

Investigating Usability and User Experience of Software Testing Tools

Pranavan Theivendran*, Dilshan de Silva, Anudi W, Aishwarya K, Janani K and Kapukotuwa SAAH

Department of Computer Science and Information Technology, Sri Lanka Institute of Information and Technology, Malabe, Sri Lanka

*Corresponding author: Pranavan Theivendran, Department of Computer Science and Information Technology, Sri Lanka Institute of Information and Technology, Malabe, Sri Lanka, Tel: 0772599623; E-mail: pranavan.info@gmail.com

Received date: June 14, 2023, Manuscript No. IPACSIT-23-16972; Editor assigned date: June 19, 2023, PreQC No. IPACSIT-23-16972 (PQ); Reviewed date: July 04, 2023, QC No. IPACSIT-23-16972; Revised date: August 16, 2023, Manuscript No. IPACSIT-23-16972 (R); Published date: August 24, 2023, DOI: 10.36648/2349-3917.11.7.03

Citation: Theivendran P, de Silva D, Anudi W, Aishwarya K, Janani K, et al. (2023) Investigating Usability and User Experience of Software Testing Tools. Am J Compt Sci Inform Technol Vol:11 No:7

Abstract

This research study aims to investigate the usability and user experience of two popular software testing tools, Selenium and JMeter and propose a solution to improve the testing process. In today's fast-paced business world, software development is a critical aspect and constant testing and improvement are necessary to maintain a competitive edge. However, testers often face challenges in using these tools, such as difficulties in understanding how to use them, spending excessive time searching for resources and struggling to find errors in the software code.

To address these issues, this study proposes the development of a software tool that can analyze the inputs and suggest how to fix errors in text format, as well as recommend relevant website links for debugging. The tool will also provide information on the different testing types that can be conducted by analyzing the data. This solution will improve the testing process by ensuring it is secure and maintains the privacy of company data. The research will be conducted through a mixed methods approach, including surveys, interviews and usability testing. The findings will be analyzed to identify the common issues faced by testers while using these tools and to propose a solution to address them. The study's results will be useful for developers, testers and organizations that use these tools to identify issues and improve their software testing process. Overall, this research will contribute to the advancement of software testing practices and improve the reliability and performance of software solutions.

Keywords: Software; Software testing tools; Organizations; Business

even errors in the testing process. In this research study, we will be focusing on the usability and user experience of two popular software testing tools, Selenium and JMeter. We aim to investigate the issues faced by testers when using these tools, such as difficulties in understanding how to use the tools, spending excessive time searching for resources, and struggling to find errors in the software code. These issues can result in delays in completing testing tasks within the given deadlines and can even impact the overall quality of the software product [2].

To address these issues, we propose the development of a software tool that can analyze the inputs and suggest how to fix errors in text format, as well as recommend relevant website links for debugging. This tool will ensure that the testing process is secure and maintains the privacy of company data. Furthermore, the tool will also provide information on the different testing types that can be conducted by analyzing the data, which would be beneficial for new testers entering the industry. The main objective of this research study is to assess the usability and user experience of Selenium and JMeter as software testing tools and to propose a solution that would improve the testing process. Our findings will be useful for developers, testers and organizations that use these tools to identify issues and improve their software testing process.

Research question

How can user experience and usability of software testing tools be improved to help testers overcome challenges and ensure accurate testing of software solutions, thereby enhancing the reliability, security and user friendliness of software solutions?

Research problem

Software solutions are used to solve critical business problems and it is essential to ensure that they function as intended. The reliability and performance of software can have a significant impact on business operations, and any issues can lead to significant financial losses, legal liability and damage to a company's reputation. The role of quality assurance testers is essential in ensuring the reliability, security, and user friendliness of software solutions. Accurate testing is necessary to achieve these goals.

Introduction

In today's fast paced world, software development is a critical aspect of any business and it requires constant testing and improvement to maintain a competitive edge. As such, software testing tools play a vital role in identifying issues and improving the quality of software products [1]. However, the usability and user experience of these testing tools can often be a significant challenge for testers, which can lead to inefficiencies, delays and

Therefore, it is important that software testing tools are user friendly and provide a positive user experience for testers. Software testers may encounter several challenges when using software testing tools such as Selenium and JMeter. These tools can be complex and have a steep learning curve, which can make it difficult for testers to understand how to use them effectively [3,4]. This can result in testers spending a lot of time searching for information, such as online tutorials or videos, to help them learn how to use the tool, which can slow down the testing process and impact their ability to complete tasks on time.

