper, reporting interim results, and claiming that the lived experience of testers is an important factor to consider when attempting to understand the success of and impediments to adoption of test automation and test tools, we divide our discussion into five areas. (1) We confirm that there are benefits to automation and tools for software testing. (2) We consider the effect of automation and tools on motivation of software testers. (3) We discuss whether tester skill sets or usability improvements are the solution to tools implementation problems. (4) We confirm that there are other problems that affect the success of automation and tools adoption for testing. (5) We demonstrate that the lived experience of testers is important, at the least to the testers, but also we suggest to their managers, team mates, and organisations, and this is affected by their encounters with automation and tools.

3.1 Benefits of Tools and Automation

The Cambridge English dictionary defines the term 'tool' as a piece of equipment used by a person, often handheld, to enable them to carry out a task. The person learns to use the tool, is in control of it, and uses it to enhance the way they carry out their activities and to improve the way they can deliver their craft. 'Automation' on the other hand is defined as the use of tools by machines to complete activities without or with minimal human intervention. An example of very simple tools are the needles and scissors used by an embroiderer. A simple sewing machine is still a tool. The sewer is in in charge of the machine, and guides the fabric through the needles to make the embroidery. A more complex modern machine includes software and automation; an embroidery pattern can be selected, a button pressed, and then the sewing machine guides the fabric through the needles and completes the embroidery with no human intervention. Similarly, laundry can be done by hand using simple tools such as a washboard and scrubbing brush, or using a simple washing machine, such as a twin-tub. This requires a lot of manual intervention and heavy, repetitive work. An automatic washing machine, once plumbed in, loaded and started with appropriate settings, removes the need for heavy and repetitive activities.

When tasks are automated successfully, we can measure productivity gains. We could also measure the effect on people who formerly carried out that task. This might be a positive effect, if tedious, repetitive, difficult or dangerous tasks are eliminated, freeing the person to do more interesting things. This automation could be life-enhancing and beneficial. One might think of the automatic washing machine removing the drudgery of laundry day, the modern sewing machine allowing long seams on heavy fabric to be sewn securely and also precise reproduction of the same embroidery pattern easily and quickly, or of industrial production lines being made safer with reduced industrial accidents.

Software testing is an essential part of software development and increasingly a focus for automation. Testing is hard to do well, time-consuming and expensive. It includes activities that are repetitive, such as executing tests on multiple occasions and also activities that are cognitively expensive, such as data comparisons. Some testing is not possible without the use of tools, either because it is too difficult or because it is too time consuming: "*no matter how valuable in-person testing is, effective automation is able to increase the value of overall testing by increasing its . . . range*" (Harty, 2011). These activities are candidates for automation, and many organisations are pursuing projects to automate some or all of their testing.

3.2 Effect of Automation on Motivation

Successful automation, where a person's work and life are enhanced may be a motivating factor for that person. However, there are circumstances where automation of all or part of a job could be demotivating. These include (1) the fear of redundancy (2) the effect of an unbalanced task mix (3) the effect of the automation being flawed.

3.2.1 Motivation, Redundancy and Dissociation

When a role or job can be completely automated, then the experience of the person who formerly had the role is of redundancy and this can have a massive negative impact on the individual, their families, and whole communities. The community of people sharing a task can build a sense of belonging and of self-worth. One's work-role can be the way one defines oneself. When we meet people, we commonly ask, "*What do you do?*" and the reply is often a job title, or even a former job title: "*I'm retired but I used to be...*".

In software testing, the dream of automatic testing is talked of by academics (Bertolino, 2007) and refuted by testing community experts, with software testing expert Michael Bolton rapping at a number of conferences "*Just don't tell me you can automate the testing*" (Toledo, 2017) and other testers publishing blogs, books, and industry conference papers on the limits of automation (Fewster and Graham, 1999; Johnson, 2011; Martin, 2017; Rachel, 2017). During conversations at industry conferences during 2019, software testers reported that their organisations had "*sacked all the testers*" as part of their desire to have all tests automated and replace people with technology.