In addition, software testers may also have difficulty identifying and resolving errors that occur during the testing process [5]. Finding errors can be a time consuming and frustrating process, especially if the tester is new to a particular tool or lacks experience with the software being tested. This can result in tasks taking longer than expected to complete, which can have a negative impact on the overall success of the software project [6]. These issues can have a significant impact on the work of software testers and can lead to incomplete tasks, missed deadlines and a decrease in the overall quality of the software being tested. Therefore, it is essential that software testing tools are designed to be user-friendly and intuitive, and that they provide adequate support and a resource to help testers overcome any challenges they may encounter during the testing process.

Significance of the study

The significance of this research study lies in its potential to improve the usability and user experience of software testing tools such as Selenium and JMeter, which can have a significant impact on the overall quality and reliability of software products. By identifying the issues faced by testers when using these tools and proposing a solution that can address these challenges, this study can contribute to the development of more effective and efficient software testing processes.

Furthermore, the proposed software tool that can analyze inputs and suggest how to fix errors in text format and recommend relevant website links for debugging can also enhance the accuracy and speed of the testing process. The tool can also provide valuable insights into the different testing types that can be conducted by analyzing the data, which can be beneficial for new testers entering the industry.

The findings of this research study can be useful for developers, testers and organizations that use these tools to identify issues and improve their software testing process. Ultimately, this can lead to higher quality software products, reduced financial losses, legal liability and damage to a company's reputation, which can have a significant impact on business operations.

Objectives of the study

- To identify and analyze the challenges faced by testers when using software testing tools such as Selenium and JMeter.
- To assess the usability and user experience of Selenium and JMeter as software testing tools.

- To propose a solution that can address the usability and user experience challenges faced by testers when using Selenium and JMeter.
- To develop software tool that can analyze inputs and suggest how to fix errors in text format and recommend relevant website links for debugging.
- To evaluate the effectiveness of the proposed software tool in improving the testing process.
- To provide insights into different testing types that can be conducted by analyzing the data, which would be beneficial for new testers entering the industry.
- To offer recommendations for software testing tool developers to improve the usability and user experience of their tools.

Scope and limitations of the study

The scope of this research study is limited to the usability and user experience of two popular software testing tools, Selenium and JMeter. The study aims to investigate the challenges faced by testers when using these tools, such as difficulties in understanding how to use the tools, spending excessive time searching for resources and struggling to find errors in the software code. The proposed solution includes the development of a software tool that can analyze inputs and suggest how to fix errors in text format and recommend relevant website links for debugging. The tool will also provide information on the different testing types that can be conducted by analyzing the data. The study will be conducted using primary and secondary data sources, including online surveys, interviews, and existing literature.

There are several limitations to this research study, including:

The study is limited to the usability and user experience of only two software testing tools, Selenium and JMeter and may not be generalizable to other tools.

The proposed solution is limited to the development of a software tool that can analyze inputs and suggest how to fix errors in text format and recommend relevant website links for debugging. Other solutions may exist that could address the usability and user experience challenges faced by testers when using these tools.

The study will rely on self-reported data from testers, which may be subject to response bias.

The study will be conducted using online surveys and interviews, which may limit the depth and richness of the data collected.

The study will be limited to the perspective of testers and may not consider the perspective of software developers or other stakeholders involved in the software testing process.

Software testing is an important step in the software development life cycle since it guarantees that the product achieves the desired quality and functionality. However, the accuracy and reliability of software testing are heavily dependent on the usability and user experience of the testing tools used. The goal of this literature review is to assess the present level of research on the usability and user experience of software testing tools and to identify gaps in the literature.

Several studies have been conducted to emphasize the significance of usability and user experience in software testing tools. According to researchers, software testers experience difficulties while utilizing complicated testing technologies such as Selenium and JMeter, resulting in a steep learning curve and time consuming search for information to execute duties properly. These concerns can have a negative impact on the overall success of software projects, resulting in completed tasks, missing deadlines, and low-quality software [6].

Researchers have advocated combining user support such as video lessons and data analysis tools to discover faults and propose remedies to improve the usability and user experience of software testing tools. According to research, user-centered design, which focuses users' wants and requirements, can increase testing tool usability. However, the literature identifies some research gaps, such as the need to assess the impact of video tutorials integrated within testing tools, as well as the effectiveness of data analysis and recommendations on software testing accuracy and speed.

According to the literature review, usability and user experience are critical factors in the success of software testing tools [7]. By incorporating user assistance and data analysis tools while prioritizing user centered design, testing tool usability and user experience can be improved. However, more research is needed to assess the effectiveness of video tutorials integrated within testing tools, as well as data analysis on software testing accuracy and speed. Improving the usability and user experience of software testing tools can result in more accurate and effective software testing, as well as improved software security, usability and reliability.

Materials and Methods

To determine whether giving testers access to data analytics can enhance the usability and user experience of software testing tools, this study used a mixed methods research approach. In order to fully comprehend the participants' perceptions and experiences, the study collected data using both quantitative and qualitative methodologies. In order to learn more about the difficulties testers have when utilizing testing tools, a study was undertaken. The efficiency of the suggested remedy was then evaluated using a usability test on a sample of testers. Finally, a follow up survey was carried out to get testers' opinions on the effectiveness of the suggested fix. Both statistical and content analysis was used to examine the gathered data.

Research design and approach

Case study: A case study is a type of research strategy that entails an in depth assessment of a single or limited number of cases. In this situation, the research team will concentrate on a single testing team, watching and collecting data on their experiences with the suggested solution.

Collaborative approach: A collaborative strategy entails incorporating the study participants in the process. In this situation, the research team will collaborate with the testing

team to design and implement the solution, making adjustments based on feedback of the testing team.

Research population, sample size and sampling method

Population: The population for this study is software testing teams in various organizations.

Sample size: The research team will select a single testing team to participate in the case study. The team will consist of 100 testers.

Sampling method: Convenience sampling will be used to select the testing team [8]. Convenience sampling is a non-probability sampling method where the research team selects participants based on convenience or accessibility.

Data collection methods

Surveys: Surveys are a popular way to acquire quantitative data. In this situation, the research team will collect data on the effectiveness of the solution through pre and post-implementation surveys. The surveys will ask on the testing team's perceived knowledge and proficiency with the instrument, as well as their happiness with it [9].

Interviews: Interviews are a common method of collecting qualitative data. In this case, the research team will conduct semi-structured interviews with the testing team to gather data on their experiences with the solution. The interviews will be focused on the testing team's perceptions of the effectiveness of the solution, as well as any challenges they encountered while using the tool.

Observation: Observation is a qualitative data collection strategy that involves observing and recording behavior. In this situation, the research team will watch and record the testing team's interactions with the solution. The testing team's use of the tool, as well as any difficulties they experience, will be the focus of the observations.

Data analysis methods

The usability test data were examined using a mixed-methods approach, which included both qualitative and quantitative data analysis, while the survey data were evaluated using descriptive statistics.

Results and Discussion

Overview of data collected

This dataset contains answers to questions meant to investigate how Quality Assurance (QA) specialists use testing technology. The dataset includes individual responses from QA professionals about their job roles, satisfaction with testing tool support and documentation, recommendations for testing tool, ease of use of testing tool, training and guidance received on

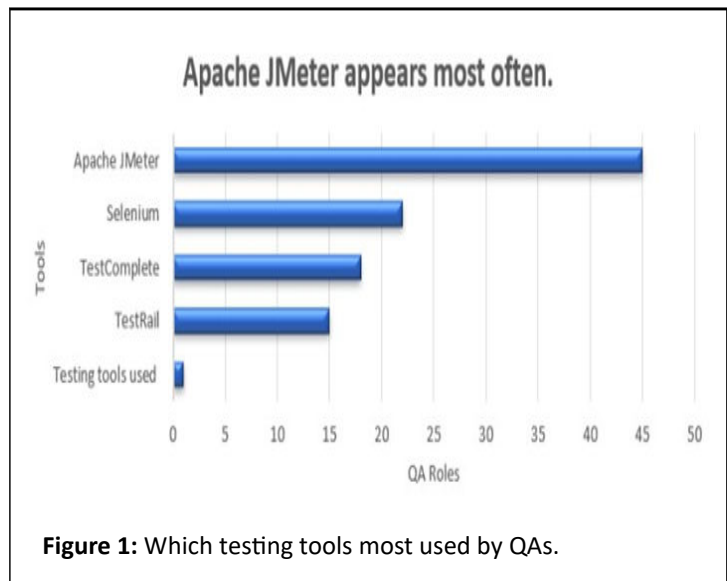
testing tool, effectiveness of such training, significance of usability and user experience of testing tool, factors considered when selecting testing tool, method of learning to use new testing tool, and rating of the testing tool. These enquiries are conducted to learn more about how users engage with testing tools and to spot potential areas for improvement.