Testers have a strong community (evidenced by the number of conferences, meetups and slack channels worldwide) which are mutually supporting and full of lively debate, but there is a fear in the community, noted for example by Bach and Bolton (2016) that "*the term 'test automation' threatens to dissociate people from their work.*"

Bach and Bolton distinguish between testing, an activity that by definition can only be done by a human because it involves high cognitive skill, and checking, a subset of the testing activity that can be automated because it is routine. Their view is: "*the basic problem is a shallow, narrow, and ritualistic approach to tool use. This is encouraged by the pandemic, rarely examined, and absolutely false belief that testing is a mechanical, repetitive process. Good testing, like programming, is instead a challenging intellectual process.*"

The threat posed by automation to testers' self-perception as worthy, intellectual, highly skilled individuals may be a real threat, or a perceived threat, and in either case, the attitude of testers to automation, and their experiences of automation projects are an interesting area of investigation, worthy of research, to help us understand whether the testers' experiences affects the automation project, as well as how the automation project affects the testers. Understanding the threat to testers' self-perception will require a multi-disciplinary understanding of the world of the testers, including insights from sociology, psychology and organisational engineering, among possible influences, and as part of our future work, we want to explore and use insights from those disciplines.

3.2.2 Motivation and Job Task Mix

In studies during the 1990's of the job design and motivation of IT workers, Warden and Nicholson (1996) found that the quality and testing roles are remarkable in being both the most boring and the most overstimulating and stressful of all the jobs in IT. Later motivation studies by Reid (2015) indicate that Hackman and Oldham's job design measures (Hackman and Oldham, 1974) are still a reasonable predictor of testers' motivation. Hackman and Oldham noted that if the task mix within a job is not optimized then that can lead to de-motivation, because jobs need to include a mix of the routine with the more challenging aspects of a role, providing a balance. It is possible for jobs to be stressful and demotivating if they are too boring, and also if they are too stimulating. If all the tedious and repetitive parts of a role are removed, then only the more stressful aspects are retained by the human, an overstimulating role remains. Therefore, the job mix for a tester needs to include routine as well as more challenging tasks to be maximally motivating.

3.2.3 Motivation and Flaws in Automation

All the above assumes that automation is successfully implemented and works well. If the implementation is not successful, and the automation is flawed - not working at all, giving the wrong results, working inconsistently - and is still being used, perhaps because of organisational factors, then whoever is responsible for the completion of the activity is left with both the more challenging and stressful, non-automated parts of their roles, together with stress of dealing with the fallout from flawed automation. Test automation is proving problematic for many teams and organisations, with reports of shelfware (that is tools acquired but not used) and reports of difficulties in using and maintaining tools (Gamba and Graham, 2018b; Graham and Fewster, 2012; Wiklund, 2015). Difficulties – what Wiklund refers to as impediments – that prevent successful testing via automation are often divided into technical and organizational impediments (Gamba and Graham, 2018b), and additionally researchers and practitioners report usability issues with tools and automation (Wiklund, 2015). If testers have to handle the flaws in automation, they may be demotivated by having to repeat tasks manually that should have completed in the automation, double-check results, and deal with increased uncertainty. Approximately one third of the testers we have interviewed and surveyed reported issues with their automation and tools. This, together with the findings from our literature review, indicate that investigating the experience of and attitude to problems in tools and automation is worth further investigation, particularly to look at how flaws in automation affect testers' motivation, and how it affects the UX of the testing tools themselves.

The scope of automation and tools for testing is also problematic. While Bertolino (2007) discusses 100% automatic testing as a dream for academia, and some tools vendors claim to reduce the time needed for people to be testing [1] or propose automation can cover up to 90% of testing [2], test experts and test automation experts make opposing claims, for example Bach and Bolton (2016). If organisations attempt to automate 90% of their testing (whether to refocus staff onto different work, increase the workflow, or to reduce staff numbers), or if an attempt is made to automate activities that cannot be automated, then people will be frustrated by a disconnect between what is expected of them, and what they are able to accomplish. The ironies of automation, noted by Bainbridge in 1983 (Bainbridge, 1983) and revisited more recently by Baxter et al. (2012) could be argued to particularly affect software testers; by the very nature of their role, they monitor, check, and challenge automation rather than trusting it, and the automation of part of their tasks can present them with the additional challenge of testing the test automation. Further, the complexities and varieties of the various environment and ecosystems within which software under test could be used is both a reason to automate, and yet a challenge to understanding whether the test automation can be trusted to accurately reflect real-life software use, especially with an increased emphasis in the industry on the UX of software over and above its functional accuracy.