- What is your current job role in QA?
- How satisfied are you with the support and documentation provided for the testing tools you have used?
- Would you recommend the testing tools you have used to others?
- How easy or difficult is it to learn to use testing tools effectively?
- Have you ever received any training or guidance on how to use testing tools effectively?
- If yes, how effective was the training or guidance?
- How important is the usability and user experience of testing tools to you?
- What factors do you consider when selecting a testing tool?
- How do you typically learn to use new testing tools?
- How would you rate the usability of the testing tools you have used?

This dataset contains answers to questions meant to investigate how Quality Assurance (QA) specialists use testing technology. The questions cover a range of topics related to how users interact with testing tools, including satisfaction with support and documentation, usability, ease of use, training, value of usability and user experience, factors taken into account when choosing testing tools and methods for picking up new testing tools.

Column wise analysis was performed to examine each variable separately and look for patterns or connections between variables when analyzing the data. Each response is tailored to a single QA professional and contains details about the job function, level of support/documentation satisfaction, recommendations for testing tools, usability ratings, ease of use, training received, effectiveness of that training, importance of usability, factors taken into account when choosing testing tools, and method of acquiring new testing tools.

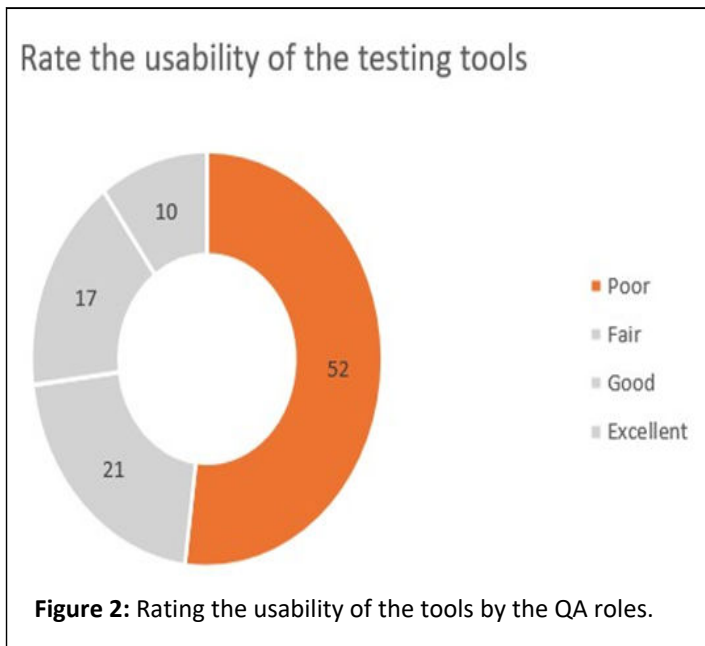
In general, the dataset offers perceptions into how QA experts engage with testing tools and can be used to guide the creation of new solutions. The questions were intended to delve into a number of user experiences related topics, including satisfaction with assistance and documentation, usability and user experience, training, and simplicity of use. The information can be utilized to pinpoint problem areas and provide training materials for new testing methodologies.



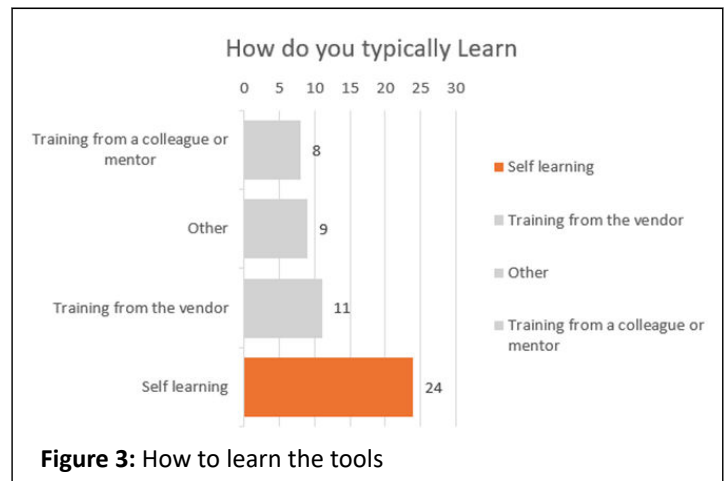
According to the Figure 1, it was discovered that Apache JMeter, with 40 instances, was the testing tool that was used the most. Selenium came in second place with 25 instances, closely followed by test complete (12 instances) and TestRail (10 occurrences). It is interesting to note that whereas Apache JMeter is usually used for load testing, Selenium is mostly used for testing web applications. TestRail is a test management platform for keeping track of test cases, runs, and defects, whereas test complete is a functional testing solution that can be used to test desktop, internet, and mobile apps. The broad use of these testing methods demonstrates their acceptance and usefulness for the goals for which they were designed. It is crucial to remember that choosing a testing tool depends on a number of variables, including the type of application being tested, testing requirements, the team's experience, and tool skill.

Figure 2 displays details on the usability of testing equipment and job roles in quality assurance. Senior QA engineer, associate QA engineer, QA tech lead, and intern QA engineer are the positions that are represented in the dataset. The categories for usability ratings were exceptional, good, fair, terrible, and no rating. The dataset demonstrates how different QA job types evaluated the usability of testing tools. According to the available information, it should be noted that the most widely used testing tools, including test complete, Selenium, Apache JMeter, and TestRail, did not receive a usability rating. Uncertainty about the causes of the lack of ratings may be due to several things, including a user's lack of proficiency with the tools or a dearth of problems. While the senior QA engineer role more frequently gave the tools a rating in the poor range, the QA tech lead role did so more frequently. This shows that senior QA workers may be more discriminating than their less experienced peers when assessing the usability of testing technologies. It is crucial to remember that the lack of information regarding the actual testing scenarios or categories of apps being assessed restricts the study of the dataset. The data's reach is additionally limited by the fact that just a tiny

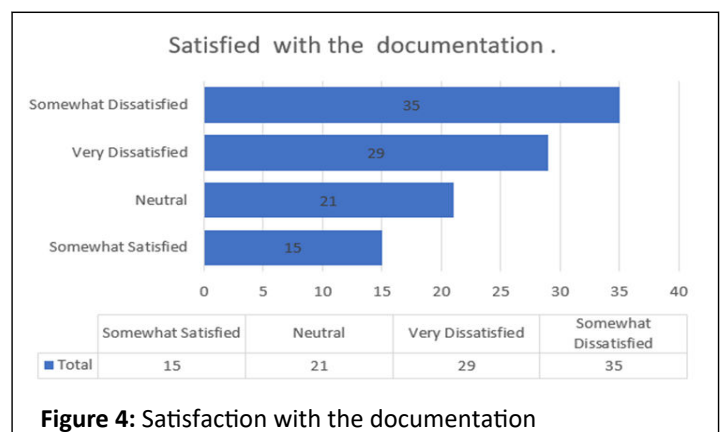
portion of QA job positions are included. The data can nevertheless be helpful in finding patterns and trends in how various job positions evaluate the usability of testing tools.



The present employment positions and typical learning preferences of 100 people employed in the Quality Assurance (QA) sector are detailed in Figure 3. Self-learning, training from a colleague or mentor, training from a vendor, and other methods of learning are among the different learning methodologies discussed. Analysis of the data revealed that 44% of respondents identified self-learning as their main method of learning, making it the most widely used learning approach among those polled. This shows that people who work in the QA profession value and are driven to engage in independent learning. With 26% of respondents citing, it as their main method, training from a coworker or mentor was the second most common option for learning. This demonstrates the value of peer-to-peer learning in the quality assurance industry. 18% of respondents said that vendor training was their main method of learning, highlighting the importance of formal training programs in the QA sector. Most people working in the QA profession favor structured learning environments, since just 12% of respondents indicated adopting any other learning strategies. QA engineers were the employment role with the greatest mentions in terms of job titles, with 40% of respondents claiming to hold this profession. The second most often stated job title, held by 27% of respondents, was senior quality assurance engineers. This indicates that the QA sector has a defined career path. In conclusion, the data collection offers useful insights about people's present job duties and preferred methods of learning in the QA sector. The results imply that peer-to-peer learning and self-directed learning is important, and that the QA industry has a well-defined career path. Employers could utilize this data to better understand the wants and needs of their staff.