3.3 Usability Flaws or Skills Gaps?

Testers are under increasing pressure to improve their technical skills, in order to be able to build and maintain test automation. A debate has started in the testing comunity about whether tester skill sets need to be improved, or whether improving the usability of tools would improve adoption (Hendrikson, 2010; Lambert, 2014; Reckless, 2017b). Some tools vendors are starting to address the perceived shortcomings in terms of tool usability, specifically in terms of learnability and interface design [based on personal observations at industry conferences and expos and in projects with a tool vendor]. Additionally, work is happening within the community to increase skill sets, for example the Test Automation University (Applitools) [3]. This discussion of tester skill sets was the initial impulse for the research study described in this paper. We started to look at whether the incidence of shelfware could be reduced, and the benefits of tools more clearly realized if tools were designed

[1] Automate Workday Testing, https://www.kainosworksmart.com/testing-workday/ (accessed October 2019)

[2] The Buyer's Guide for Selecting Software Test Automation Tools, https://www.microfocus.com/media/white-paper/the_buyers_guide_for_selecting_software_test_automation_tools_wp.pdf (accessed October 2019)

[3] The Automation University, https://testautomationu.applitools.com (accessed October 2019)

with usability for the existing testers, rather than testers needing to upskill to use the existing tools. Findings from our literature review (e.g. (Wiklund, 2015)) indicate that usability is an impediment to successful test automation, but not the only impediment. Our own findings indicate that improving usability may not necessarily result in a better experience for testers. We are investigating this idea further in our existing data, and in follow up data collection.

Our interim hypothesis is that usability is a necessary but not a sufficient condition for tools and automation to be successfully implemented and used. Furthermore, we suggest that if usability techniques are applied superficially this may add to the shelfware problem, by causing tools suppliers to provide attractive interfaces that make tools saleable, but which disguise other issues which manifest under long-term use, for example maintenance problems. This is a focus for our current research.

We also believe it is worth investigating the effect of the pressure on testers to improve their skills, and whether this challenge is more motivating or demotivating for testers. We propose this as a focus for future research, as motivation factors at work form part of people's lived experience. Based on personal experience and discussions at software testing industry conferences over the last two years, for many people their life at work significantly affects their overall wellbeing, health and lived experience. This was particularly observed during conversations at events such as 'Women Who Test' events in 2018 and 2019, workshops presented by one of the authors on 'People Factors for Test Automation' in 2018 and 2019, a workshop at the UK Software Testers' Retreat Weekend, and discussions during the 'Test Automation Face Off' Panel at STARCanada in October 2019.

3.4 Other Impediments to Automation

Our literature review to date confirms a number of other technical and organisational impediments to test automation. Literature identifies security and user access issues, environmental set up issues, and interoperability among groups of tools, among the technical impediments. Organisational problems include lack of management support, lack of budget for training, misunderstanding the scope of tools, and strategic/tactical direction changes which mean the chosen automation solution is discarded (Bach and Bolton, 2016; Gamba and Graham, 2018b; Graham and Fewster, 2012; Wiklund, 2015). These are major problems in tool and automation adoption, and various solutions to them are suggested by those authors.

Although we might treat these as simply technical or organisational issues to be overcome objectively, we also find evidence of a more human and emotional side to how those problems manifest. Bach and Bolton use the word "*pain*" to describe the interaction of testers and tools, while Gamba and Graham wrote a novel about test automation patterns, with the characters expressing their emotions about their work as well as their objective solutions to challenges (Gamba and Graham, 2018a).

In our own data, we have discovered more emotions, and more emotional language that we had expected, about both technical and organisational challenges. "*Stuck in limbo*" was a phrase used in one survey response to describe encountering security and access problems when trying to use a mandated automation tool, while the word "*magical*" was used by two respondents describing management expectations of automation as a solution to their perceived problem of testing being hard, expensive and time consuming.