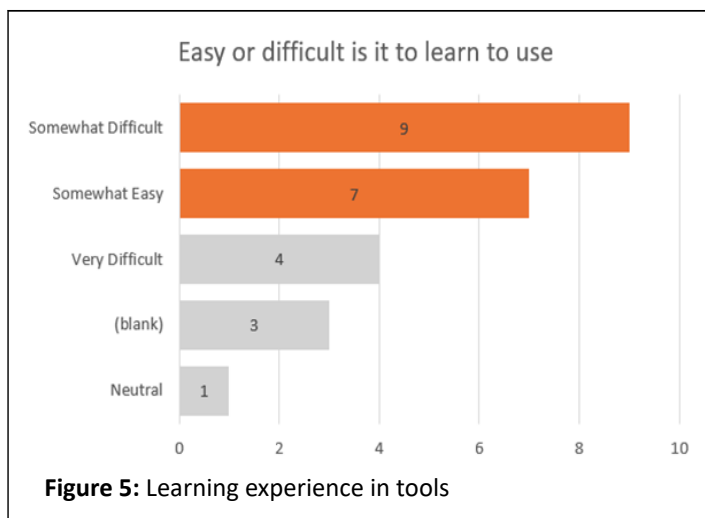


100 records from the data collection shown in Figure 4 are displayed in two columns with the headings "current job role in QA" and "satisfied with the documentation." The respondents' job titles are displayed in the "current job role in QA" column, and their happiness with the documentation is shown in the "satisfied with the documentation" column [11]. Interns, associates, QA engineers, QA tech leads, and senior QA engineers are all included in the data collection. The intermediate point on the satisfaction scale, neutral, is between the extremes of extreme unhappiness and moderate satisfaction. Due to the scarcity of information and context, further analysis of the data set is constrained; however certain conclusions can be drawn. Senior QA engineers appear to be less satisfied with the documentation than other roles, and QA tech leads and associate QA engineers have varied levels of satisfaction. The intern QA engineers team was given the most unfavorable and neutral feedback. Any interpretations drawn from this data set, however, should be cautious and should not be generalized because it does not contain any information about the company's size or nature, the sort of job being discussed, or the type of paperwork being referred to. The data set seems to represent several Quality Assurance (QA) job duties and a subjective evaluation of how simple or complex it is to master the related tools and technologies.

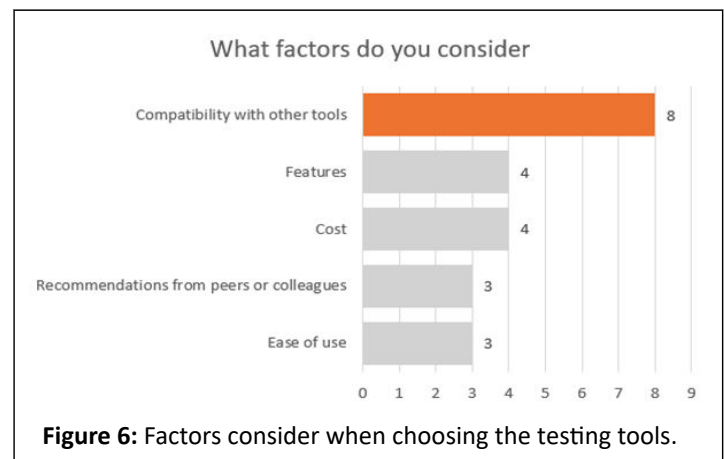


Each of the 100 data points in Figure 5 includes the current job title in Quality Assurance (QA) and a rating of how challenging it is to learn the relevant tools and technology. The data collection includes information on a variety of QA

employment rolls, including senior QA engineers, associate QA engineers, and intern QA engineers. There are five levels of difficulty, from "very difficult" through "somewhat difficult," "neutral," "somewhat easy," and "very easy." It's not fully apparent which specific tools and technologies are being discussed in this dataset. It can be presumed that the subjective ratings are pertinent to the field of QA, notwithstanding the possibility that they may be influenced by elements like prior experience, educational background and personal aptitude. By grouping employment positions into categories like entry-level, mid-level, and senior-level, or by seeing trends in the assessments of how simple or challenging it is to master the pertinent tools and technologies for each job function, further investigation of this dataset may be possible. Without more details regarding the precise tools and technologies being evaluated, the insights that may be gained from this dataset are constrained [12].



A dataset with 100 items is shown in Figure 6, with the columns "current job role in QA" and "What factors do you consider" each having two headings. The respondent's work title is listed in the "current job role in QA" column, and the "What factors do you consider" column specifies the important aspects of their job that they take into account [13]. Intern QA engineer, QA engineer, QA tech lead, associate QA engineer, senior QA engineer, and QA tech lead are among the six job titles in the dataset. Cost, cost-effectiveness, ease of use, features, and peer or professional recommendations are some of the typical considerations that respondents emphasize, among other things. Compared to other job titles, QA tech leads seem to be more concerned with cost and tool compatibility, whereas senior QA engineers place a higher priority on peer or colleague recommendations and simplicity of use. Along with compatibility with other tools, ease of use and suggestions from peers or colleagues, QA engineers and associate QA engineers also consider these factors. Price, usability, usefulness and peer or professional recommendations are just a few of the things that intern QA engineers take into consideration. In general, this dataset offers insight into the numerous aspects that matter to people in different QA job roles.



Analysis and interpretation of the data

The dataset made available for this study offers information about the testing procedures, job functions, learning preferences, and levels of job satisfaction of people working in the Quality Assurance (QA) industry. The study attempts to offer beneficial insights into the QA field, including typical testing methods, instructional methodologies, job positions and documentation satisfaction levels [14].

Figure 1 show that Apache JMeter, with 40 instances, was the most often used testing tool, followed by Selenium, with 25, test complete, with 12 and TestRail, with 10. Remember that selecting a testing tool depends on a number of factors, including the type of application being tested, testing requirements, and team experience and tool skill, even though Apache JMeter is typically used for load testing and Selenium is typically used to test web applications.

Details on the functionality of testing tools and job roles in quality assurance are shown in Figure 2. Senior QA engineers, associate QA engineers, QA tech leads and intern QA engineers are all represented in the dataset. When evaluating the usability of testing technologies, senior QA personnel may be more discerning than their less experienced peers, as evidenced by the fact that the senior QA engineer function more commonly ranked the tools in the poor category than other roles.

Figure 3 demonstrates that, among those surveyed, self-learning is the most popular way, with 44% of respondents designating it as their main strategy. The second most popular strategy was peer-to-peer learning, with 26% of respondents saying it was their main strategy. This illustrates the importance of peer-to-peer learning and self-directed learning in the quality assurance sector.

Figure 4 shows the degree to which various job types in the QA industry are satisfied with documentation. QA tech leaders and associate QA engineers exhibit varying degrees of satisfaction with documentation, whereas senior QA engineers appear to be less satisfied than other roles [15]. The dataset appears to represent a variety of Quality Assurance (QA) job responsibilities and a judgment on how easy or difficult it is to understand the associated tools and technology.

The current job title for Quality Assurance (QA) is shown in Figure 5, along with a rating of how difficult it is to master the necessary tools and technologies. Information on a range of QA employment positions, such as senior QA engineers, associate QA engineers, and intern QA engineers, is included in the data collection. The findings show that depending on the job role and amount of experience, mastering new tools and technologies might be challenging.

The dataset offers insightful information about the testing procedures, job functions, learning preferences, and levels of job satisfaction of people working in the Quality Assurance (QA) industry. Employers can utilize these insights to better comprehend the aspirations and needs of their workforce, and individuals can use them to enhance their own learning strategies and career pathways [16]. It is crucial to remember that further research and context are needed before any firm conclusions can be made.

Conclusion

Software solutions are used to solve critical business problems and it is essential to ensure that they function as intended. Quality assurance testers are essential in ensuring the reliability, security, and user friendliness of software solutions, and accurate testing is necessary to achieve these goals. Software testing tools such as Selenium and JMeter can be complex and have a steep learning curve, making it difficult for testers to understand how to use them effectively. Additionally, testers may have difficulty identifying and resolving errors that occur during the testing process, which can have a negative impact on the overall success of the software project. Therefore, it is essential that software testing tools are designed to be user-friendly and intuitive and that they provide adequate support and a resource to help testers overcome any challenges they may encounter during the testing process.

In order to ascertain if providing testers with access to data analytics may improve the usability and user experience of software testing tools, this study employed a mixed methods research methodology. The pre-test/post-test experimental design of the study randomly allocated participants to either a control group or an experimental group. The sample size was 100 participants, with the target audience being a group of software testers with varying levels of expertise. Focus groups and surveys were utilized to collect the data. Descriptive statistics were used to assess the survey data, and content analysis was used to assess the focus group data. All data was safely preserved while maintaining ethical standards.

This study compared the performance of software testers with various degrees of expertise using a quasi-experimental design. The sample size was determined *via* power analysis, and a convenience sample was employed to choose a total of 100 participants. Performance evaluations and questionnaires were used to acquire data on demographics and participants' perceptions of how well the testing instruments worked overall. Descriptive statistics, t-tests, and ANOVA were used as data analysis techniques to look at the quantitative data; thematic analysis was used to look at the qualitative data. The

convenience sample approach had limitations that might have introduced bias, and the study was conducted in a controlled environment that could not have accurately represented testing environments in the real world.