A technical issue does not just lead to an objective search for a solution to the problem; the situation is not emotionless. Respondents to our survey expressed both frustration and anger at their being blocked by technical defects in automation and its environment, as well as pride and joy in being able to resolve those bugs in automation and tools. Strong emotions were also expressed around organisational issues, including frustration with managers and tool providers. Some respondents also expressed fear and self-doubt around their skill sets needed for using tools, using words such as "*scary*". Emotions were shown in the body language of interviewees and also in the written survey responses with negative emojis, strong language, and even a level of despair:

> "*I think I should leave my job and look for company who actually values testing.*"

> "*...if we make this crap again, what's the point?*"

What Wiklund (2015) describes as impediments to test automation were primarily technical and organizational, with a large proportion of usability issues. In our data, testers' responses to all their challenges and successes are emotional and related to their lived experiences. We therefore argue that these emotional responses, together with the effect on testers' work and lives makes lived experience of test automation a worthwhile area of research. Technical flaws - bugs in the automation and tools - and organisational issues are impediments to successful automation, and also directly affect the testers themselves, causing a degraded UX of the tool, negative emotions and demotivating factors affecting their lived experience.

Actual and perceived skill sets problems, lack of self-confidence, and a perception of the tools as difficult to use lead both to the need to improve the UX and usability of tools and automation, and also to the need to improve testers' self-confidence and learning opportunities. Flawed attempts to improve usability leads to tools that look appealing, but which provide a poor UX on prolonged use, because of issues with maintenance, interoperability and security.

4 CONCLUSIONS

The lives of testers whose organisations want to automate part, or all of the testing activities are affected in a similar way to any other job when automation is introduced in an imposed manner. Along with the benefits which improve people's lives, come the actual and perceived threats to motivation: redundancy, sub-optimal task mix in a job, and the stress of dealing with flaws in the automation and tools. The latter may affect even testers who are self-motivated to adopt automation. For academia, the possibility of automatic testing may be considered a desirable goal (Bertolino, 2007), however, this does not match the lived experience of testers, and their possible perception of automatic testing as either impossible or threatening their redundancy.

5 PROPOSED RESEARCH DIRECTION

Bringing together the discussion above, and our current findings, we argue that lived experience of tools and automation is a specific topic for further research, and includes their user experience. Our research includes analysis of our existing data, and interviewing and surveying more test practitioners about their lived and user experiences of tools and automation. We continue to collect and analyse testers' stories, and are also examining in the data about testers' backgrounds, skills and aspirations in order to better understand the tester personas.

We might ask "what is special about software testers?" Why do they merit a study of the automation of their tasks? Are testers different to other IT or knowledge workers - in their skills, the intellectual content of their work, or in their attitude to automation and tools? We acknowledge that future research could compare software testers with other knowledge workers to see if the phenomena we collectively observed over several decades are part of a general pattern or are specific to this group.

We present the concept of *TX: The Testers' Lived Experiences of Tools and Automation* which we regard as important both to individual testers and to the software testing community, and important to the software industry as a whole. It is a concept built from a complex mix of technical, organisational, usability, UX and human factors, which informs our understanding of the outcomes of attempts to automate testing, and provides a potentially rich seam of research in the HCI area and includes multidisciplinary strands across sociology, psychology and organisational change management.

ACKNOWLEDGEMENTS

The authors thank the HUCAPP paper reviewers for their useful and stimulating comments. We also thank the expert testers and automators we interviewed, and the test practitioners who took part in our surveys. We thank the organisers of the following conferences for allowing us to collect data at their events: UKSTAR, EuroSTAR, STARWEST, STAREAST, STARCanada, Women Who Test. We thank the ANZTB for enabling us to run a participants workshop in New Zealand, and also the UK Software Testing Retreat for discussing our research and interim results.

REFERENCES

Bach, J. and Bolton, M. (2016). A context driven approach to automation in testing. https://www.satisfice.com/download/a-context-driven-approach-to-automation-in-testing.

Bainbridge, L. (1983). Ironies of automation. In *Analysis, design and evaluation of man–machine systems*, pages 129–135. Elsevier.