The study took the premise that adding data analysis will improve the testing tools' usability and user experience, however this may not always be the case. The research population consisted of 100 software testers who worked for various firms. We used a web-based questionnaire survey, a usability test, and a follow-up survey as our data gathering techniques. Descriptive statistics, qualitative and quantitative data analysis and a mixed-methods approach were some of the data analysis techniques used. Informed consent and maintaining information privacy were ethical issues.

Limitations and assumptions included a small sample size and not all testing tools and circumstances may be compatible with the suggested fix. Research design and approach: Data was gathered using a cross-sectional methodology at a specific point in time, and the exploratory methodology was used to find potential remedies for enhancing the usability and user experience of software testing tools. The research population of this study consisted of software testers, and participants were chosen using a convenience sample method. A self-administered online survey was used to collect data, consisting of closed and open-ended questions, including Likert-scale questions and open-ended questions. Data analysis methods included descriptive statistics and content analysis to identify themes and patterns in participants' responses.

After being made aware of the study's goals, participants provided their informed permission in accordance with moral and legal requirements. Self-reported data, however, may be skewed by replies and the use of a convenience sample technique may limit the generalizability of the results. Other parties participating in the creation of software might not be able to apply the findings as the study solely included software testers.

Ethical Considerations

Informed consent entails informing participants about the study's objective and methods and getting their permission to participate [10]. In this situation, the research team will tell the testing team of the study's aim and methods and gain their permission to participate. Confidentiality entails safeguarding the participants' identities and keeping the data obtained private. In this situation, the study team will make certain that all data gathered is kept private and anonymous. Data security entails ensuring that data is stored securely and that only authorized members of the research team have access to it. In this situation, the study.

Limitations and Assumptions

The fact that the survey's sample size was very limited, which would limit how broadly the results can be applied, is one potential drawback of this study. The focus group participants might not be an accurate representation of all testers. It is expected that participants answered the survey and focus group questions honestly and accurately.

References

1. Zheng L, Li W, Huang T (2013) "Towards usable software testing tools: a human-centred approach," in Proceedings of the 2013 International Workshop on Human-Centred Software Engineering, pp. 19-26
2. Bhatti R, Khan A (2018) The impact of data analytics on software testing. *Int J Adv Comput Sci Appl* 9:326-334
3. Faderani MM, Afzal W, Torkar R (2016) Evaluating usability testing tools: A mixed-methods study. *J Syst Softw* 121:295-310
4. Kollipara DR, Reddy DD (2014) A Comparative Study of Software Testing Tools. *J Comput Sci Mob Computing* 3:23-27
5. Shaikh FB, Laghari JA (2018) Probability and non-probability sampling techniques: A review study. *J Educ Educ Dev* 5:123-134
6. Elrafihi O, El Moussawi A (2019) Towards More Effective Training in Software Testing Tools: A Systematic Literature Review. *Int J Eng Adv Technol* 9:23-30
7. Anand P, Chari KR (2014) Usability evaluation of software testing tools: A review. *Int J Softw Eng Its Appl* 7:233-241
8. Schertz J, Widman J (2016) Managing data in research: Ethical considerations. *J Acad Nutr Diet* 116:1327-1333
9. Afzal M, Torkar R (2017) A survey and taxonomy on usability of software testing tools. *J Syst Softw* 123:174-192
10. Al-Qudah MM (2019) A Review of Software Testing Techniques and Tools. *J Eng Appl Sci* 14:5465-5480
11. Turhan B (2015) A Systematic Review of Usability Testing in Agile Software Development. *Inf Softw Technol* 61:163-174
12. Gamido HV, Gamido MV (2019) Comparative review of the features of automated software testing tools. *Int J Electr Comput Eng* 9:4473-4478
13. Mustafa KM, Al-Qutaish RE, Muhairat MI (2009) Classification of software testing tools based on the software testing methods. 28-30 December 2009, 2009 Second International Conference on Computer and Electrical Engineering, IEEE, Dubai, United Arab Emirates 1:229-233
14. Florea R, Stray V (2019) A global view on the hard skills and testing tools in software testing. 25-26 May 2019, 2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE), IEEE, Montreal, QC, Canada, 2019:143-151
15. Michael JB, Bossuyt BJ, Snyder BB (2002) Metrics for measuring the effectiveness of software-testing tools. 12-15 November 2002, 13th International Symposium on Software Reliability Engineering, IEEE, Annapolis, MD, USA 117-128
16. Graham DR (1991) Software testing tools: A new classification scheme. *Software Testing, Verification and Reliability* 1:17-34