Baxter, G. D., Rooksby, J., Wang, Y., and Khajeh-Hosseini, A. (2012). The ironies of automation: still going strong at 30? In *ECCE*, pages 65–71.

Bertolino, A. (2007). Software testing research: Achievements, challenges, dreams. In *2007 Future of Software Engineering*, pages 85–103. IEEE Computer Society.

Bolton, M. (2016). A context driven approach to automation in testing. https://www.developsense.com/blog/2016/01/a-context-driven-approach-to-automation-in-testing.

F. Lehner, R. D. and Abran, A. (2013). *Software Metrics: Research and practice in software measurement*. Springer.

Fewster, M. and Graham, D. (1999). *Software test automation: Effective use of test execution tools*. Addison-Wesley.

Gamba, S. and Graham, D. (2018a). *A Journey through Test Automation Patterns: One teams adventures with the*

Test Automation Patterns wiki. CreateSpace Independent Publishing Platform.

Gamba, S. and Graham, D. (2018b). Test automation patterns. http://testautomationpatterns.org.

Graham, D. and Fewster, M. (2012). *Experiences of test automation: Case studies of software test automation*. Addison-Wesley.

Groover, M. (2019). Automation. https://www.britannica.com/technology/automation.

Hackman, J. R. and Oldham, G. R. (1974). The job diagnostic survey: An instrument for the diagnosis of jobs and the evaluation of job redesign projects.

Harty, J. (2011). Finding usability bugs with automated tests. *Communications of the ACM*, 54(2):44–49.

Hendrikson, E. (2010). Do testers have to write code? http://testobsessed.com/2010/10/testers-code/.

ISO (2018). Iso 9241-11:2018 ergonomics of human-system interaction - part 11: Usability: Definitions and concepts. https://www.iso.org/obp/ui/iso:std:iso:9241:-11:ed-2:v1:en.

ISTQB (2018). Downloads - istqb® international software testing qualifications board. https://www.istqb.org/downloads/send/51-ctfl2018/208-ctfl-2018-syllabus.html.

Johnson, D. (2011). Is automated testing replacing the software tester? https://searchsoftwarequality.techtarget.com/tip/Is-automated-testing-replacing-the-software-tester.

Jones, C. (2015). Wastage: The impact of poor quality on software economics. *Software Quality Professional*, 18(1).

Kaner, C. (1988). Avoiding shelfware: A managers' view of automated gui testing. http://www.kaner.com/pdfs/shelfwar.pdf.

Lambert, R. (2014). Why testers really should learn to code. http://thesocialtester.co.uk/why-testers-really-should-learn-to-code/.

Martin, J. (2017). Using automation to assist –not replace – manual testing. https://smartbear.com/blog/test-and-monitor/using-automation-to-assist-not-replace-manual-test.

Norman, D. and Nielsen, J. (2019). The definition of user experience (ux). https://www.nngroup.com/articles/definition-user-experience/.

Porter, C. (2015). *Designing for experience-a requirements framework for enrolment based and public facing e-government services*. PhD thesis, University College London.

Rachel, L. (2017). Test automation can't replace manual testing. https://dzone.com/articles/why-test-automation-cannot-replace-manual-testing.

Reckless, C. (2017a). So what is software testing? https://www.ministryoftesting.com/dojo/lessons/so-what-is-software-testing.

Reckless, C. (2017b). Software testing shouldn't require expert coding skills. https:// www.ministryoftesting.com/ dojo/ lessons/ software-testing-shouldn-t-require-expert-coding-skills.

Reid, S. (2015). Motivated or motivating? what sort of tester are you? http://www.stureid.info/home-page/white-papers/motivation-testing-roles.

Tassey, G. (2002). *The Economic Impacts of Inadequate Infrastructure for Software Testing: Final Report*. Diane Publishing Company.

Toledo, F. (2017). Don't try to tell me you can automate the testing. https://dzone.com/articles/dont-try-to-tell-me-you-can-automate-the-testing.

Warden, R. and Nicholson, I. (1996). Motivational survey of it staff. *Software Futures*.

Wiklund, K. (2015). *Impediments for Automated Software Test Execution*. PhD thesis, Mälardalen University.